IN5100: Probabilistic Systems and Probabilistic and Statistical Model Checking

Peter Ölveczky

October 18, 2023

In general, we go light on the theory; try to understand the main concepts. Especially, no (non-trivial) background in statistics or probability theory is assumed.

Curriculum (pensum)

In some sense, the questions below and the lecture 7 and its exercises broadly defines the curriculum for this part. Officially, the curriculum consists of the following:

1. The paper

   > Gul Agha, José Meseguer, Koushik Sen: *PMaude: Rewrite-based Specification Language for Probabilistic Object Systems*, Electronic Notes in Theoretical Computer Science, Volume 153, Issue 2, 2006, Pages 213-239, ISSN 1571-0661, https://doi.org/10.1016/j.entcs.2005.10.040. Should be available from UiO at `https://www.sciencedirect.com/science/article/pii/S1571066106002672`.

2. The paper

   > Musab AlTurki and José Meseguer: *PVeStA: A Parallel Statistical Model Checking and Quantitative Analysis Tool*, in Proc. CALCO 2011, volume 6859 of Lecture Notes in Computer Science, Springer, 2011. `https://doi.org/10.1007/978-3-642-22944-2_28`.

3. Section 17.2 in my book *Designing Reliable Distributed Systems.*

4. The lecture, its slides, exercises, etc.

The following background papers, which are basically the papers introducing SMC, are also worth reading for some background:

- Håkan L. S. Younes and Reid G. Simmons: *Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling.* In Proc. CAV 2002, volume 2404 of Lecture Notes in Computer Science, Springer, 2002.

- Koushik Sen, Mahesh Viswanathan, and Gul Agha: *On Statistical Model Checking of Stochastic Systems.* In Proc. CAV 2005, volume 3576 of Lecture Notes in Computer Science, Springer, 2005.

What to Know

What you should take away from the lecture, and be able to answer well, are:

- What are probabilistic systems, and why are they needed?
- What are (simple discrete-time) Markov chains (DTMCs) and Markov decision processes (MDPs)?
- How does the curriculum suggest that we specify probabilistic systems in (an extension of?) rewriting logic? (I.e., what are probabilistic rewrite theories?)
- Specify simple probabilistic systems as probabilistic rewrite theories.
- What does it mean that a system is *fully probabilistic*?
- How can we *execute/simulate* such simple probabilistic rewrite theories in Maude, using randomized simulations where the new variables of the variables are sampled according to the given probability distribution?
- Have a rough idea of what is *probabilistic temporal logic* (such as PCTL) and *continuous stochastic logic* (CSL), and be able to write simple probabilistic requirements in these formalisms.
- What is probabilistic model checking? What is the main drawback of probabilistic model checking mentioned in the course?
- What is *statistical model checking* (SMC)?
  - what are the parameters α and δ (in the case of QuaTEx model checking with PVeStA)?
  - rough idea of how SMC works
- What are the main advantages of SMC discussed in the course? Why is it useful? What is the main drawback?
- Have a rough idea (no details needed) of what QuaTEx is.
- How can we turn the problem of statistically model checking a (simple) probabilistic property into statistically model checking a QuaTEx property? (That is, in what sense does QuaTEx SMC generalize SMC of probabilistic temporal logic properties?)
- What is the "main requirements" of systems so that they can be subjected to statistical model checking?
- How does the curriculum propose to make a somewhat large class of object-based Maude specifications fully probabilistic (with probability 1)?
- What is PVeStA?
- What are the differences between *probabilistic* and *statistical* model checking?