Model Checking Meets Probability: A Gentle Introduction

Joost-Pieter KATOEN [a,b,1]
[a] *Software Modeling and Verification Group, RWTH Aachen University, Germany*
[b] *Formal Methods and Tools Group, University of Twente, the Netherlands*

Abstract. This paper considers fully probabilistic system models. Each transition is quantified with a probability—its likelihood of occurrence. Properties are expressed as automata that either accept or reject system runs. The central question is to determine the fraction of accepted system runs. We also consider probabilistic timed system models. Their properties are timed automata that accept timed runs iff all timing constraints resent in the automaton are met; otherwise it rejects. The central question is to determine the fraction of accepted timed system runs.

Keywords. Markov chain, automata theory, linear equation system, integral equation system, transient probabilities.

1. Prologue

Probabilities are widely applicable. In system performance and dependability evaluation they are used to quantify job arrival processes or to indicate random delays. In reliability analysis, they are vital to quantify processor failures, message loss, and the like. In distributed computing, randomization is used to break the symmetry between identically behaving processes. This provides a viable workaround of the impossibility result stating that reaching consensus in an asynchronous distributed system is infeasible using deterministic computation. Finally, thanks to probabilities certain decision problems become decidable. For example, the question whether a given configuration is repeatedly reachable in a lossy channel system—a set of finite-state automata communicating via unbounded lossy channels that can nondeterministically loose messages—is undecidable. If the loss probability is quantified, however, repeated reachability becomes decidable. More precisely, the question whether a configuration is repeatedly reachable almost surely is decidable.

This chapter focuses on the verification of Markov chains, a popular probabilistic system model. They appear in two forms: discrete-time and continuous-time Markov chains. We cover both. Extensions of such models with non-determinism, or with real-valued clocks (as in timed automata) do exist, but are not covered. We thus assume that the behaviour of real-life systems is modelled by a Markov chain. The central problem that we will treat is the computation of reachability probabilities, i.e., what is the prob-

[1] Supported by the EU-projects MoVeS, CARP and SENSATION, as well as the DFG-NWO bilateral project ROCKS and the EU FP7 MEALS project.

ability to reach a goal state from a given start state? In the continuous-time setting, we generalise this by imposing a deadline on reaching the goal state. We will show that these problems are at the heart of automata-based model checking of Markov chains and can be tackled by solving equation systems. In that setting, we use an automaton, e.g., a traditional finite-state, a Büchi, or a timed automaton as "observer" of the system. This observer considers runs of the system, and either accepts or rejects them. The key question is how to determine the acceptance probability, i.e., the fraction of runs that are accepted. It is shown that this all reduces to (timed) reachability events.

Organization of this chapter. Part 1 (Discrete time). We start off by introducing discrete-time Markov chains. Then we argue that reachability probabilities in finite Markov chains can be obtained by solving a linear equation system. Subsequently, we show how acceptance probabilities of automata accepting infinite runs of a Markov chain can be reduced to a reachability problem in a product Markov chain. This provides a basis for verifying whether a Markov chain satisfies an LTL formula with a certain probability.

Part 2 (Continuous time). Assigning a random sojourn time to each state in a discrete-time Markov chain yields a continuous-time Markov chain. The sojourn times are governed by exponential distributions. We first discuss some elementary properties of negative exponential distributions. Subsequently we discuss the semantics of continuous-time Markov chains and how to determine the probability state distribution after a given amount of time (called transient probabilities). Then we argue that timed reachability probabilities in finite continuous-time Markov chains can be obtained by solving an integral equation system. Finally, we show how acceptance probabilities of deterministic timed automata accepting finite timed runs of a Markov chain can be reduced to solving several timed reachability problems in a product Markov chain.

Disclaimer: rather than providing formal definitions of all notions and concepts we use, we rather explain most issues at an informal level, thus providing a gentle introduction into the topic. References to the literature are provided where the interested reader can find a more theoretical treatment. Readers are expected to be familiar with the main principles of model checking.

2. Discrete-Time Markov Chains

A discrete-time Markov chain (DTMC) is Kripke structure in which all transitions are equipped with a probability. The example DTMC on the right [2] models the simulation of a six-sided die by a fair coin, due to Knuth and Yao [KY76]. The model contains 13 states with initial state s_0. In s_0, a coin is flipped. Depending on the outcome of the coin flip, the next state is either s_1 (on "tails") or s_2 (on "heads"). We de-

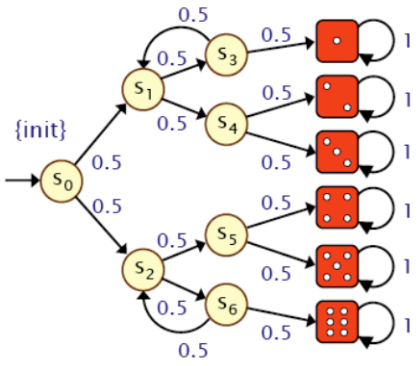

Figure 1.: Knuth-Yao algorithm for simulating a six-sided die by repeatedly tossing a fair coin.

[2] Thanks to Dave Parker for this figure.

note $\mathbf{P}(s_0, s_1) = \frac{1}{2}$ and $\mathbf{P}(s_0, s_2) = \frac{1}{2}$. The coin is flipped until finally the outcome is between one and six.
These final states are equipped with a self-loop with probability one, so e.g., $\mathbf{P}(3, 3) = 1$. States are labelled with sets of atomic propositions; e.g., state s_0 is labelled with *init* whereas all other states are labelled with $\varnothing$ (not depicted). Due to the cycles between states s_2 and s_6 (and s_1 and s_3), it is not evident that repeatedly tossing a coin yields an adequate model of a six-sided die. Is the probability of each of the outcomes indeed $\frac{1}{6}$?

In order to answer this question, we first consider the notion of a *path* through the Markov chain. A path is an infinite traversal through the topological structure of the DTMC. Thus, e.g., $s_0\, s_2\, s_5\, 4\, 4\, 4\, \ldots$ is a path. (The infinite repetition of state 4 is also written as 4^ω.) But, $s_0\, s_1\, s_2\, s_5\, 4^\omega$ is not a path, as $\mathbf{P}(s_1, s_2) = 0$. This can easily be seen, as there is no transition between s_1 and s_2. As a next step, we'd like to define the probability of e.g., the set of paths that finally end up in state 4. This is not trivial, as a path (by definition) is infinite. Thus, the naive idea to just multiply the transition probabilities along a path yields a probability mass zero. The way around this, is to consider sets of paths that all share a common prefix. Such sets of paths are called *cylinder sets*. Formally, the cylinder set of a path fragment $s_0\, s_1 \ldots s_n$ is the set of paths in the DTMC at hand that all have $s_0\, s_1 \ldots s_n$ as prefix. For example, the cylinder set of $s_0\, s_2\, s_5$ is the set of infinite paths consisting of $s_0\, s_2\, s_5\, 4^\omega$ and $s_0\, s_2\, s_5\, 5^\omega$. We now define the probability of a cylinder set C of $s_0\, s_1 \ldots s_n$, denoted $Pr(C)$, simply as $\mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)$. In case $n{=}0$, this probability is one. Thus, the probability of $\{\, s_0\, s_2\, s_5\, 4^\omega, s_0\, s_2\, s_5\, 5^\omega \,\}$ becomes $\mathbf{P}(s_0, s_2) \cdot \mathbf{P}(s_2, s_5)$ which equals $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. (Technically speaking, the σ-algebra on infinite paths in a DTMC is generated using cylinder sets.) Any set of paths that can be written as the complement and/or countable union of cylinder sets is now measurable. For instance, the probability of the set $C_1 \cup \overline{C_2}$, with C_1 and C_2 being cylinder sets, is $Pr(C_1) + (1 - Pr(C_2))$. This provides the basis for considering reachability probabilities.

Probabilities of sets of infinite paths are defined using cylinder sets.

3. Reachability Probabilities

Recall that we are interested in checking whether the usage of a fair coin to simulate a six-sided die is correct. This amounts to check whether the probability of any outcome is exactly $\frac{1}{6}$. We denote the set of paths that at some point reach a state in G, where G denotes the set of goal states, as $\Diamond G$. For our running example we have that $\Diamond\, 4$ contains e.g., the paths $s_0\, s_2\, s_5\, 4^\omega$ and $s_0\, s_2\, s_6\, s_2\, s_5\, 4^\omega$, and so forth. Written as a kind of regular expression we have

$$\Diamond\, 4 = \bigcup_{k \in \mathbb{N}} s_0\, (s_2\, s_6)^k\, s_2\, s_5\, 4^\omega$$

where $(s_2\, s_6)^k$ denotes that the cycle between s_2 and s_6 is taken k times. Given this characterization of the set $\Diamond 4$ we obtain:

$$Pr(\Diamond 4) = \sum_{s_0 \ldots s_n \in (S \setminus 4)^* 4} \mathbf{P}(s_0 \ldots s_n)$$

Here S denotes the set of states in the Markov chain at hand. This yields $\mathbf{P}(s_0 s_2 s_5 4) + \mathbf{P}(s_0 s_2 s_6 s_2 s_5 4) + \ldots\ldots$, or written shortly:

$$\sum_{k=0}^{\infty} \mathbf{P}(s_0\, (s_2\, s_6)^k\, s_2\, s_5 4)$$

This can be simplified to $\frac{1}{8} \cdot \sum_{k=0}^{\infty} (\frac{1}{4})^k$, as traversing the cycle once takes probability $\frac{1}{4}$, whereas reaching the cycle, and from there reaching 4 takes probability $\frac{1}{8}$. This yields a geometric series:

$$\sum_{k=0}^{\infty} \mathbf{P}(s_0 (s_2 s_6)^k s_2 s_5 4) \;=\; \frac{1}{8} \cdot \frac{1}{1 - \frac{1}{4}} \;=\; \frac{1}{8} \cdot \frac{4}{3} \;=\; \frac{1}{6}.$$

In a similar way, one can check that any outcome 1 through 6 of our Markov chain each has probability $\frac{1}{6}$.

