Homework till the 13th of June

1) Let the following two threads be given.

```
x, y <- 0

thread 1 do
   x <- x + 1
   y <- y + 1
end thread

thread 2 do
  x <- x + 2
  y <- y + 2
end thread
```

Define an LTS for this program.

2) In a concurrent program, a process cycles continuously through two sections of code. The first section, denoted by n, is noncritical, whereas the second section, denoted by c, is critical, i.e., it is required that at most one process may access it. Before executing the critical section, the process visits a state, denoted by w, where it waits for access to the critical section. Informally, an execution path of a process is therefore

$$n \to w \to c \to n \to w \to c \to \cdots \ .$$

Access to the critical section is granted by a scheduler which may pick any of the waiting processes nondeterministically.

1. Define a suitable LTS that models the program with two processes. (Hint: you may want to label a state with $\langle s_1, s_2 \rangle$, where $s_i \in \{n_i, w_i, c_i\}$, for $i = 1, 2$.)

2. Execution might not be fair. There are infinite paths starting at $\langle w_1, w_2 \rangle$ where thread 1 never enters the critical section. Specify one such path.

3. Sketch roughly how three, instead of two, parallel processes could be modelled by means of an LTS.

4. Can you give a finite upper bound to the total number of states of the LTS that models the program with *twenty* processes?

5. Suppose now that the scheduler serves the waiting processes according to a first-come first-served policy. Define a suitable LTS that models the program with two processes.