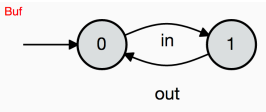
SWEN224 2015 Model Checking - Lab 2 - Concurrent Processes
Look up the README documentation

Activity 1

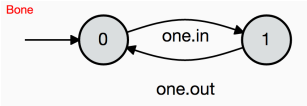
The goal is to both improve your understanding and to familiarize you with the process language.

A buffer is like a pipe that holds messages: messages are pushed in one end, move along the pipe with out changing order and later are removed from the other end of the pipe. An `N` place buffer is a pipe that can hold `N` messages.

1. Build a one place buffers:



2. Build the he labelled process **one:Buf**. The automata is built from `Buf` by prefixing it's events with `one`.



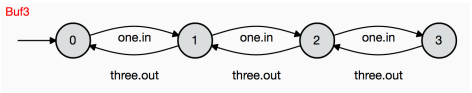
3. Construct a two place buffer from two one place buffers by renaming events to be `move` the output from one buffer and the input to the other buffer. Thereby synchronizing the output of one buffer and the input of the other buffer.

4. The synchronized `move` event is private, that is neither observed nor blocked by any other process. Hence use the function `_\{move}` to rename the `move` event as a `tau` event then remove the `tau` event and reduce the number of states using `simp(abs(_))`. Inspect the resulting automata to verify that you have built a two place buffer.

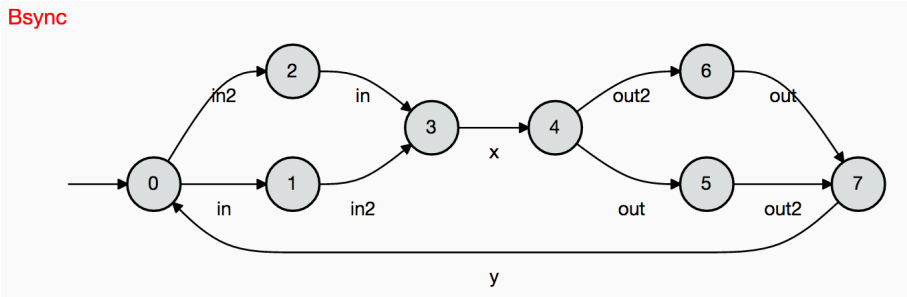
Activity 2

1. Build a three place buffer by composing three one place buffers.

   You can check your definitions by building their minimized automata.



2. Define two process, one with events `in,out` and the other with `in2,out2` and both with synchronizing event `x` and `y`. Define then so that the parallel composition of the two processes behaves like the automata below:



In words this is the interleaving of `in` and `in2` followed by the `x` event followed by the interleaving of the `out` and the `out2` events and finally the `y` event returns to the initial state.

Activity 3

1. Traffic lights, sensors and cars. Define the following system.

```
Sensor = carOn -> senOn->carOff->senOff -> Sensor.
TLred = red->TLred | turnGreen -> G,
    G = green ->G | turnRed -> TLred.
Car = carOn -> green -> carOff -> Car.

System = Sensor||TLred || Car.
```

   (a) Can a car move off the sensor if the light is green?

   (b) Does the system ever **deadlock** (end in a state from which no events can be executed)?

   (c) In the system, as defined, is it possible for the traffic light to keep changing colour before the driver moves?

2. Define a new Traffic Light `TLredX` by changing `TLred` so that it can only change from showing green to red after the car has moved of the sensor.

Marking Guide

Each lab is worth just under 1% of your overall mark for SWEN224. The lab should be marked during the lab sessions, according to the following grade scale:

- **0**: Student didn't attend lab.
- **E**: Student did not really participate in the lab.
- **D**: Student's participation was *poor*. For example, he/she made some attempt to work on the lab, but did not complete any activities.
- **C**: Student's participation was *satisfactory*. That is, he/she completed activity 1.
- **B**: Student's participation was *good*. That is, he/she completed activities 1 and 2.
- **A**: Student's participation was *excellent*. That is, he/she completed activities 1, 2 and 3.