Mandatory Exercise 1

Hand in 23-2-2005, before 12:00
in the pigeon hole of Mikkel
Bundgaard in the Dept. of
Theoretical CS, 4C corner
Part 1

- Write a CCS expression $T(\text{start}, \text{stop}, \text{join})$ denoting the LTS:

  \[
  \begin{array}{ccc}
  \text{Created} & \xrightarrow{\text{start}} & \text{Alive} \\
  & \xrightarrow{\text{stop}} & \\
  & \xrightarrow{\text{stop}} & \text{Dead} \\
  & \xrightarrow{\text{join}} & \\
  & \xrightarrow{t} & \\
  \end{array}
  \]

  where $t$ is the special label for an “internal” transition. The action join represents that the main program can wait for the Thread to finish by the join() method.

- Write the expression in the syntax of Mobility Workbench, and verify that the syntax is correct.
Part 2

- Draw the LTS for the CCS process

TwoThreads(startA,stopA,joinA,startB,stopB,joinB) =

\[ t.(T<startA,stopA,joinA> \parallel t.(T<startB,stopB,joinB> )) \]

where T is the thread you defined in part 1. Explain in your own words how this expression models a Java program that creates two Java Threads.

- How many states would the LTS have if 3 threads was created? (Extra non-mandatory exercise, a bit more difficult: How many states would the LTS have if k threads was created?)

- Step through the first 3 steps of the process in MWB
Part 3

• **Draw the LTS for the CCS process**
  TwoThreads(printfinished) =

  (^startA,startB,joinA,joinB,stopA,stopB)(
  t.(T<startA,stopA,joinA> |
  t.(T<startB,stopB,joinB> |
  'startA.'startB.'joinA.'joinB.'printfinished.0)))

  where T is the thread you defined in part 1. Explain in your own words what this process expression models.

• **Give a formal inference of two of the transitions of the process, at least one that uses the SYNC rule.**