This raises the question whether the reachability event $\Diamond\, G$ can be assigned a probability. Yes, this turns out to be true, as the set of paths that each hit a state in G at some point, can be written as a countable union of cylinder sets. The finite path fragments that need to be considered are all of the form $\overline{G}^*\, G$, i.e., path fragments that end once they reach a state in G. Technically speaking, this means that the event $\Diamond\, G$ is measurable. This holds for every DTMC with a countable state space. (For uncountable state spaces, one can establish a similar result but this goes beyond the scope of this chapter.) Other events can be defined in a similar way. For instance, $\Box G$ is the set of paths that only visit states in G. Nesting $\Box$ and $\Diamond$ yields the events $\Diamond\Box G$ —from some point on, only states in G are visited— and $\Box\Diamond G$ —infinitely often a state in G is visited. All these sets of paths are measurable. That is to say, $Pr(\Box G)$, $Pr(\Box\Diamond G)$ and $Pr(\Diamond\Box G)$ are all mathematically well-defined.

As a next step, we will discuss an algorithmic way to obtain $Pr(\Diamond G)$. More precisely, we want to determine the probability of the set of paths in $\Diamond G$, given that we start in some state s. We denote this as $Pr(s \models \Diamond\, G)$. Assuming that the Markov chain at hand has finitely many states, we let variable $x_s = Pr(s \models \Diamond G)$ for state s. Our aim is to obtain a value for x_s for every state s. The following recursive characterization will be helpful:

1. if G is not reachable from s, then $x_s = 0$
2. if $s \in G$, then $x_s = 1$
3. otherwise: $x_s \;=\; \underbrace{\sum_{t \notin G} \mathbf{P}(s,t) \cdot x_t}_{\text{reach } G \text{ via } t \in \overline{G}} \;+\; \underbrace{\sum_{u \in G} \mathbf{P}(s,u)}_{\text{reach } G \text{ in one step}}$

The first two cases are self-explanatory. The last case considers the situation in which G can be reached from s, but s does not belong to G. Then, in order to reach G, a transition

emanating s should be taken. There are two options. Either such transition leads to a state in G (second summand), or it leads to a state, t say, with $t \notin G$ (first summand).

Let us apply this to Knuth and Yao's example. Consider (again) the event $\Diamond 4$. Using the above characterisation we obtain:

1. $x_1 = x_2 = x_3 = x_5 = x_6 = x_{s_1} = x_{s_3} = x_{s_4} = 0$
2. $x_4 = 1$, and
3. for the states s_0, s_2, s_5 and s_6 that can all reach G:

$$x_{s_0} = \tfrac{1}{2}x_{s_1} + \tfrac{1}{2}x_{s_2}, \text{ and } x_{s_2} = \tfrac{1}{2}x_{s_5} + \tfrac{1}{2}x_{s_6},$$
$$x_{s_5} = \tfrac{1}{2}x_5 + \tfrac{1}{2}x_4, \text{ and } x_{s_6} = \tfrac{1}{2}x_{s_2} + \tfrac{1}{2}x_6$$

Gaussian elimination yields: $x_{s_5} = \frac{1}{2}$, $x_{s_2} = \frac{1}{3}$, $x_{s_6} = \frac{1}{6}$, and $\boxed{x_{s_0} = \frac{1}{6}}$. It is not surprising that we obtain the same result as before. Instead of using cylinder sets and geometric series, we however now obtained the reachability probability by solving a set of linear equations.

In fact, one can show that reachability probabilities are *unique* solutions of a given linear equation system. Such linear equation system is obtained in the following way. Let $S_?$ be the set of states that can reach G by > 0 steps. These states are thus exactly the states that correspond to the third case in the former recursive characterization. Let matrix $\mathbf{A} = \big(\mathbf{P}(s,t)\big)_{s,t \in S_?}$, the transition probabilities between the states in $S_?$. Let the vector $\mathbf{b} = \big(b_s\big)_{s \in S_?}$, the probabilities to reach some state G in a single step, i.e., $b_s = \sum_{u \in G} \mathbf{P}(s,u)$. Then: $\mathbf{x} = (x_s)_{s \in S_?}$ with $x_s = Pr(s \models \Diamond G)$ is the unique solution of:

$$\mathbf{x} \;=\; \mathbf{A}{\cdot}\mathbf{x} \;+\; \mathbf{b} \quad \text{or in short form} \quad (\mathbf{I} - \mathbf{A}){\cdot}\mathbf{x} \;=\; \mathbf{b}$$

where $\mathbf{I}$ is the identity matrix of cardinality $|S_?| \cdot |S_?|$.[3] For our running example, we have $S_? = \{\, s_0, s_2, s_5, s_6 \,\}$, and are interested in the unique solution of:

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{s_0} \\ x_{s_2} \\ x_{s_5} \\ x_{s_6} \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}$$

Gaussian elimination yields: $x_{s_5} = \frac{1}{2}$, $x_{s_2} = \frac{1}{3}$, $x_{s_6} = \frac{1}{6}$, and $\boxed{x_{s_0} = \frac{1}{6}}$.

To conclude: reachability probabilities can be obtained as the unique solution of a simple linear equation system. Any technique to solve such system either exactly (e.g., Gaussian elimination) or iteratively (e.g., the Power method) can be used to obtain reachability probabilities. This is nice, as these algorithms have a low time complexity — typically $\mathcal{O}(N^3)$ where $N = |S?|$— and are numerically stable.

Reachability probability = unique solution of a linear equation system.

[3] It follows that all Eigenvalues of matrix $\mathbf{A}$ are strictly less than one and that the infinite sum of powers of $\mathbf{A}$, that is, $\sum_I \mathbf{A}^i$ converges to the inverse of $\mathbf{I} - \mathbf{A}$. This yields that the matrix $\mathbf{I} - \mathbf{A}$ is non-singular and the linear equation system has a single solution.

Events of the form $\Diamond G$ are referred to as *unbounded* reachability events. This stems from the fact that it is not specified how many transitions (or: steps) are needed before reaching G. In contrast, *bounded* reachability events put an upper bound on the number of transitions that are allowed to reach G. So, for natural number k, the set $\Diamond^{\leqslant k} G$ denotes the set of paths that reach a state in G in at most k steps. For example, the path $s_0\, s_2\, s_5\, 4^\omega$ belongs to $\Diamond^{\leqslant 3}\, 4$, but does not belong to $\Diamond^{\leqslant 2}\, 4$. How can bounded reachability probabilities be determined? This turns out to be not that difficult. Our aim is to compute $Pr(\Diamond^{\leqslant k} G)$ in DTMC $\mathcal{D}$, say. Observe that once a path reaches G, then the remaining behaviour along that path is not important. Only the fact that it hits G at least once is of relevance. This suggests to make all states in G *absorbing*, i.e., replace all outgoing transitions from every state $s \in G$ by a self-loop with probability one. Denote the resulting Markov chain by $\mathcal{D}[G]$. The crucial observation is now that once a path reaches a state in G within k steps, that path will still be in that state in G after exactly k steps. That is:

$$\underbrace{Pr(s \models \Diamond^{\leqslant k} G)}_{\text{reachability in } \mathcal{D}} = \underbrace{Pr(s \models \Diamond^{=k} G)}_{\text{reachability in } \mathcal{D}[G]} = \underbrace{\mathbf{1}_s \cdot \mathbf{P}_G^k}_{\text{in } \mathcal{D}[G]} \quad \text{where} \quad \mathbf{P}^k = \underbrace{\mathbf{P} \cdot \ldots \cdot \mathbf{P}}_{k \text{ times}}.$$

Here the event $\Diamond^{=k} G$ denotes the set of infinite paths that exactly after k steps reach a state in G, and the vector $\mathbf{1}_s$ is a vector that equals one in state s and zero otherwise. The stochastic interpretation of $\mathbf{1}_s \cdot \mathbf{P}_G^k$ is the probability to be in a G-state after exactly k steps when starting from state s. This is also referred to as the *transient state distribution* of the Markov chain $\mathcal{D}[G]$ (when starting in state s). Summarizing,

Bounded reachability probability = transient probability distribution.

4. Observing Markov Chains by Finite Automata

In the following, we will discuss how the likelihood of events such as $\Diamond\Box G$ and $\Box\Diamond G$ can be obtained. The key result will be that these probabilities can be reduced to reachability probabilities. In order to understand this, we consider the *long-run* behaviour of a finite DTMC. Suppose we start in state s. This corresponds to a probability distribution $\mathbf{1}_s$ which is one for s and zero otherwise. That is, $\mathbf{1}_s(s) = 1$ and $\mathbf{1}(s') = 0$ whenever $s' \neq s$. We now consider the probability distribution after taking a single step. This corresponds to $\mathbf{1}_s \cdot \mathbf{P}$. Then take a next step. The distribution among the states is then given by $\mathbf{1}_s \cdot \mathbf{P}^2$. After three steps we obtain $\mathbf{1}_s \cdot \mathbf{P}^3$. And so forth. This raises the question: which states have a positive probability on the long run? Stated otherwise, which states occur infinitely often in a given path? It turns out that these states belong to bottom strongly connected components (bottom SCCs, for short) of the Markov chain. In a strongly connected graph, each pair of states is mutually reachable. Such graph is an SCC whenever it is not properly contained in another strongly connect graph. A *bottom* SCC T is an SCC that contains no transition that leads to a state outside T. The set $\{\, s_1, s_3 \,\}$ of our running example DTMC is an SCC. Due to the transitions $s_1 \to s_4$ and $s_3 \to 1$, it is not a bottom SCC. The set $\{\, 4 \,\}$ however is a bottom SCC. The set of bottom SCCs in the running example contains $\{\, 1 \,\}$, $\{\, 2 \,\}$, $\{\, 3 \,\}$, $\{\, 4 \,\}$, $\{\, 5 \,\}$, and $\{\, 6 \,\}$. The intuition is that the probability to stay within an SCC infinitely long is zero, as the

probability to eventually take one of its transitions that leave the SCC is one. However, for a bottom SCC there is no possibility to leave it, i.e., a bottom SCC acts like a sink. Once a bottom SCC is reached, each of its states will be visited infinitely often, as all states are mutually reachable. Thus, almost surely any finite DTMC eventually reaches a bottom SCC and visits all its states infinitely often.

On the long run, each path in a finite Markov chain ends in a bottom SCC.

