Formal Modeling and Verification
Master CORO M2 – Embedded Real-time Systems Track
Lab work

Duration: 8h

1 Basic Understanding of Uppaal

Q1. (If using your own machine) Download Uppaal at `http://www.uppaal.org`;

Q2. Look at the demo of the teacher and do the same;

Q3. For further inspiration, you may want to also look at examples `fischer.xml` and `train-gate.xml` in the Uppaal distribution. You can find their descriptions and some explanations, as well as some material explaining how Uppaal works, in the *Uppaal Tutorial* at `http://www.it.uu.se/research/group/darts/papers/texts/new-tutorial.pdf`.

2 Peterson's Mutual Exclusion Protocol

Q4. Model Peterson's mutual exclusion protocol;

Q5. Verify mutual exclusion and the absence of deadlocks;

Q6. Verify the possibility of starvation;

Q7. Suppose now that each process always waits at least 10 time units between two accesses to the critical section. Suppose also that the work in critical section takes 3 time units or less to complete. Add these constraints to the model and verify again the starvation property.

3 A Real-time Scheduling Problem

We consider a scheduling problem, adapted from [BFSV04] for a non-preemptive setting in [JLR15] : we have three real-time tasks τ_1, τ_2 and τ_3. Task τ_1 is periodic with period a and has an execution time $C_1 \in [10, 20]$. Task τ_2 is sporadic : it only has a minimal delay between two activations and that delay is $2a$. The execution time of τ_2 is $C_2 \in [18, 28]$, with $c \leq d$. Finally, τ_3 is periodic with period $3a$ and has an execution time $C_3 \in [20, 28]$. These three tasks are scheduled using a non-preemptive[1] priority policy defined by $\tau_1 > \tau_2 > \tau_3$.

We first assume that $a = 50$.

Q8. Model this system using Uppaal ;

Q9. Is the system schedulable, when deadlines are supposed to be on requests (i.e. only one instance of each task may exist at all times) ?

Q10. What is the smallest value of a for which the system is schedulable ?

Q11. What is the worst-case response time of task τ_3 ?

4 A Job-shop Scheduling Problem

We now look into a problem of job-shop scheduling. In such problems, we are given m machines and n jobs to perform. Each of those job use in sequence all the machines, but not necessarily in the same order. A given machine can only process one job at all times, and each job may need a different duration on a given machine. The example described here is taken from [AAM06].

The following table gives the order of use of machines for each job ($m = 6, n = 4$).

Job	M1	M2	M3	M4	M5	M6
1	2	4	3	5	1	6
2	2	5	1	3	6	4
3	6	5	3	1	2	4
4	1	3	6	4	5	2

The next table gives the minimum and maximum time that each job spends on each machine.

Job	M1	M2	M3	M4	M5	M6
1	$[34, 34]$	$[21, 54]$	$[74, 74]$	$[6, 26]$	$[5, 5]$	$[43, 43]$
2	$[24, 24]$	$[13, 28]$	$[53, 53]$	$[8, 8]$	$[16, 23]$	$[45, 45]$
3	$[35, 75]$	$[14, 14]$	$[8, 15]$	$[31, 31]$	$[24, 24]$	$[6, 6]$
4	$[12, 42]$	$[25, 32]$	$[15, 15]$	$[42, 42]$	$[62, 62]$	$[18, 18]$

Question 12 Model this problem with Uppaal ;

Question 13 Find the minimum time for the completion of all jobs.

Références

[AAM06] Yasmina Abdeddaïm, Eugene Asarin, and Oded Maler. Scheduling with timed automata. *Theoretical Computer Science*, 354(2) :272 – 300, 2006.

[BFSV04] G. Bucci, A. Fedeli, L. Sassoli, and E. Vicario. Timed state space analysis of real-time preemptive systems. *IEEE Trans. on Soft. Eng.*, 30(2) :97–111, February 2004.

[JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for real-time systems. *IEEE Transactions on Software Engineering (TSE)*, 41(5) :445–461, 2015.

1. A running task cannot be interrupted even if another task with a greater priority is ready.