UNIVERSITY OF TARTU

1632

Institute of Computer Science

MTAT.03.083 – Systems Modelling

Final Exam – 12 January 2010

QUESTION 1: The Way to Princess Ada **[40 points]**

Imagine a castle consisting of rooms connected by doors. In one room is Princess Ada. One room has the entrance gate of the castle. Outside the castle is a prince who wants to marry the princess. Every room has a cost, which has to be paid each time someone enters the room. The prince has an amount of money in his pocket. The prince needs to find a way through the castle, so that he will be able to marry the princess. The princess will only marry him, if when he reaches her, he has spent exactly all of his money. If he enters her room, and he has still money left, he will be considered greedy and the princess will not marry him. If he enters her room and has a debt he will be considered wasteful. The prince can enter into the same room several times, but he can only enter Princess Ada's room once.

Let us assume we have the following setup:

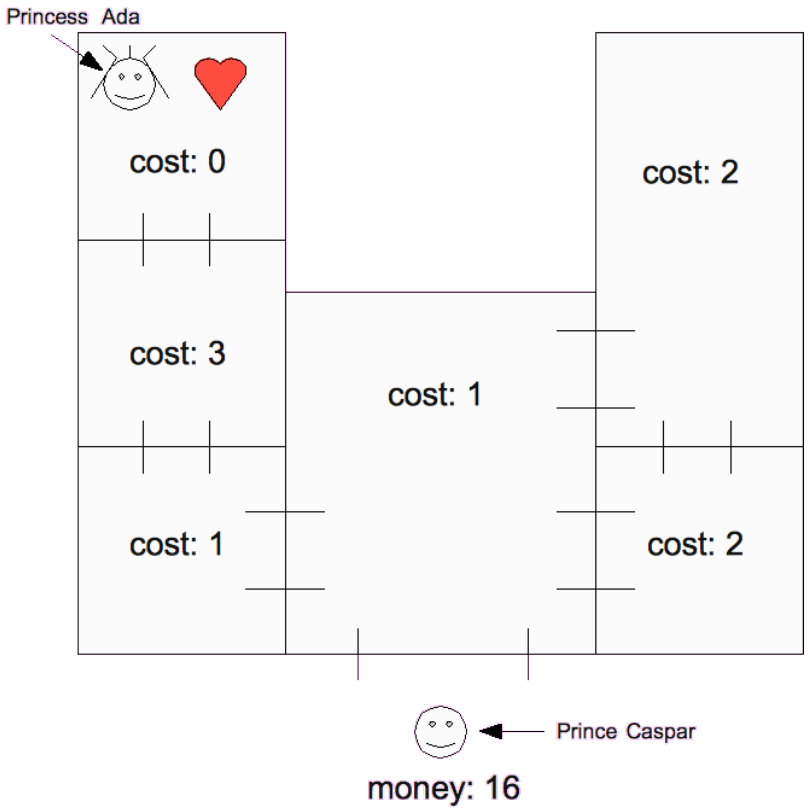

Figure 1: A Castle with Princess Ada

a) Draw an object diagram capturing the situation shown in Figure 1. **[5 points]**
b) Draw an object diagram after the prince has walked to a second room into the castle. **[5 points]**
c) Derive a class diagram from the two object diagrams you drew in questions (a) and (b). **[9 points]**
d) Draw a plain Petri net capturing the situation shown in Figure 1. The Petri net should prevent the prince to cross a door if he does not have enough money left to do so. **[7 points]**
e) Could you use the reachability graph of your Petri net in order to find a sequence of moves that would allow the Prince to marry Princess Ada? If your answer is no, explain how would you modify your Petri net so that you can find a way to Princess Ada using the reachability graph. If your answer is yes, explain how would use this reachability graph in order to find a way to Princess Ada. **[3 points]**
f) Is there enough information in the object diagram you drew in question (a) to automatically generate the Petri net you drew in question (d)? **[1 point]**
g) Draw a CPN capturing the situation shown in Figure 1. This CPN should prevent the prince to cross a door if he does not have enough money left to do so. The CPN should include guards and arc inscriptions **[7 points]**
h) Is it possible to run a single simulation of your CPN and to use the output of the simulation in order to to find a sequence of moves for the Prince to marry Princess Ada? If your answer is no, explain how would you modify your CPN so that you can find a way to Princess Ada by means of a single simulation. If your answer is yes, explain how would use the output of the simulation in order to find a way to Princess Ada. **[3 points]**

QUESTION 2: Digital Watch **[10 points]**
Consider the statechart of a digital watch shown in Figure 2. Modify this statechart to support a fast setting of the time. If button B is pressed and held for more than 5 seconds, the hours or minutes (depending on the current state) will start incrementing once every half second while button B is kept pressed.