Now consider the event $\Diamond\,\Box G$ with G a set of states. What is the probability of the set of paths starting in s that only visits G-states from some point on? As we just have argued, on the long run, the Markov chain will end up in a bottom SCC. For a bottom SCC only containing G-states, it is ensured that only G-states will be reached once this bottom SCC has been reached. Stated formally, $\mathit{Pr}(s \models \Diamond\,\Box G) = 1$ for any state s in such bottom SCC. This does not hold for bottom SCCs that contain at least one state in $\overline{G}$—in such bottom SCCs every now and then a state in $\overline{G}$ will be visited, so $\mathit{Pr}(s \models \Diamond\,\Box G) < 1$. Thus, $\mathit{Pr}(s \models \Diamond\,\Box G)$ equals the probability to reach from s a bottom SCC that only contains states that are in G. In algorithmic terms, this means that one first determines the bottom SCCs of the DTMC $\mathcal{D}$. This can be done using a slight adaptation of Tarjan's algorithm to find SCCs in directed graphs. Its worst-case time complexity is linear in the number of states and the number of transitions in DTMC $\mathcal{D}$. Then: $\mathit{Pr}(s \models \Diamond\Box G) \;=\; \mathit{Pr}(s \models \Diamond U)$ where U is the union of all bottom SCCs T in $\mathcal{D}$ with $T \subseteq G$. Putting things together, this means that $\mathit{Pr}(s \models \Diamond\Box G)$ can be obtained by solving a linear equation system in which the goal states are the states in the set U. As explained above, U is determined by a graph analysis.

$\mathit{Pr}(\Diamond\,\Box G)$ = reachability probability of bottom SCCs in which all states are in G.

As an example, let $G = \{\,1, 3, 5\,\}$, that is, all possible odd outcomes of the mimicked die. We obtain that $U = \{\,\{\,1\,\}, \{\,3\,\}, \{\,5\,\}\,\}$, and as $\mathit{Pr}(s_0 \models \Diamond\, U) = \frac{1}{2}$, it follows that $\mathit{Pr}(s_0 \models \Diamond\,\Box G) = \frac{1}{2}$. For $\Box\Diamond G$, the bottom SCCs that contain at least a state in G are of importance. Then we obtain: $\mathit{Pr}(s \models \Box\Diamond G) \;=\; \mathit{Pr}(s \models \Diamond U)$ where U is the union of all bottom SCCs T in DTMC $\mathcal{D}$ with $T \cap G \neq \varnothing$.

$\mathit{Pr}(\Box\Diamond\, G)$ = reachability probability of bottom SCCs in which some state is in G.

The sets $\Diamond\,\Box G$ and $\Box\Diamond\, G$ are ω-regular. As we have seen, their likelihood can be reduced to reachability probabilities. This raises the question whether this applies to arbitrary ω-regular properties. Such as $\Diamond\,\Box G \wedge \Box\Diamond\, P$: what is the probability of the paths that from some point on only visit good states (i.e., G), while visiting premium states (i.e., P) infinitely often? In the following, we will see that the probability of all ω-regular properties in finite Markov chains can be reduced to reachability probabilities. The quantitative analysis of Markov chains against automata specifications goes back to Courcoubetis and Yannakakis [CY95]. The approach is to represent the ω-regular property by a deterministic finite automaton. In contrast to the familiar finite-state automata that accept finite words, we need automata that accept *infinite* words. This stems from the fact that (in general) one can only decide whether an ω-regular property holds on the basis of an infinite computation. (When restricting the properties to, e.g., safety properties whose bad prefixes constitute a regular language, automata accepting finite words would do.

The approach then is however basically the same.) The intuition is that these automata will act as observers of the Markov chain. They observe paths of the DTMC, and decide which paths are accepting (those that satisfy the ω-regular property of interest) and which ones are not. The final objective is to determine the fraction of paths that are accepted by the automaton. Or, stated differently, what is the likelihood that the Markov chain exhibits a behaviour that satisfies the ω-regular property? For the running example, such property could e.g., be to restrict the number of "tails" outcomes of the coin flip, say maximally five times "tails", on obtaining the final outcome 1 or 3. To simplify the property somewhat, we assume that the first coin flip has outcome "tails". The resulting property can be modelled by the following automaton: This automaton takes a first tran-

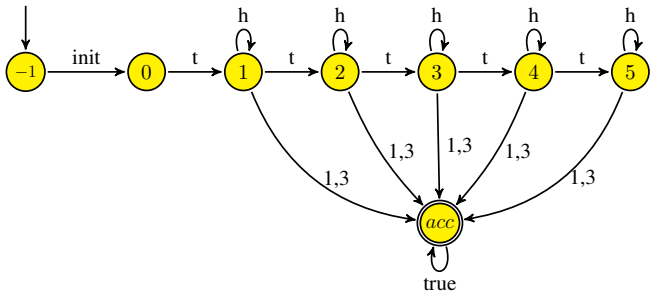

Figure 2. An automaton specifying that after an initialization and an initial tails outcome, the outcome 1 or 3 is eventually obtained provided that in total no more than five times tails is thrown.

sition (from state -1 to state 0) when the coin flipping process starts. After a first "tails" outcome (indicated by transition label t), it moves to state 1, in which it either waits for a "heads" (while staying in state 1, indicated by transition label h), or a "tails" (moving to state 2). So, state i is occupied when i times the outcome of the coin flip has been "tails". On the outcome of the coin flipping process being 1 or 3, the automaton moves to the accepting state, in which it will stay *ad infinitum*. As it will stay in that state regardless of the input symbol, we label the self-loop with true. (Rather than drawing two transitions, labelled with 1 and 3, respectively, we indicate a single transition labelled with 1,3.) If in a state an input is obtained for which there is no outgoing transition, the automaton moves to an error state in which the automaton rejects. In order not to blur the picture, this error state is not depicted. For instance, in state 5, an outcome "tails" will lead to the error state as the total number of "tails" outcomes exceeds five.

Recall that states in DTMCs —like in Kripke structures— are labeled with sets of atomic propositions. These labels have not been of importance so far, but now are of relevance. The *trace* of a path $s_0\, s_1\, s_2 \ldots$ is the infinite sequence $L(s_0)\, L(s_1)\, L(s_2) \ldots$, where $L(s_i)$ is the set of atomic propositions associated with state s_i. A trace of a path is thus its element-wise projection onto the state-labelings. As an example, the trace of the path $\pi = s_0\, s_2\, s_6\, s_2\, s_5\, 4^\omega$ is $\mathit{trace}(\pi) = \{\mathit{init}\}\, \varnothing\, \varnothing\, \varnothing\, \varnothing\, \{\mathit{four}\}^\omega$, where it is assumed that the outcome 4 of the simulated die is labelled with the proposition *four*. The idea is now that traces obtained from paths of the Markov chain are fed into the automaton. That is, the observer automaton reads the traces as input words. This entails that the alphabet of the automaton are sets of atomic propositions (such as $L(s_i)$). We now need to fix when an observer automaton accepts a trace, and when it does not. This

is determined by the acceptance condition of the automaton. For reasons that become clear later on, we use deterministic Rabin automata (DRA) as observers. These automata have the same ingredients as finite-state automata, except for the acceptance condition. Their acceptance condition is as follows. Let $\{ (L_i, K_i) \mid 0 < i \leqslant m \}$ with L_i, K_i be a family of states of the DRA. This is thus a set of pairs, where each element of a pair is a set of states. A run of the automaton is now accepting whenever for some pair (L_j, K_j) the run visits all states in L_j only finitely often and visits some state in K_j infinitely often. Stated in terms of an LTL formula:

$$\bigvee_{0<i\leqslant m} (\Diamond \Box \neg L_i \wedge \Box \Diamond K_i)$$

where the proposition L_i holds in every state belonging to the set L_i, and similarly for the proposition K_i. For $\{ (\varnothing, F) \}$, a run is accepting if some state in F is visited infinitely often.[4] As $L = \varnothing$, there is no state constrained to be visited only finitely often. In contrast, for family $\{ (F, \{ q \}) \}$ a run is accepting if all states in F are only visited finitely often whereas state q is visited infinitely often. In case $q \in F$, this automaton thus does not have any accepting run, as q cannot be visited finitely and infinitely often at the same time. A family $\{ (Q, \varnothing) \}$ where Q is the set of states of the DRA, has no accepting run: each run has to visit some state infinitely often (as a DRA has only finitely many states), contradicting that each state must only be visited finitely often to be accepting (due to the first element of the pair on the set being Q.). As an example, for family of sets $\{ \{ q_0 \}, \{ q_1 \} \}$, the DRA depicted just above accepts all traces that satisfy $\Diamond \Box a$. Readers familiar with Büchi automata recall that there does not exist a deterministic Büchi automaton for $\Diamond \Box a$; indeed such automata are strictly less expressive than DRA. DRA are however as expressive as ω-regular properties.

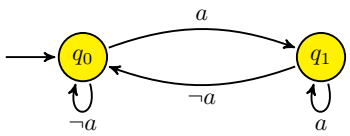

Figure 3.: A DRA for the property $\Diamond \Box a$.

A language of infinite words is ω-regular iff there exists a DRA that generates it.

The same theorem holds for *non*deterministic Büchi automata, but as we like to stay within the realm of deterministic automata[5], we prefer DRA here. To summarize, the setting is as follows: the *possible* system behaviour is modelled by a finite-state DTMC $\mathcal{D}$, and the *required* system behaviour is given as a DRA, $\mathcal{A}$, say. The problem of interest now is to determine the fraction of paths of the Markov chain that are accepted by the DRA. Stated differently, what is the probability mass of the set of paths generated by $\mathcal{D}$ which are accepted by $\mathcal{A}$? We denote this probability by $Pr(\mathcal{D} \models \mathcal{A})$. Formally, we have

$$Pr(\mathcal{D} \models \mathcal{A}) = Pr\{ \pi \text{ is a path of DTMC } \mathcal{D} \mid \mathit{trace}(\pi) \text{ is accepted by DRA } \mathcal{A} \}.$$

[4] This coincides with the acceptance condition of a (deterministic) Büchi automaton.

[5] There is a way to also deal with nondeterministic automata [CY95], but this falls outside the scope of this tutorial.

