CS 512 Formal Methods, Spring 2018 — Instructor: Assaf Kfoury

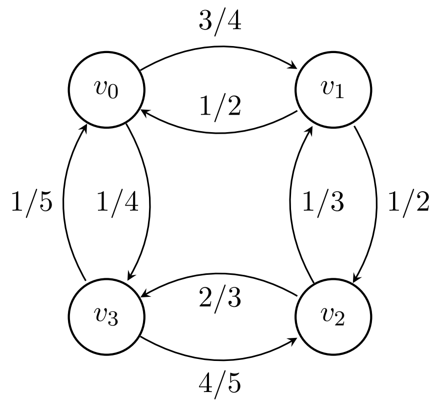
Lecture 23

April 12, 2018 — Hannah Catabia

(These lecture notes are **not** proofread and proof-checked by the instructor.)

1 Counterexamples in Probabilistic Model Checking

The following is a diagram of a discrete time Markov process:


$$\begin{tikzcd}
v_0 \arrow[r, bend left, "3/4"] \arrow[d, bend left, "1/4"] & v_1 \arrow[l, bend left, "1/2"] \arrow[d, bend left, "1/2"] \\
v_3 \arrow[u, bend left, "1/5"] \arrow[r, bend right, "4/5"'] & v_2 \arrow[l, bend right, "2/3"'] \arrow[u, bend left, "1/3"]
\end{tikzcd}$$

1.1 Stochastic matrices and discrete-time Markov processes

The transition probabilities of a Markov process may be represented in a matrix, as in this matrix for the process above:

$$A = \begin{bmatrix} 0 & \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ \frac{1}{5} & 0 & \frac{4}{5} & 0 \end{bmatrix}$$

Notice that A is an $n \times n$ stochastic matrix. A stochastic matrix is one in which the entries in each row sum up to 1. (A is not, however, a doubly-stochastic matrix, which is a matrix in which all the rows *and* columns sum to 1.)

Here are a few useful facts about discrete time Markov processes represented as stochastic matrices:

- If A is a $n \times n$ stochastic matrix, then A^k (A multiplied with itself k times) is also a $n \times n$ stochastic matrix.

- Let $q(i)$ be a vector with n entries that represents the probability distribution of being in each of the n states at time step i. If $q(0) = \{q_{0,1}, q_{0,2}, \dots, q_{0,n}\}$ is an initial distribution, then the probability distribution at time step t is:

$$q(t) = q(0) * A^t$$

- As k becomes very large, $q(t)$ converges to a unique probability distribution $\tilde{q}$, also called a *stationary distribution*:

$$\tilde{q} = \tilde{q} * A$$

If you wish to solve a problem involving matrix multiplication on the homework, there are many sources available, including MATLAB (and WolframAlpha).

1.2 Methods for analyzing discrete-time Markove processes

There are three main methods to find counterexamples with discrete-time Markov processes:

1. **Multiplying matrices**: This was covered above, but multiplying matrices is computationally expensive. Also, predicting how many times we need to multiply the matrix to arrrive at a stationary distribution is a difficult problem.

2. **Linear algebra**: In the last lecture, we covered a method of turning probability distributions into a system of linear equations. This method is computationally faster than multiplying matrices.

3. **Another method**: We can use an algorithm presented in lecture to transfrom the graph of the Markov process, and then we can use Dijkstra, Bellman Ford, or another path algorithm to find the shortest path on it. That path will represent our conterexample. Since the algorithm is described in Handout 22, p.12, I won't bother recreating it here. However, here are couple of useful points about this algorithm from lecture:

   (a) We convert the edge weights into logarithms so we can add them, which is less computationally expensive than multiplying them. (This follows from the fact that: $\log(2^a * 2^b) = \log(2^a) + \log(2^b)$.)

   (b) This algorithm can be performed on graphs with or without cycles. However, if you are working with a cyclic graph, be wary of which shortest path algorithm you choose to implement afterwards. For instance, Dijkstra algorithm's runs into problems on graphs that are not DAGs (directed acyclic graphs).

2 Modal Logic

We spent the last part of lecture introducing the final topic of the semester: modal logic.[1]

In modal logic, we have Kripke models (defined in Handout 23, p. 7). Kripke structures were developed by Saul Kripke in the 1960s, and they are essentially transition systems used in a different way. A Kripke model $\mathcal{M} = (W, R, L)$ is a triple containing:

- W, the set of all possible worlds
- $R \subseteq W \times W$, a binary accessibility relation

[1]Please note that although modal logic will not be covered on Homework 11, it will be covered on the exam.

- $L : W \to \mathcal{P}(AP)$, a labeling function

Modal logic has no probabilities, but it can have non-determinism. Like the logics we've looked at before in the course, modal logic defines syntax inductively and the formal semantics are syntax-directed. However, it uses a different symbol to show syntactic derivability. So far in this class, we're used to seeing:

- For syntax: $A \vdash B$, which means that B may formally be derived from A using proof rules in the logic, and which is read as "A deduces B"
- For semantics: $A \vDash B$, which means that A semantically implies B, and which is read as "A models B"

Modal logic uses the same symbol for semantic implication, but it uses a different symbol for syntactic derivability:

- For syntax: $A \Vdash B$, which means that B may be formally derived from A using proof rules in modal logic, and which is read as "A forces B"
- For semantics: $A \vDash B$, which means that A semantically implies B, and which is read as "A models B"