**Exercises: Plain Petri nets (session 2)**

**Exercise 1 Reachability graph**
Give the reachability graph of the following Petri net.

**Exercise 2. The philosophers problem**
Five philosophers sitting around the table share five chopsticks: chopsticks are located in-between philosophers. A philosopher is either in state eating or thinking and needs two chopsticks to eat. Model the behaviour of the philosophers as a Petri net. Compute the reachability graph of this net. What can you conclude from it?

**Exercise 3. Properties of Petri nets**
Is the following Petri net deadlock-free? Is it bounded? Is it live?
Acknowledgment. This exercise is inspired by a similar exercise by Barbara König, University of Duisburg-Essen.

Exercise 4. Automated Factory Line

We consider a segment of a factory with two conveyor belts, two machines, one robot and one buffer (see Figure 1). 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 will move the part from the buffer into machine M2. Machine M2 will immediately start processing it. Eventually, machine M2 will finish processing the part and the robot may then move that part from machine M2 into a second conveyor belt (called the finished line). 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. The buffer can hold at most 7 parts. The conveyor belts can hold any number of parts. The following figure shows the case 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.

Tasks

1) Capture the above factory line as a Petri net.
2) Modify the Petri net from question (1) so that it is possible for a part to be moved directly from machine 1 (M1) to machine 2 (M2) if machine M2 is empty. Can you modify the resulting net so that if the part can be moved from M1 to M2 directly, it is moved this way (i.e. the part in M1 is not put into the buffer if it can be moved to M2)?
3) Modify the Petri net from question (1) so that when the buffer is full, the robot will not put a part into M1, even if M1 is empty.
4) Is the net in task 3 live? Does it have a deadlock?
5) Does the net you designed for task 3 have an unbounded place? If not, how can you modify it so that the net is bounded?
Exercise 5. Insurance Claims Handling Workflow

Insurance company X processes claims which result from traffic accidents with cars where customers of X are involved in. Therefore, it uses the following procedure for the processing of the insurance claims. Every claim, reported by a customer, is registered by an employee of department CD (CD = Car Damages). After the registration of the claim, the insurance claim is classified by a claim handler of rank A or B within department CD. There are two categories: simple and complex claims. For simple claims two tasks need to be executed: check insurance and phone garage. These tasks can be executed in parallel. The complex claims require three tasks to be executed: check insurance, check damage history and phone garage. These tasks need to be executed sequentially in the order specified. Both for the simple and complex claims, the tasks are done by employees of department CD. After executing the two respectively three tasks a decision is made. This decision is made by a claim handler of rank A and has two possible outcomes: OK (positive) or NOK (negative). If the decision is positive, then insurance company X will pay. An employee of the finance department handles the payment. In any event, the insurance company sends a letter to the customer who sent the claim. An employee of the department CD writes this letter.

Capture the above process as a Petri nets.