Let us first remark that this probability is well-defined, as the set of paths accepted by a DRA can be defined in terms of cylinder sets. As a next step, we consider the *product* of a Markov chain and an automaton. The intuition behind this stems from the product construction for ordinary finite-state automata. Recall that for two finite-state automata $\mathcal{A}_1$ and $\mathcal{A}_2$, say, the synchronous product $\mathcal{A}_1 \otimes \mathcal{A}_2$ recognizes all finite words that are accepted by both automata. If $\mathcal{A}_1$ thus models all possible behaviours and $\mathcal{A}_2$ all required behaviours, $\mathcal{A}_1 \otimes \mathcal{A}_2$ is a model exhibiting all possible, required behaviours. The "only" difference now is that we do not have two automata, but a DTMC $\mathcal{D}$ and a DRA $\mathcal{A}$. The product $\mathcal{D} \otimes \mathcal{A}$ is a DTMC with the Cartesian product $S \times Q$ as state space, where S is the finite state space of $\mathcal{D}$ and Q the state space of $\mathcal{A}$. A transition $s \to s'$ in DTMC $\mathcal{D}$ is combined with $q \xrightarrow{L} q'$ in DRA $\mathcal{A}$, where L is an input symbol of the DRA, i.e., a set of atomic propositions, whenever L is the labelling of the target state of the transition $s \to s'$, that is, $L = L(s')$. Thus, in this case there is a transition from state $\langle s, q \rangle$ to $\langle s', q' \rangle$. Given that the DRA is deterministic, this is the only way in which the automaton can match the transition of the DTMC. The probability of this transition is $\mathbf{P}(s, s')$. The initial state of the DTMC $\mathcal{D} \otimes \mathcal{A}$ is $\langle s_0, q_1 \rangle$ where $q_0 \xrightarrow{L} q_1$ and $L = L(s_0)$. It follows directly from this construction that for each path $\pi = s_0\, s_1\, s_2 \ldots$ in DTMC $\mathcal{D}$ there exists a unique run $q_0\, q_1\, q_2 \ldots$ in DRA $\mathcal{A}$ for $\mathit{trace}(\pi) = L(s_0)\, L(s_1)\, L(s_2) \ldots$. Their combined path in $\mathcal{D} \otimes \mathcal{A}$ is then $\langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \langle s_2, q_3 \rangle \ldots$. Reversely, for every path in the product, there is a unique corresponding path in DTMC $\mathcal{D}$ and a unique run in $\mathcal{A}$.

Let us return to our running example. We slightly adapt the DTMC for Knuth and Yao's six-sided die example by labelling states with t if they are reached after throwing "tails" and h after reaching via "heads". The same holds for the states modelling the final outcomes. In fact, the self-loops at these states are unfolded once, yielding a state i labelled with t or h, and one labelled with i. So, for instance if in state s_3 "heads" is thrown, the resulting state is 1, and this state is labelled with t. Taking the self-loop in state 1 once leads to a copy of state 1 that is labelled with 1. That state has a self-loop only. The product of DTMC $\mathcal{D}$ and the automaton for the property that the outcome must be 1 or 3 and be obtained after first throwing "tails" and throwing at most five times "tails" in total is given below:

In this product automaton, we considered all possible behaviours of the automaton, including the ones in which the automaton rejects (these were not depicted before in the property automaton). These behaviours all end in state where the second element is err. For instance, the state $(3, \mathit{err})$ is reached when more than five times the flip outcome has been "tails", whereas the state $(2, \mathit{err})$ is reached when the final outcome of the coin flipping process is two. In both cases, the property that we are considering is violated.

The final ingredient that we need to consider are the bottom SCCs in the product $\mathcal{D} \otimes \mathcal{A}$. As DRA $\mathcal{A}$ accepts infinite words, we are interested in the long run behaviour of the product Markov chain. As the product $\mathcal{D} \otimes \mathcal{A}$ has finitely many states, we know that on the long run, this DTMC will end up in one of its bottom SCCs. Which ones are now "acceptable" for the DRA? In order to understand this, let us recall the acceptance criterion of DRA $\mathcal{A}$. Let $\{\, (L_i, K_i) \mid 0 < i \leqslant m \,\}$ with L_i, K_i be a family of states of the DRA. A run is accepting whenever for some pair (L_j, K_j) the run visits all states in L_j only finitely often and visits some state in K_j infinitely often. An "acceptable" bottom SCC T in the product $\mathcal{D} \otimes \mathcal{A}$ is now required for some $j \in \{\, 1, \ldots, m \,\}$ to fulfill the following condition:

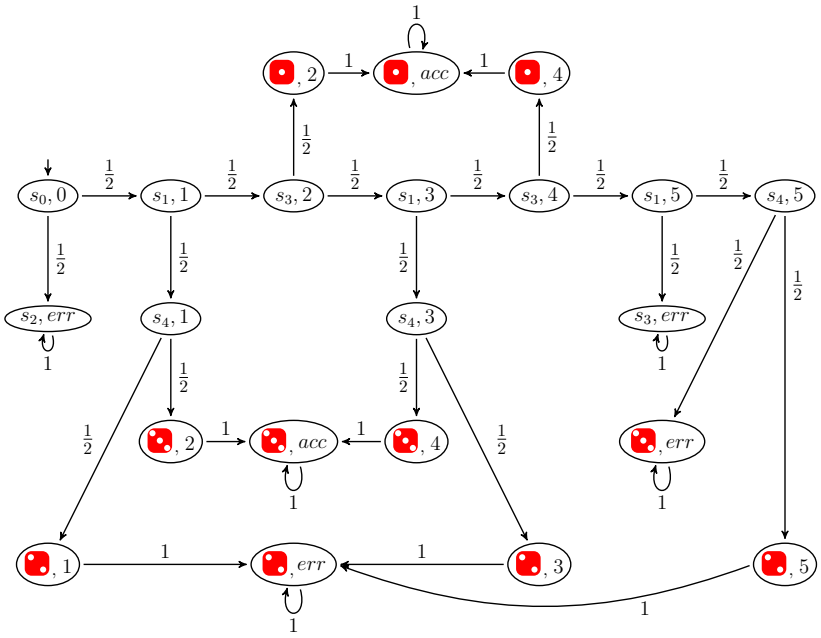

Figure 4. Product Markov chain obtained from Knuth-Yao's model for the six-sided die, and the property automaton restricting the number of tails outcomes.

$$T \cap (S \times L_j) = \varnothing \quad \text{and} \quad T \cap (S \times K_j) \neq \varnothing.$$

Stated in words, T contains no states that correspond to automaton states in L_j and at least one state corresponding to an automaton state in K_j. Any path of the product DTMC $\mathcal{D} \otimes \mathcal{A}$ —and thus the corresponding path in the Markov chain $\mathcal{D}$— that reaches such an "acceptable" bottom SCC T, is guaranteed to fulfill the accepting condition of the DRA. Before reaching T, states in L_j and K_j can be visited arbitrarily often. As such path is finite, L_j is only visited finitely often. Within the bottom SCC T, it is guaranteed that no L_j state will ever be visited, and infinitely often a K_j state is visited. Such paths thus contribute to $Pr(\mathcal{A} \models \mathcal{D})$, our measure of interest. Let U be the union of all "acceptable" bottom SCCs in the product Markov chain. We now obtain that:

$$Pr(\mathcal{D} \models \mathcal{A}) \;=\; Pr\big(\, \langle s_0, q_{s_0} \rangle \;\models\; \Diamond U \,\big) \quad \text{where} \quad q_0 \xrightarrow{L} q_{s_0} \text{ with } L = L(s_0).$$

The state $\langle s_0, q_{s_0} \rangle$ is the initial state in the product $\mathcal{D} \otimes \mathcal{A}$. This result has several important repercussions. First, the fraction of paths in DTMC $\mathcal{D}$ that are accepted by $\mathcal{A}$ can be obtained by computing the reachability probability of accepting bottom SCCs in the product $\mathcal{D} \otimes \mathcal{A}$. It is straightforward to determine "acceptable" bottom SCCs by a graph analysis. Reachability probabilities in $\mathcal{D} \otimes \mathcal{A}$ can be determined by solving a linear equation system. Variables in this equation system correspond to states in $S \times Q$. The resulting worst-case time complexity is polynomial in the size of $\mathcal{D}$ and $\mathcal{A}$. As DRA are as expressive as ω-regular properties, the above provides a recipe to determine the probability of satisfying an ω-regular property. In particular, as any LTL-formula is ω-regular, that is, the set of traces satisfying an LTL-formula is ω-regular, this implicitly provides

an algorithm to determine the probability of satisfying an LTL-formula. Here, it should be noted that the size of a DRA for a given LTL-formula can be double-exponential in the size of the LTL-formula.[6]

$Pr(\mathcal{D} \models \mathcal{A})$ = the reachability probability of "accepting" bottom SCCs in $\mathcal{D} \otimes \mathcal{A}$.

For our running example, we need to consider the "accepting" bottom SCCs in the product $\mathcal{D} \otimes \mathcal{A}$ where DRA $\mathcal{A}$ is the automaton depicted before with family of accepting sets $\{ (\varnothing, \{ q_{acc} \}) \}$. In fact, this DRA is a simple finite-state automaton that accepts once state q_{acc} has been reached. This product $\mathcal{D} \otimes \mathcal{A}$ has several bottom SCCs, but only two accepting bottom SCCs: one that contains the state $(1, q_{acc})$ and one that contains $(3, q_{acc})$. The probability to reach one of them equals $\frac{1}{8} + \frac{1}{8} + \frac{1}{32} + \frac{1}{32} = \frac{5}{16}$.

5. Negative Exponential Distributions

In this part of this tutorial paper, we will consider Markov chains in which the state residence time is random. More precisely, the amount of time spent in a state will be governed by a negative exponential distribution. Why exponential distributions, and not any other type of distribution, such as uniform, normal, or Weibull distributions? There are various arguments for this. For instance, exponential distributions are best when only mean values are known. Here, we will argue that it is natural continuous time extension of DTMCs. In order to explain this in a bit more detail, let random variable X_s be the number of epochs of DTMC $\mathcal{D}$ to stay in state s given that the current state is s. Then $Pr\{ X_s = 1 \} = 1 - \mathbf{P}(s,s)$, the probability to leave state s by taken an outgoing transition of s. In a similar way, we have that $Pr\{ X_s = 2 \} = \mathbf{P}(s,s) \cdot (1 - \mathbf{P}(s,s))$, i.e., the probability to take a self-loop in state s followed by leaving state s. By induction, it follows that

$$Pr\{ X_s = n \} \;=\; \mathbf{P}(s,s)^{n-1} \cdot (1 - \mathbf{P}(s,s)).$$

So, the state residence times in a DTMC obey a *geometric* distribution with parameter $\mathbf{P}(s,s)$. The expected number of time steps to stay in state s thus equals $\frac{1}{1-\mathbf{P}(s,s)}$. The plots below [7] depict the geometric distribution for different values of $p = \mathbf{P}(s,s)$ (left), and its cumulative distribution function (right). An important result is that a geometric distribution is the only discrete probability distribution that is memoryless. That is, for any geometrically distributed random variable X:

$$Pr\{X > k + m \mid X > m\} \;=\; Pr\{X > k\} \quad \text{for any} \quad k, m \in \mathbb{N}, k > 0$$

This is called the *memoryless* property, and X is a memoryless random variable.

