Games with perfect information
Exercise sheet 4

Sebastian Muskalla

TU Braunschweig

Summer term 2019

Out: May 8

Due: May 15

Submit your solutions on Wednesday, May 15, during the lecture.

Exercise 1: An intricate scheduling problem

Consider the instance of MOFST with 2 processors, and the set of tasks $\mathcal{T} = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6\}$, where the computation time C_τ, the relative deadline D_τ, and the minimal interarrival time T_τ are given by the following table.

	C_τ	D_τ	T_τ
τ_1	2	2	5
τ_2	1	1	5
τ_3	1	2	6
τ_4	2	4	100
τ_5	2	6	100
τ_6	4	8	100

Recall that the jobs can be freely migrated between processors after each tick, but they have to be processed sequentially, i.e. not both processors can work on the same job during one tick.

a) Assume that each task generates a job as soon as the minimal interarrival time has elapsed, i.e. all tasks generate a job at time 0 (at the beginning of the first tick), τ_1 and τ_2 generate a job at time 5 (at the beginning of the sixth tick), τ_3 generates a job a time 6, and so on.

   Consider the time interval $[0, 8]$ (i.e. the first 8 ticks). Show that there is a scheduling of the jobs for this interval that makes no job miss its deadline. Give a graphic representation of your scheduling.

b) Now consider the execution in which the generation of the second jobs for the first two tasks is delayed: The tasks τ_1, τ_2 and τ_3 generate their second job at time 6.

   Prove that there is a scheduling of the jobs in the interval $[0, 8]$ such that no job misses its deadline by giving a graphic representation.

Note: One can actually prove that the instance is **feasible for offline scheduling**: If the scheduler knows when each task will generate jobs a priori, then it is always possible to find a schedule so that no jobs misses its deadline.

Exercise 2: An intricate scheduling problem II

Consider again the instance of MOFST from the first exercise. Prove that the input is a no-instance of MOFST, i.e. it is infeasible for online scheduling, where the delay with which each task is generated is not known to the scheduler a priori.

Hint: Towards a contradiction, assume that an online scheduler exists. Show that by time 8, at least one job has missed its deadline. Structure your proof as follows:

- Assume that all tasks generate a job at time 0. Note that this fixes the jobs for the time interval $[0, 5]$, and since the online scheduler has no knowledge when which job will be generated later, fixes a scheduling on the interval.
- For this fixed scheduling, there are two cases:
  - Case 1: The job generated by task τ_5 is not scheduled on any processor in the time interval $[2, 4]$.
  - Case 2: The job generated by task τ_5 is scheduled for at least one step on a processor in the time interval $[2, 4]$.

  Show that for each of the cases, there is a possible generation of jobs in the interval $[5, 8]$ that makes a job miss its deadline.