Games with perfect information
Exercise sheet 5

Out: May 4, **Updated: May 7** Due: May 12

Submit your solutions until Friday, May 12, 14:00, in the box next to office 343. Note the change in the schedule – there will be no exercise classes on May 9. The exercise classes for this sheet will take place on May 16.

Exercise 1: Proof of Lemma 7.13

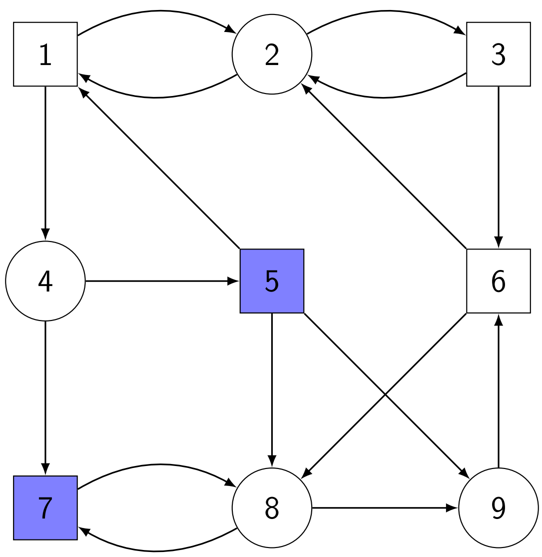
Prove Lemma 7.13, Part c1) and c2) from the lecture notes:

c1) For all $x \in P^m \cap V_{\bigcirc}$, and all successors y of x, $\delta(x) \geq \delta(y)$ holds.

c2) If additionally $x \in B$, the inequality from c1) is strict for all successors y of x, $\delta(x) > \delta(y)$

Exercise 2: A Büchi game

Consider the following game arena. As usual, vertices of prover are drawn as boxes, those of refuter as circles.



Consider the Büchi game with respect to the winning set $\{5, 7\}$, i.e. refuter wants to visit the blue-colored vertices infinitely often.

Solve the Büchi game using the recurrence construction. Give the sets B^i, P^i for all i, and give all sets $\mathrm{Attr}_{\bigcirc}$ and $\mathrm{CPre}_{\square}$ that are needed to compute them.

Exercise 3: An intricate scheduling problem

Consider 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

We assume that we have 2 processors. 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, τ_1 and τ_2 generate a job at time 5, τ_3 generates a job a time 6, and so on.

   Consider the time interval $[0, 8]$. 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) Prove that the input is infeasible for online scheduling if we allow the tasks to delay the generation of jobs.

   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 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 that makes a job miss its deadline.