Each memoryless discrete probability distribution is geometric.

[6] This can be reduced to single exponential procedure by a variant of the described procedure in which nondeterministic Büchi automata are used and determinized using the standard Rabin-Scott determinization algorithm for finite-state automata, before building the product Markov chain.

[7] Taken from wikipedia.

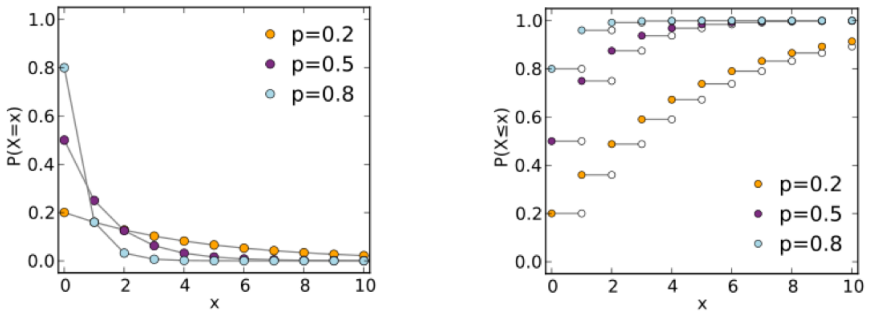

Figure 5. Probability mass function (left) and cumulative distribution function (right) of geometric distribution for various parameters p

We now consider negative exponential distributions. The cumulative distribution of continuous random variable X with rate $\lambda \in \mathbb{R}_{>0}$ is defined for $d \in \mathbb{R}_{\geqslant 0}$:

$$\int_0^d \lambda \cdot e^{-\lambda \cdot x}\, dx \;=\; [-e^{-\lambda \cdot x}]_0^d \;=\; 1 - e^{-\lambda \cdot d}.$$

The function $\lambda \cdot e^{-\lambda \cdot x}$ is also known as the density function of X. It follows by standard means that the expectation of X equals $\frac{1}{\lambda}$. The rate $\lambda \in \mathbb{R}_{>0}$ uniquely determines an exponential distribution. The following plots [8] show the density function for several values of the rate λ (left) and the cumulative distribution function (cdf): From the right plot, we

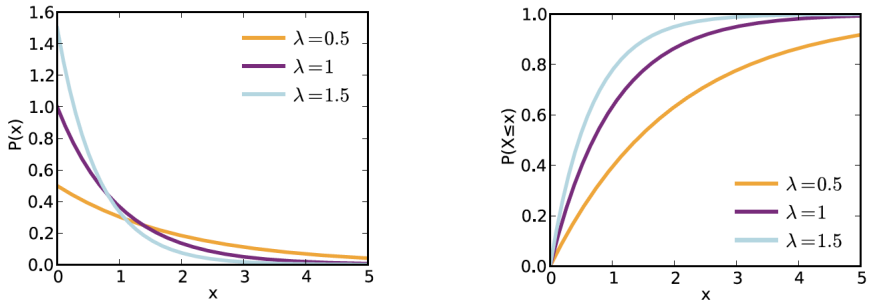

Figure 6. Probability density function (left) and cumulative distribution function (right) of exponential distribution for various rates λ

infer that the higher the rate λ, the faster the cdf approaches one. The reader is invited to observe the strong similarities between the shape of the plots for the exponential distribution and those for the geometric distribution. Indeed, the exponential distribution is the continuous counterpart of the geometric distribution. If we are about to consider a continuous version of DTMCs, it is thus very natural to take negative exponential distributions to govern state residence times. An important result is that a negative exponential distribution is the only continuous probability distribution that is memoryless. That is, for any exponentially distributed random variable X:

[8] Taken from wikipedia.

$$Pr\{X \leqslant k + m \mid X > m\} \;=\; Pr\{X \leqslant k\} \quad \text{for any} \quad k, m \in \mathbb{R}, k > 0$$

This is called the *memoryless* property, and X is a memoryless random variable.

Each memoryless continuous probability distribution is exponential.

Before introducting the model of continuous-time Markov chains (CTMCs, for short), we first discuss a couple of useful properties of exponential distributions. The first property of interest is that the class of exponential distributions is closed under minimum. In fact, the rate of the minimum of two exponential distributions is simply the sum of the rates of the individual distributions. That is to say, for independent, exponentially distributed random variables X and Y with rates $\lambda, \mu \in \mathbb{R}_{>0}$, the random variable $\min(X, Y)$ is exponentially distributed with rate $\lambda{+}\mu$. This generalizes to taking the minimum over an arbitrary number of exponential distributions. Thus for independent, exponentially distributed random variables $X_1, X_2, \ldots, X_n$ with rates $\lambda_1, \lambda_2, \ldots, \lambda_n \in \mathbb{R}_{>0}$, the random variable $\min(X_1, X_2, \ldots, X_n)$ is exponentially distributed with rate $\sum_{0<i\leqslant n} \lambda_i$.

The second property of interest is concerned with the probability that an exponential distribution is "less than" another one. It turns out that this is a simple expression in terms of the rates of the involved exponential distributions. More precisely, for independent, exponentially distributed random variables X and Y with rates $\lambda, \mu \in \mathbb{R}_{>0}$:

$$Pr\{X \;=\; \min(X, Y)\} \;=\; \frac{\lambda}{\lambda{+}\mu}.$$

Also this result generalizes to several exponentially distributed random variables. For independent, exponentially distributed random variables $X_1, X_2, \ldots, X_n$ with rates $\lambda_1, \lambda_2, \ldots, \lambda_n \in \mathbb{R}_{>0}$ it holds:

$$Pr\{X_i \;=\; \min(X_1, \ldots, X_n)\} \;=\; \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}.$$

These properties will be relevant later on when discussing the semantics of CTMCs.

6. Continuous-Time Markov Chains

A CTMC is now actually quite simple: it is an extension of a DTMC over state space S, with a function $r : S \to \mathbb{R}_{>0}$ that assigns to each state s the rate of the exponential distribution governing the residence time in s. Thus, the residence time in state s is exponentially distributed with rate $r(s)$. That is, the probability to wait maximally d time units in state s equals $1 - e^{-r(s)\cdot d}$. Phrased alternatively, the average residence time of state s is $\frac{1}{r(s)}$ time units. Thus, the higher the rate $r(s)$, the shorter the average residence time in s. The figure (left) below depicts a CTMC on four states with $r(0) = 25$, $r(1) = 4$, $r(2) = 2$ and $r(3) = 100$. As the rate of state 1 is twice the rate of state 2, its residence time is —on average— half the residence time in state 2. Stated differently, the frequency of taking some outgoing transition of state 1 is twice that for state 2 (on average). Similarly, on average the frequency of taking the self-loop at state 3 is 50 times the frequency

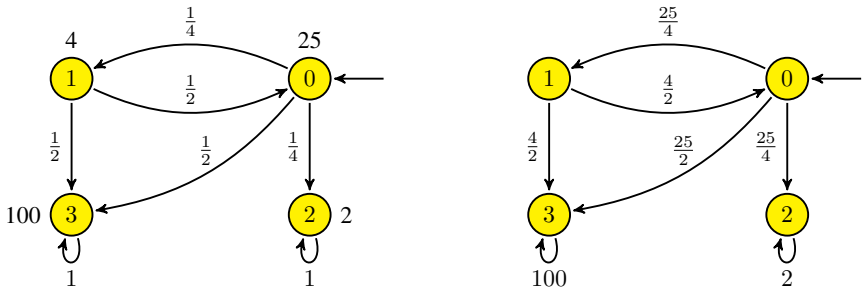

Figure 7. Two equivalent perspectives on continuous-time Markov chains.

at state 2. A completely equivalent view of a CTMC is to combine the transition probabilities given by function $\mathbf{P}$ and the rate-function r by defining $\mathbf{R}(s,s') = \mathbf{P}(s,s')\cdot r(s)$. This is called the transition rate between state s and s'. Figure 7(right) depicts the alternative view of the CTMC on the left. As said just above, both perspectives, or better said both definitions are fully equivalent. It is evident how to obtain the right figure from the left, namely by just applying $\mathbf{R}(s,s') = \mathbf{P}(s,s')\cdot r(s)$. Vice versa, one obtains the left figure from the right one by defining $r(s) = \sum_{s'} \mathbf{R}(s,s')$. This follows directly from the fact that $\sum_{s'} \mathbf{R}(s,s') = \sum_{s'} \mathbf{P}(s,s')\cdot r(s) = r(s)\cdot \sum_{s'} \mathbf{P}(s,s') = r(s)$. In addition it then immediately follows: $\mathbf{P}(s,s') = \frac{\mathbf{R}(s,s')}{r(s)}$. In the sequel, we will use (and change between) both equivalent definitions whenever convenient.

We are now in a position to explain the semantics of a CTMC. A simple way to do that is to associate to transition $s \to s'$ an exponentially distributed random variable $X_{s,s'}$ with rate $\mathbf{R}(s,s')$. The probability to go from state 0 to, say, state 2 is:

$$\begin{array}{c} \mathit{Pr}\{X_{0,2} \;=\; \min(X_{0,1}, X_{0,2}, X_{0,3})\} \\ = \\ \dfrac{\mathbf{R}(0,2)}{\mathbf{R}(0,1)+\mathbf{R}(0,2)+\mathbf{R}(0,3)} \;=\; \dfrac{\mathbf{R}(0,2)}{r(0)} \;=\; \mathbf{P}(0,2). \end{array}$$

This follows directly from the closure of exponential distributions under taking the minimum. Using another property of exponential distributions discussed before, it follows that the probability of staying at most t time units in state 0 is:

$$\mathit{Pr}\{\min(X_{0,1}, X_{0,2}, X_{0,3}) \leqslant t\} = 1 - e^{-(\mathbf{R}(0,1)+\mathbf{R}(0,2)+\mathbf{R}(0,3))\cdot t} \;=\; 1 - e^{-r(0)\cdot t}.$$

The operational interpretation of a CTMC is thus as follows. On entering a state s, the residence time is determined by an exponential distribution with rate $r(s)$. On leaving the state s, the probability to move to state s' is then given by $\mathbf{P}(s,s')$.