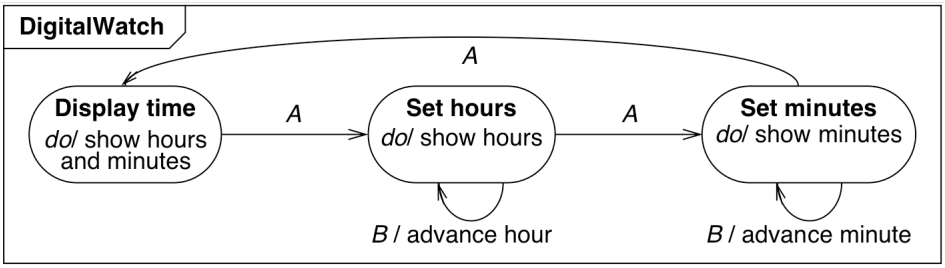

Figure 2: Statechart of a digital watch

QUESTION 3: Factory Line **[10 points]**
We consider a segment of a factory with two conveyor belts, two machines, one robot and one buffer (see Figure 3). Raw parts arrive through a first conveyor belt, called the *raw line*. The robot moves each part from the *raw line* into machine M1, then into the buffer, then into machine M2 and finally to the 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.

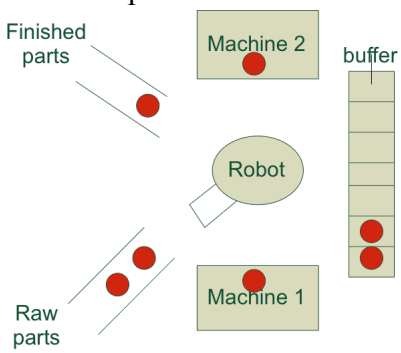

Figure 3: Factory Line Structure

This system (in its initial state) is modelled as a Petri net in Figure 4.

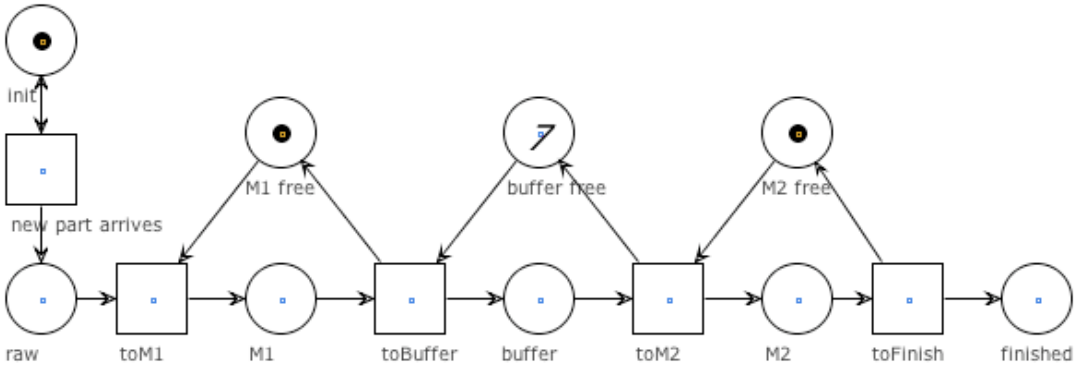

Figure 4: Petri net of the factory line

Tasks:

a) Modify the Petri net in Figure 4 so that when the buffer is full, the robot will not put a part into machine 1, even if machine 1 is empty. **[5 points]**

b) Modify the Petri net in Figure 4 so that when machine 2 is free, the robot is able to move a part directly from machine 1 into machine 2. **[5 points]**