Petri nets homework

- **Due on Monday 5 December at 23:59**
- **You may complete the homework individually or in teams of up to 4 students.**
- **Submit your homework as a zip file using the Submit button in the course Web page. The zip file should contain a Woped (.pnml) file with the solutions to tasks 1 and 2 (part A), and a Word, OpenOffice or PDF file with the answers to part B.**

Part A. Automated Factory Line

We consider a segment of a factory with two conveyor belts, two machines, one robot and one buffer. Raw parts arrive through a first conveyor belt, called the **raw line**. The robot moves each part from the **raw line** into machine M1. Machine M1 immediately starts processing the raw part. Eventually, the machine will finish processing the part and the robot may then move the processed part into the buffer. Eventually, the robot moves the part from the buffer into machine M2. Machine M2 will start processing it. When machine M2 finishes processing the part, the robot will then move that part from machine M2 into a second conveyor belt (called the **finished line**).

Figure 1 illustrates the factory line in the state where the raw line holds 2 parts, M1 and M2 hold one part each, the buffer holds two parts, and the finished line holds one part.

![Figure 1: Factory Line Structure](image)

M1 can hold at most one part at a time, and the same applies for M2. The robot can only move one part a time. When machine 2 is free, the robot can move a part directly from machine 1 into machine 2 instead of moving the part to the buffer.

The conveyor belts can hold any number of parts but the buffer can hold at most 7 parts. When the buffer is full, the robot will not put a part into machine 1, even if machine 1 is empty. This constraint is intended to avoid the situation where machine M1 finishes processing a part and the robot is not able to take away the processed part from M1.

This segment of factory line (in its “empty” state) is modelled as a Petri net in Figure 2.
**Figure 2:** Petri net of the system in its initial (empty) state.

**Task 1. [1 point]**
From time to time, it happens that one of the machines breaks down. In this case the machine moves into a “broken” state and an alarm is triggered so that an operator fixes the problem. If the machine breaks down while it is processing a part, the operator discards this part. If the machine breaks while it holds an already processed part, the robot moves this processed part into the buffer (in the case of M1) or into the finished line (in the case of M2). Once the machine is empty, the operator eventually repairs it and restarts the machine. While one of the machines is broken, the other one may continue to work.

Modify the above Petri net from in order to capture the possibility of machine break-downs.

**Task 2. [1.5 points]**
Sensors are added in the finished line in order to measure the size and weight of the finished parts. If the sensor detects that one of the parts does not fit certain requirements, a robotic arm attached to the finished line is triggered. The robotic arm removes the defective part from the finished line and drops it into a bin. If two consecutive parts are found to be defective, a red light in the raw line is turned on and the robot stops taking new parts from the raw line. Eventually, a technician checks the factory line, perform some adjustments and restarts the line. When the line is restarted, the red light is turned off and the robot can start taking parts from the raw line again.

**Acknowledgments:** This exercise is inspired by a similar by exercise by R. Wattenhofer, ETH Zürich.

**Part B. Properties of Petri nets and BPMN models**

**Task 3. [1.5 points]**
We consider the following three Petri nets: A, B and C. For each of these Petri nets, state if it is bounded, live and deadlock-free. Explain how did you reach your conclusions.
Acknowledgments: This exercise is inspired by a similar exercise by J. Esparza, TU Munich.

Task 4. [2 points – out of which 1 point is “bonus”, meaning on top of the 5 points of this homework]

In homework #1, you saw a translation from a small subset of the BPMN modeling notation to Petri nets. This translation is such that if the input BPMN process model has only one start event and one end event, the resulting Petri net has one single place with no incoming arc (hereby called S) and one single place with no ongoing arc (hereby called E).

a) Give an example of a BPMN process model and its corresponding Petri net, such that there exists a firing sequence starting from the initial marking and leading to a marking M in which there is no token in place E. Note: In the field of process modeling, this is called a “deadlock” because the execution of the process gets blocked before the end event is reached.

b) Give an example of a BPMN process model and its corresponding Petri net, such that there exists a firing sequence starting from the initial marking, and ending in a marking M in which there is a token in place E, but there is also at least one token in at least one other place besides E. In the field of process modeling, this is called “improper termination”, because the end event has been reached, but there is at least one token (or more) left behind.

c) Give an example of a BPMN process model and its corresponding Petri net, such that the Petri net is not bounded.

d) Give an example of a BPMN process model and its corresponding Petri net, such that the Petri net is bounded, and there exists a firing sequence starting from the initial marking (i.e. the marking in which S has one token and all other places are empty) to a marking M in which there are two tokens in the same place (and possibly other tokens in other places).