We conclude this section by providing an example from systems biology: enzyme-catalytic substrate conversion. This example shows a rather natural example of the usage of CTMCs. Enzyme kinetics investigates of how enzymes (E) bind substrates (S) and turn them into products (P). About a century ago, Henri considered enzyme reactions to take place in two stages. First, the enzyme binds to the substrate, forming the enzyme-substrate complex. This substrate binding phase catalyses a chemical reaction that releases the product. Enzymes can catalyse up to several millions of reactions per

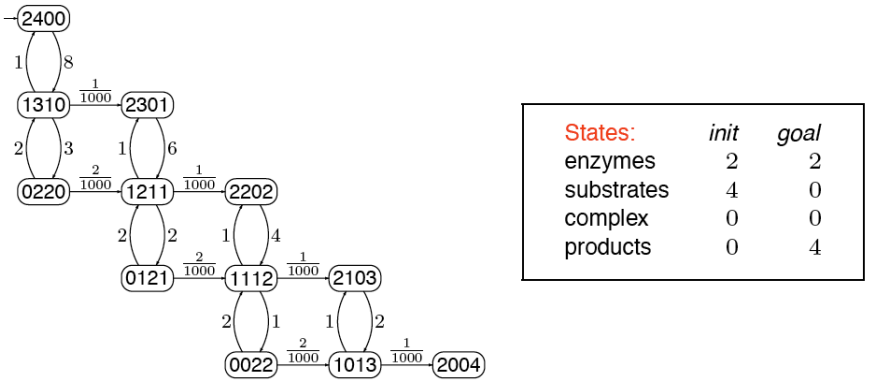

Figure 8. CTMC for enzyme-catalytic substrate conversion for initially 2 enzyme and 4 substrate species with $k_1 = k_2 = 1$ and $k_3 = 0.001$. The transition labels are rates of exponential distributions, i.e., the reciprocal of the average duration of a reaction.

second. Rates of kinetic reactions are obtained from enzyme assays [9], and depend on solution conditions and substrate concentration. The enzyme-substrate catalytic substrate conversion reaction is described by the stoichiometric equation:

$$E + S \underset{k_2}{\overset{k_1}{\rightleftharpoons}} C \xrightarrow{k_3} E + P$$

where k_i is the Michaelis-Menten constant for reaction i, which is the substrate concentration required for an enzyme to reach one-half of its maximum reaction rate. Now let us suppose we have N different types of molecules that randomly collide. The state $X(t)$ of the biological system at time instant $t \in \mathbb{R}_{\geqslant 0}$ is given by $X(t) = (x_1, \ldots, x_N)$ where x_i denotes the number of species of sort i. In the enzyme-catalytic substrate conversion case, N=4 and $i \in \{ C, E, P, S \}$. Let us number the types of reaction, e.g., $E{+}S \to C$ and $C \to E{+}S$ could be the first and second reaction, respectively. The reaction probability of reaction m within the infinitesimally small time-interval $[t, t{+}\Delta)$ with $\Delta \in \mathbb{R}_{\geqslant 0}$ is given by:

$$\alpha_m(\vec{x}) \cdot \Delta \;=\; Pr\{\text{reaction } m \text{ in } [t, t{+}\Delta) \mid X(t) = \vec{x}\}$$

where $\alpha_m(\vec{x}) = k_m \cdot$ the number of possible combinations of reactant molecules in $\vec{x}$. For instance, in state (x_E, x_S, x_C, x_P) where $x_i > 0$ for all $i \in \{E, S, C, P\}$, the reaction $E + S \to C$ happens with rate $\alpha_m(\vec{x}) = k_1 {\cdot} x_E {\cdot} x_S$ and yields the state $(x_E{-}1, x_S{-}1, x_C{+}1, x_P)$. This stochastic process possesses the Markov property, i.e., its future is completely described by the current state of the system. Moreover, it is time-homogeneous, i.e., its behaviour is invariant with respect to time shifts. In fact, it is a CTMC. Let us now consider the following question: given a certain concentration of enzymes and substrates, what is the likelihood that after four days all substrates have engaged in a catalytic step and resulted in products? In terms of the CTMC, this boils down to determining the probability that starting from the state $(x_E, x_S, 0, 0)$ we can reach a state of the form $(x_E, 0, 0, x_P)$ within four days. We will show that the key to solving this question is to consider transient probabilities.

[9] Enzyme assays are laboratory methods for measuring enzymatic activity.

7. Transient Probabilities

A CTMC starts in an initial state s. It then delays a bit, and probabilistically moves to a next state. There it will stay some period, and randomly select a next state. This process goes on *ad infinitum*. The probability distribution among the CTMC states after doing this for n steps is given by $\mathbf{1}_s \cdot \mathbf{P}^n$ where (as before) $\mathbf{1}_s$ is a function that yields one for state s and zero otherwise. That is, to determine what is the probability to be in a given CTMC state after taking n transitions can be answered as for DTMCs. The delays in the visited states do not play a role. A more interesting question however is to consider the probability distribution after a certain amount of time $t \in \mathbb{R}_{\geqslant 0}$ has elapsed. This amounts to taking a snapshot of the CTMC at a given time point t. This notion is called the *transient probability distribution* of the CTMC. For instance, for the aforementioned biology example, the transient distribution at time $\sqrt{2}$ is the probability to be in each of its state at time $\sqrt{2}$. The transient probability vector $\underline{p}(t) = (p_{s_1}(t), \ldots, p_{s_k}(t))$ satisfies:

$$\underline{p}'(t) \;=\; \underline{p}(t) \cdot (\mathbf{R} - \mathbf{r}) \qquad \text{given} \qquad \underline{p}(0)$$

where $\mathbf{r}$ is the diagonal matrix of vector $\underline{r}$ (interpreting function r that assigns rates to states as a a vector) and $\underline{p}'(t)$ is the first derivative of $\underline{p}(t)$. For the example four-state CTMC at time $\sqrt{2}$, we obtain that $\underline{p}(\sqrt{2}) = (p_0(\sqrt{2}), \ldots, p_3(\sqrt{2}))$ satisfies the following equation, where $\underline{p}(0)$ is the transposed version of $(1,0,0,0)$:

$$\underbrace{\begin{pmatrix} p_0'(\sqrt{2}) \\ p_1'(\sqrt{2}) \\ p_2'(\sqrt{2}) \\ p_3'(\sqrt{2}) \end{pmatrix}}_{=\underline{p}'(\sqrt{2})} = \underbrace{\begin{pmatrix} p_0(\sqrt{2}) \\ p_1(\sqrt{2}) \\ p_2(\sqrt{2}) \\ p_3(\sqrt{2}) \end{pmatrix}}_{=\underline{p}(\sqrt{2})} \cdot \left(\underbrace{\begin{pmatrix} 0 & \frac{25}{4} & \frac{25}{4} & \frac{25}{2} \\ \frac{4}{2} & 0 & 0 & \frac{4}{2} \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix}}_{=\mathbf{R}} - \underbrace{\begin{pmatrix} 25 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix}}_{=\mathbf{r}} \right)$$

Transient CTMC probabilities are a solution of a linear differential equation system.

The solution of the linear differential equation system using standard knowledge from analysis yields: $\underline{p}(t) \;=\; \underline{p}(0) \cdot e^{(\mathbf{R}-\mathbf{r})\cdot t}$. Here, the main problem is to tackle the matrix exponential $e^{(\mathbf{R}-\mathbf{r})\cdot t}$. There are several techniques to do so, but as in our setting the exponential is of a special form there is an elegant solution [10]. As a first step, we apply Taylor-Maclaurin expansion. This yields:

$$\underline{p}(t) \;=\; \underline{p}(0) \cdot e^{(\mathbf{R}-\mathbf{r})\cdot t} \;=\; \underline{p}(0) \cdot \sum_{i=0}^{\infty} \frac{((\mathbf{R}-\mathbf{r})\cdot t)^i}{i!}$$

The main drawback is the numerical instability due to fill-in of the matrix powers $(\mathbf{R}-\mathbf{r})^i$. This is (mainly) due to the presence of positive and negative entries in the matrix $\mathbf{R}-\mathbf{r}$. The way out to this problem is to use *uniformization* [GM84,Jen53].

[10] Computing the matrix exponential is a well-known problem in numerical mathematics with several pitfalls. In fact, there are several (dubious) ways on how to compute such matrix exponential as witnessed by the paper [MvL78]. Interesting enough, the same authors reported twenty-five years after their investigation of this problem an update of these numerical techniques [MvL03].

A CTMC is called uniform whenever the exit rates of all states are equal. That is, $r(s)$ equals a constant r, say, for each state s. Being uniform seems a severe restriction—under which circumstances is the rate of each state the same? Well, it turns out that each CTMC $\mathcal{C}$ can be transformed into an equivalent uniform CTMC $\overline{\mathcal{C}}$. This equivalence is a weak bisimulation. It goes beyond the scope of this tutorial to discuss in detail what this means, but the important repercussion is that the transient probabilities in $\mathcal{C}$ and in $\overline{\mathcal{C}}$ coincide. Let us explain how the uniformization works. The idea is to consider the rate $r(s)$ of the "fastest" state s on average (or any larger rate) in the CTMC $\mathcal{C}$. Let this rate be denoted r. Thus, $r = \max_s r(s)$. Thus, $\frac{1}{r}$ is the shortest mean residence time in some of the states in the CTMC $\mathcal{C}$. The procedure now is to adapt the rates of all states, such that every state is equipped with rate r. Every state s' with $r(s') = r(s) = r$ remains unaffected; neither its rate, nor its outgoing transition probabilities are affected. Now consider a state s' that is "slower" (on average) than s. In order to normalize the frequency of taking outgoing transitions of state s', the state is equipped with a self-loop. This self-loop mimics the fact that on average s' is slower than s. When increasing the rate of s' from $r(s')$ to $r = r(s)$, the frequency of taking one of the outgoing transitions of s' increases. In order to compensate, we need to increase the probability of staying in s'. This is done by means of introducing a self-loop with probability $1 - \frac{r(s')}{r}$. (If a self-loop already exists at state s', its probability becomes $\frac{r(s')}{r} \cdot \mathbf{P}(s', s') + (1 - \frac{r(s')}{r})$.) For other outgoing transitions, i.e., non self-loops, we set $\overline{\mathbf{P}}(s, s') = \frac{r(s)}{r} \cdot \mathbf{P}(s, s')$.

