CM0604 Probabilistic Models - Lab Work

Michael Brockway

November 17, 2016

These four lab sessions are to complement the four lectures on probabilistic models, model checking, PCTL logis and PRISM.

Your main effort will be in becoming familiar with the PRISM modelling language and a the PRISM model checking tool.

1 First Lab: Introduction

Visit the PRISM web pages: `www.prismmodelchecker.org`. Read the home page, which gives a good overview.

You can obtain your own copy of PRISM from here. A 64-bit linux version is installed in the machines in F1, in /usr/prism.

The Manual tab takes you to an 80-odd page manual which is quite a straightforward introduction to both the concepts and the practicalities – you should read the 'chapter' on the PRISM language soon: *start browsing now*.

The Tutorial tab takes you to a 6-part series of tutorial sessons with the PRISM model checking tool. Part 1 explores the die-throw simulation with coin tosses which we discussed in the lecture.

The Documentation tab provides additional links to the manual and the tutorial. There is also a link to a FAQ page and link to a page of links to lectures on the theory underlying probabilisitic model checking. The link

’Probabilistic model checking’ leads to some lectures by Dave Parker - the first 7 provide additional detail to our lectures, for your interest.

There is a Publications tab and a Case Studies tab providing links to reports of interesting research and application, for your interest.

Task Work through Part 1 of the tutorial now.

Task Review the content of the introductory lecture and your work through the practical tutorial session you have just completed, in view of each other.

- Does the PRISM model the die-throw simulation make sense? Can you explain how it works?
- Can to explain the PRISM Simulator output?
- Explain the content of the *properties file* `die.pctl`.
- Explain the difference between *simulation* and *verification*. How does *statistical model checking* fit into the picture?
- What is meant by the *expected termination time*? Explain how adding a ‘reward’ to the model and a check on a ‘reward property’ allows us to measure the expected termination time.

2 Second Lab: PRISM Models

The lecture on PRISM models should have clarified many of the questions arising from the previous lab.

Task Work through Part 2 of the tutorial now. This illustrates renaming and synchronisation among other things.

- Can you explain the algorithm being modelled?
- Explain the effect of the six ‘renaming’ module definitions.

- Explain the effect of the synchronised actions.
- Explain the 'stable state' property.

3 Third Lab: PRISM Modelling and Calculating Probabilities

Lecture 3 covered the theory of calculating probabilities of a path in a run of a DTMC.

A method was developed for calculating *reachability probabilities*: given a set T of states of a discrete-time Markov chain, to calculate for each state the probability of reaching T from the state. This method uses a fixed-point computation on vectors of probabilities and was developed in a java application `ProbReach`.

Task Download `ProbReach`.zip and unzip in a folder `ProbReach` within your personal folder.

- Open `ProbReach.java` in a text editor and compare the extended comment at the beginning with the theory developed in the lecture.
- Build the application (`javac *.java`) and run the graphical interface with the probabilitiy data for the die-throw: `java ProbReach dieThrowSim.txt`.
- You need to specify a subset T of states of the DTMC. The states t_2, t_3 are states number 8, 9 (looking in `dieThrowSim.txt`): so $T =$ 0...001100000000 = `0x300`. Just enter '300' in the text box: it expects hex format.
- Click the button. You will see an output: [0.3333333333, 0.6666666666, 0.0000000000, 0.3333333333, 1.0000000000, 0.0000000000, 0.0000000000, 0.0000000000, 1.0000000000, 1.0000000000, 0.0000000000, 0.0000000000, 0.0000000000]. This vector gives the probabilities of reaching $T = \{t_2, t_3\}$ from each of the states in the model. For instance, looking at

state 0, the probability is 0.3333333333 that a 2 or a 3 is eventually 'thrown'.

- Experiment with other state sets T.
- Experiment with *bounded* reachability: enter a positive integer in a 'Bound' box and click. The vector returns reports the probabilities of reaching T within that many steps.
- Compare your results with results from the PRISM model that you used in tutorial part 1.

Task Develop a PRISM model of the simple communicaton protocol described in the lecture and use it in PRISM to find reachability probabilities. Compare with results given by the `ProbReach` applicaton.

Task Develop a PRISM model of the Zeroconf protocol described in the lecture and use it in PRISM to find reachability probabilities. Compare with results given by the `ProbReach` applicaton. A java application `GenZeroconf` is provided in the zip you downloaded to generate Zeroconf model with varying parameter values (num probes, num nodes, probability of message loss). The invocation is `java GenZeroconf nProbes nNodes probMsgLoss > tgtFile`.

4 Fourth Lab: PRISM Properties

The lecture on Probabilistic temporal logic models explained the theory behind PRISM *properties*. It is also useful for you to have read the section of the manual on Property Specification (pp 23-40 in the PDF version).

Task Look at PRISM tutorial parts 3, 5, 6.