VU Programm- und Systemverifikation
Homework: Model Checking with SMV

(10 points)

May 13, 2016

Encode in SMV the first algorithm—Solution with K-state Machines ($K > N$)—described in the article "E.W. Dijkstra. Self-stabilizing Systems in Spite of Distributed Control. Communications of the ACM, Volume 17, Issue 11, Nov. 1974", which is freely available, e.g., at

`http://courses.csail.mit.edu/6.852/05/papers/p643-Dijkstra.pdf`

Consider the case of 6 processes (1 bottom machine and 5 other machines), with 10 states, that is, $N = 5$ and $K = 10$. Note that the algorithm is asynchronous, so you will have to specify processes or encode the scheduling yourself (e.g., using a global variable that ranges over 0..5 and is not restricted, and determines which machine moves). In order to make the system work, you will also have to specify a fairness (or justice) constraint that specifies that each machine moves infinitely often.

Also encode the following specifications:

1. Eventually, a state is reached where all machines have the same state. (That is, the bottom machine has the privilege)
2. For each "other machine", infinitely often its state is different from the left neighbor.

Check them with

- NuSMV: `http://nusmv.fbk.eu`, or
- nuXmv: `https://nuxmv.fbk.eu`.

Also try $K = 1000$. Explain what happens and why in a comment in the file `token.smv`.

Upload a file `token.smv` with your solutions, and an `output.txt` file containing the output of the tool to TUWEL by June 1st, 2016.