Uniformization amounts to normalize the residence time in every CTMC state.

Applying this uniformization principle to the example CTMC (for convenience depicted again in the left figure) yields the CTMC on the right: As state 3 has the highest rate,

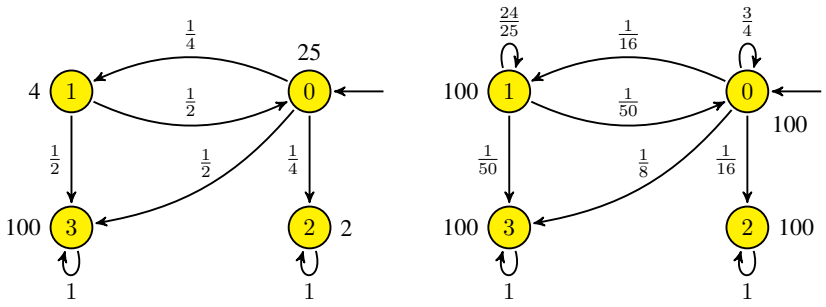

Figure 9. A CTMC (left) and its uniformized counterpart (right) when normalizing with respect to 100.

the other states are normalized with respect to $r(3) = 100$. In the uniformized CTMC $\overline{\mathcal{C}}$ (right), state 3 has not changed. As all states now have rate 100, on average every $\frac{1}{100}$ time units a transition in $\overline{\mathcal{C}}$ is taken. The other states are adapted according to the scheme described above. Let us consider state 0. The rate of this state is raised from 25 to 100. Thus, on average this state is accelerated by a factor $\frac{100}{25} = 4$. The frequency of taking a transition emanating state 0 is thus increased by a factor four. To compensate for this acceleration, in 75% of all cases, state 0 should idle, i.e., stay in state 0. This yields the self-loop with probability $\frac{3}{4}$. As in 25% of the cases state 0 should be left, the outgoing

transition probabilities of state 0 are scaled with a factor $\frac{1}{4}$. Similar reasoning applies to states 1 and 2.

As uniformization preserves transient probabilities, the probability in the CTMC (left) to be in a certain state at time $\sqrt{2}$ equals that for the probability at time $\sqrt{2}$ in the right CTMC. So, $\underline{p}(\sqrt{2})$ in $\mathcal{C}$ equals the vector $\underline{p}(\sqrt{2})$ in $\overline{\mathcal{C}}$. This holds for all time instants.

Uniformization preserves the transient probability distributions.

Let us see how this result helps in avoiding the numerical instability that occurred before in solving the linear differential equation system. As uniformization preserves transient probabilities, rather than taking $\underline{p}(t) = \underline{p}(0) \cdot e^{(\mathbf{R}-\mathbf{r})\cdot t}$ we take its counterpart for the uniformized CTMC, $\underline{p}(t) = \underline{p}(0) \cdot e^{(\overline{\mathbf{R}}-\overline{\mathbf{r}})\cdot t}$. We thus replace $\mathbf{R}-\mathbf{r}$ by $\overline{\mathbf{R}}-\overline{\mathbf{r}}$. Applying this scheme to our running example yields:

$$\overline{\mathbf{R}} - \overline{\mathbf{r}} = \begin{pmatrix} \frac{300}{4} & \frac{100}{16} & \frac{100}{16} & \frac{100}{8} \\ \frac{100}{50} & \frac{240}{25} & 0 & \frac{100}{50} \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix} - \begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{pmatrix}$$

We now have:

$$\overline{\mathbf{R}}(s,s') = \overline{\mathbf{P}}(s,s') \cdot \overline{r}(s) = \overline{\mathbf{P}}(s,s') \cdot r \quad \text{and} \quad \overline{\mathbf{r}} = \mathbf{I} \cdot r.$$

Thus: $\underline{p}(t) = \underline{p}(0) \cdot e^{(\overline{\mathbf{R}}-\overline{\mathbf{r}})\cdot t} = \underline{p}(0) \cdot e^{(\overline{\mathbf{P}}\cdot r - \mathbf{I}_{\cdot r})\cdot t} = \underline{p}(0) \cdot e^{(\overline{\mathbf{P}}-\mathbf{I})\cdot r\cdot t} = \underline{p}(0) \cdot e^{-rt} \cdot e^{r\cdot t\cdot \overline{\mathbf{P}}}$, where $\mathbf{I}$ is the identity matrix of the same cardinality as the matrix $\mathbf{R}$. Now one may argue that we gained nothing so far, as we are still left with computing a matrix exponential, namely $e^{r\cdot t\cdot \overline{\mathbf{P}}}$. This is true indeed, but the main difference is that now the matrix $\overline{\mathbf{P}}$ in the exponent is a stochastic matrix. It thus only contain values between zero and one, and each row sum equals one. Recall that before we had to consider a matrix $\mathbf{R}-\mathbf{r}$ as exponent that contains both (arbitrarily large) negative and positive entries. Let us now again exploit Taylor-Maclaurin expansion. This yields:

$$\underline{p}(t) = \underline{p}(0) \cdot e^{-rt} \cdot e^{r\cdot t\cdot \overline{\mathbf{P}}} = \underline{p}(0) \cdot e^{-rt} \cdot \sum_{i=0}^{\infty} \frac{(r\cdot t)^i}{i!} \cdot \overline{\mathbf{P}}^i = \underline{p}(0) \cdot \sum_{i=0}^{\infty} \underbrace{e^{-r\cdot t} \frac{(r\cdot t)^i}{i!}}_{\text{Poisson prob.}} \cdot \overline{\mathbf{P}}^i$$

As $\overline{\mathbf{P}}$ is a stochastic matrix, computing the matrix exponential $\overline{\mathbf{P}}^i$ is numerically stable. The infinite sum can be truncated by standard techniques, allowing for computing the number of summands that are necessary to yield the transient probabilities up to a given precision ε. Remark that there is a nice stochastic interpretation of the Poisson probabilities that occur in the summation. They express the fact that exactly i transitions have been taken in t time units in the uniformised CTMC, i.e., in a stochastic process in which transitions happen with an average frequency of r transitions per time unit.

Let us return to our four-state running example. As state 0 is initial, the initial distribution $\underline{p}(0) = (1,0,0,0)$. Let as before the time bound be $\sqrt{2}$. Applying the above scheme yields:

$$\underline{p}(\sqrt{2}) \;=\; \underbrace{\begin{pmatrix}1\\0\\0\\0\end{pmatrix}}_{=\underline{p}(0)} \cdot \sum_{i=0}^{\infty} \underbrace{e^{-100\sqrt{2}} \cdot \frac{(100\sqrt{2})^i}{i!}}_{\text{Poisson probability}} \cdot \underbrace{\begin{pmatrix} \frac{3}{4} & \frac{1}{16} & \frac{1}{16} & \frac{1}{8} \\ \frac{1}{50} & \frac{24}{25} & 0 & \frac{1}{50} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^i}_{=\overline{\mathbf{P}}^i}$$

The Poisson probability expresses the probability that i transitions have been taken in the uniformized CTMC $\overline{\mathcal{C}}$ in $\sqrt{2}$ time units. Recall that 100 was the largest rate in the original CTMC $\mathcal{C}$, and now acts as average frequency of taking transitions in CTMC $\mathcal{C}$. After a detour, we thus established that obtaining the transient distribution in a CTMC amounts to solving a linear differential equation system. In order to obtain an adequate numerical computation scheme, the CTMC is first uniformized. As this preserves transient probabilities, it suffices to solve the linear differential equation system for the uniformized CTMC. This is numerically stable.

8. Timed Paths

In order to address reachability questions on CTMCs we first need to consider paths through a CTMC. Whereas in a DTMC, a path is simply an infinite sequence of states, in a CTMC we need to keep track on how long we stay in each state along the path. A (timed) path in a CTMC is thus an infinite alternating sequence of states and time delays. In our example CTMC (before uniformization), $\underline{0}\,0.01\,\underline{1}\,\sqrt{7}\,\underline{0}\,e\,\underline{2}\,\pi\,\underline{2}\,1.2\ldots$ is a path, where for the sake of convenience we have underlined the states. This path thus visits the states in the order $0\,1\,0\,2^\omega$ while residing 0.01 time units in state 0 when visiting state 0 for the first time, then residing $\sqrt{7}$ time units in state 1, e time units on the next visit of state 0, and so forth. As a next step, we would like to define the probability of e.g., the set of paths that finally end up in state 2. Or even the set of paths that end up in state 2 before the total time spent in other states exceeds some threshold, 12 time units say. This is not trivial for as the probability of a single path is zero, as a path (by definition) is infinite. The solution is to resort—as in the discrete-time setting—to cylinder sets. This is slightly more complex than before, as the residence times in the individual states along a path needs to be considered as well. We do by considering *intervals*. Formally, the (interval) cylinder set of an (interval) path fragment $s_0\,I_0\,s_1\,I_1\,s_2\,I_2 s_3$, where I_0, I_1, and I_2 are intervals, is the set of timed paths in the CTMC at hand that all have a prefix $s_0\,t_0\,s_1\,t_1\,s_2\,t_2\,s_3$ where $t_0 \in I_0$, $t_1 \in I_1$, and $t_2 \in I_2$. For example the cylinder set of path fragment $\underline{0}\,[0.01, 0.02]\,\underline{1}\,[\sqrt{7}{-}0.1, \sqrt{7}{+}0.1]\,\underline{0}\,[e, \frac{4}{3}e]\,\underline{2}$ contains, e.g., the path $\underline{0}\,0.01\,\underline{1}\,\sqrt{7}\,\underline{0}\,e\,\underline{2}\,\pi\,\underline{2}\,1.2\ldots$ as well as the path $\underline{0}\,0.02\,\underline{1}\,\sqrt{7}\,\underline{0}\,\frac{7}{6}e\,\underline{2}\,\pi\,\underline{2}\,1.2\ldots$. We now define the probability of a cylinder set of $s_0\,I_0 \ldots I_{n-1} s_n$, denoted—with a bit of overloading of symbols—$Pr(C)$, as:

$$\prod_{j=1}^{n} \mathbf{P}(s_{j-1}, s_j) \cdot \underbrace{\int_{I_{j-1}} r(s_{j-1}) \cdot e^{-r(s_{j-1}) \cdot x}\, dx}_{\text{probability to leave } s_{j-1} \text{ in interval } I_{j-1}} .$$

For each state s_{j-1} along the path ($j > 0$), one takes the probability to move to the next state s_i along the path times the probability to leave the state s_{j-1} within the interval I_{j-1}. Any set of (timed) paths that can be written as the complement and/or countable union of (interval) cylinder sets is now measurable. The integrals can easily be solved, as

$$\int_{I_j} r(s_{j-1}) \cdot e^{-r(s_j) \cdot x} \, dx \;=\; e^{-r(s_j) \cdot \inf I_j} - e^{-r(s_j) \cdot \sup I_j}$$

where $\inf I_j$ and $\sup I_j$ are the infimum and supremum of the interval I_j respectively.

