Exercises on Probabilistic Model Checking

March 31, 2005

Exercise 1

Consider a system that continuously tosses a coin and records the last outcome head (**H**) or tail (**T**), which have equal probability.

1. Model the system as a DTMC
2. Specify in CTL the properties *it exists a path such that*:
   (a) the first outcome is **H**,
   (b) the 2nd and 3rd outcomes are **T**,
   (c) eventually there are 2 **T** in a row,
   (d) there are never 3 consecutive **H**.
3. Compute the probability with which the initial state satisfies the properties above.
4. What is the probability to obtain at least one or exactly two **H** in the first three tossings?

Exercise 2

A laboratory mouse finds himself in the maze depicted in Figure 1, which has only two exits: one where there is cheese to celebrate its freedom (up), and the other one leads directly back to the cage (down). The other nodes represent

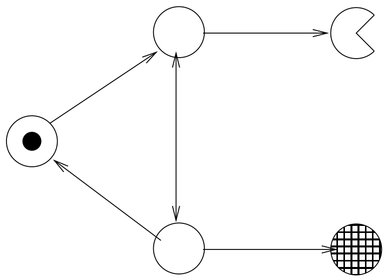

Figure 1: A simple maze.

bifurcations where the mouse chooses randomly, and with equal probabilty, one way among the possible ones. The corridors are one way, as indicated by the arrows, except from the one in the middle. Model the behaviour of the mouse in the maze as a DTMC and compute the probability with which it will find freedom and cheese.

Problem

We consider a simple probabilistic protocol designed for the *self-configuration* of IP network interfaces. We focus on the collision-avoiding mechanism of the protocol, whose purpose is to guarantee, with high probability, that a host entering the network gets a *unique address*. We assume the network to have a fixed configuration throughout the protocol.

When a fresh host joins the network, it selects at random an IP address among the $K = 65024$ possible addresses, *uniformly*. If there are m hosts in the network, then the probability of the address being already in use is $q = m/K$.

The new host tries to detect a possible collision by asking the other hosts whether they are using this address, and then waits for an answer for a limited amount of time. The new host decides that he got a unique address when he receives no answer from another host.

However, the answer can be lost, or just delayed, which can happen with probability p. To take this possibility into account, the new host repeats the question n times. Whenever an answer is received in time, the collission is detected and the new host restarts the protocol by selecting an address at random again. Only if after n questions (or probes) no answer arrived, the new host will join the network.

1. Model the protocol as a DTMC for $n = 2$ and $n = 3$.

2. Specify the property: *the host selects a new address with probability at least* 0.99.

3. Derive the corresponding system of linear equations, and compute the probability of selecting a new address, in the parameters p and q.

4. What happens when we increase n? Why does the protocol suggest n be 3 or 4 and not more?

5. Model and analyse the protocol with PRISM, and compare your results to those obtained with the tool.