Probabilities of sets of infinite timed paths are defined using (interval) cylinder sets.

In timed systems such as CTMC, so-called *Zeno* paths [11] can occur. A Zeno path is a path in which the total time that elapses converges. In case $\sum_i t_i$ does not diverge, the timed path represents an"unrealistic" computation where infinitely many transitions are taken in a finite amount of time. An example Zeno path is $s_0 \, \frac{1}{2} \, s_1 \, \frac{1}{4} \, s_2 \, \frac{1}{8} \, s_3 \ldots$. The total elapsed time along that path is $\sum_i \frac{1}{2^i} \;=\; 1$. Thus time does not progress beyond one. One may argue that this goes against nature. In real-time systems, such executions are typically excluded from the analysis. Thanks to the following result, Zeno paths may occur in CTMCs but do not harm: their probability mass is zero.

For any CTMC, the probability mass of the set of Zeno paths is zero.

9. Timed Reachability Probabilities

We are now in a position to consider reachability objectives, and timed versions thereof. As before, we denote the set of paths that at some point reach a state in G, where G denotes the set of goal states, as $\Diamond G$. In a similar way, as for the discrete-time setting, one can prove that $\Diamond G$ is measurable, that is, it can be expressed in terms of a countable union and intersection of (interval) cylinder sets. The same applies for objectives such as $\Diamond\Box G$ and $\Box\Diamond G$. As all these properties do not impose any timing constraints, we can exploit the algorithms for DTMCs to determine their probabilities. For instance, the reachability probability for $G = \{\, 2 \,\}$ in our running CTMC example can be obtained by using solving a system of linear equations with $\mathbf{P}$ being the transition probability matrix of the CTMC at hand. The same applies to properties that are specified by finite-state automata or Rabin automata. These automata only constrain the order of certain sets of state labels to occur, but do have no means to constrain the time delays in the CTMC. As a result, the probability that a CTMC $\mathcal{C}$ generates (timed) paths that are accepted by an automaton $\mathcal{A}$ equals the reachability probability of "accepting" bottom SCCs in the product $\mathcal{C} \otimes \mathcal{A}$. There is no difference with the discrete-time setting.

For CTMCs, $Pr(\Diamond G)$ and $Pr(\mathcal{C} \models \mathcal{A})$ can be determined as for DTMCs.

It becomes however more interesting when considering properties that refer to the delays in $\mathcal{C}$. Examples of such properties are: what is the expected time to reach a set G of

[11] Named after the philosopher Zeno of Elea (490-430 BC) which was famed for his paradoxes.

states?, or what is the probability to reach some state in the set G within a given deadline d? In terms of the biology example discussed before, properties of interest include: what is the expected time from state 2400 to reach 2004? Stated in natural language: how long does it take on average to convert all available substrates (four) into products? Or: what is the probability that converting all substrates into products happens within 100 time units, i.e., the probability to reach state 2004 from 2400 within 100 time units? It is evident that the techniques we have discussed for DTMCs do not suffice.

For deadline d, where d is a non-negative real number, let $\Diamond^{\leqslant d}$ denote the set of timed paths that reach a state in G within time d. That is to say, the total time that has elapsed before reaching a state in G for the first time should not exceed d. For example, $\Diamond^{\leqslant 100}\{\,2004\,\}$ for the biology example, contains a (timed) path like

$$\underline{2400}\,0.3\,\underline{1310}\,2\,\underline{0220}\,30\,\underline{1211}\,\sqrt{50}\,\underline{2202}\,2\,\underline{1112}\,2\,\underline{0022}\,20\,\underline{1013}\,10\,\underline{2004}\ldots$$

since $0.3+2+30+\sqrt{50}+2+2+20+10 \leqslant 100$. If, however, the residence time in state 0022 would be 80, say, then the resulting (timed) path does not belong to $\Diamond^{\leqslant 100}\{\,2004\,\}$. It follows that $\Diamond G$ is nothing else than $\Diamond^{\leqslant \infty} G$ as in the latter no constraint is imposed on the deadline. It can be proven that for any CTMC with a countable state space each set of timed paths $\Diamond^{\leqslant d} G$ can be assigned a probability. (For CTMCs with uncountable state spaces, a similar result can be established; this falls outside the scope of this tutorial.) The same applies to sets of paths $\Box^{\leqslant d} G$, i.e., paths that remain in G states for at least the next d time units.

The quantity $Pr(\Diamond^{\leqslant d} G)$ is thus mathematically well defined. As a next step, we discuss a recursive characterization of this probability. More precisely, we want to determine the probability of the set of paths in $\Diamond^{\leqslant d} G$, given that we start in some state s. We denote this as $Pr(s \models \Diamond^{\leqslant d} G)$. Assuming that the Markov chain at hand has finitely many states, we let function variable $x_s(d) = Pr(s \models \Diamond^{\leqslant d} G)$ for state s. Our aim is to obtain a function definition $x_s(d)$ for every s and every non-negative real value d. The following recursive characterization will be helpful:

1. if G is not reachable from s, then $x_s(d) = 0$ for all d
2. if $s \in G$ then $x_s(d) = 1$ for all d
3. otherwise: $x_s(d) \;=\; \displaystyle\sum_{s' \in S} \int_0^d \underbrace{\mathbf{R}(s,s') \cdot e^{-r(s)\cdot x}}_{\substack{\text{probability to move to} \\ \text{state } s' \text{ at time } x}} \cdot \underbrace{x_{s'}(t-x)}_{\substack{\text{prob. to fulfill} \\ \Diamond^{\leqslant d-x} G \text{ from } s'}} \; dx$

The first two cases are self-explanatory. The last case considers the situation in which G can be reached from state s, but s does not belong to G. Then, in order to reach G, a transition emanating s should be taken. Let us assume that the transition to state s' is taken. Assuming that this transition is taken after a delay of x time units in state s, the remaining time from state s' to reach the set G is $d-x$. This happens with probability $Pr(s' \models \Diamond^{\leqslant d-x} G)$, i.e., $x_{s'}(d-x)$. The probability to reside x time units in state s is given by the density $r(s)\cdot e^{-r(s)\cdot x}$. Indeed, this is the density of an exponential distribution with rate $r(s)$. Given that the probability to take the outgoing transition from s to s' is $\mathbf{P}(s,s')$, we have that the probability to move to state s' from s at time x equals:

$$\mathbf{P}(s,s')\cdot r(s)\cdot e^{-r(s)\cdot x} \;=\; \frac{\mathbf{R}(s,s')}{r(s)}\cdot r(s)\cdot e^{-r(s)\cdot x} \;=\; \mathbf{R}(s,s')\cdot e^{-r(s)\cdot x}.$$

Given that moving to state s' after a delay of x time units is stochastically independent from going from state s' to the set G of goal states, we can multiply these two probabilities. Now this applies to state s'. But, of course, state s may have several direct successor states. As we have to take all these successors into account—there might be several routes along which s can reach G, and all contribute to the probability to reach G from s—we take the sum over all states s'. Finally, as x can take any value in the dense time interval $[0, d]$, we obtain the integral from 0 to d.

Let us apply this characterization to our four-state CTMC where $G = \{ 2 \}$ and the deadline is 10. Then we obtain:

1. $x_3(d) = 0$ for all d
2. $x_2(d) = 1$ for all d
3. for the states 0 and 1 that do not belong to G but can reach G we obtain:

$$x_0(d) = \int_0^{10} \underbrace{\frac{25}{4}}_{=\mathbf{R}(0,1)} \cdot e^{-25 \cdot x} \cdot x_1(d-x)\, dx + \int_0^{10} \underbrace{\frac{25}{4}}_{=\mathbf{R}(0,2)} \cdot e^{-25 \cdot x} \cdot x_2(d-x)\, dx$$

$$x_1(d) = \int_0^{10} \underbrace{\frac{4}{2}}_{=\mathbf{R}(1,0)} \cdot e^{-4 \cdot x} \cdot x_0(d-x)\, dx + \int_0^{10} \underbrace{\frac{4}{2}}_{=\mathbf{R}(1,3)} \cdot e^{-4 \cdot x} \cdot x_3(d-x)\, dx.$$

Using the function descriptions of x_2 and x_3, and solving the simple integrals, the latter two integral equations can be simplified to:

$$x_0(d) = \int_0^{10} \frac{25}{4} \cdot e^{-25 \cdot x} \cdot x_1(d-x)\, dx + \frac{1}{4} \left(1 - e^{-250}\right) \text{ and}$$

$$x_1(d) = \int_0^{10} \frac{4}{2} \cdot e^{-4 \cdot x} \cdot x_0(d-x)\, dx.$$

We thus obtain that timed reachability probabilities can be obtained by solving integral equation systems. Mathematicians have characterized such equation systems as Volterra integral equation systems (of type one). In fact, one can show that timed reachability probabilities are least solutions of a given integral equation system. This can be shown by lifting the above recursive characterization to a higher-order function, and using fixpoint theory. We refer the interested reader for further details to [BHHK03].

Timed reachability probability = unique solution of an integral equation system.

The question now raises: how can we solve these integral equations in an efficient manner? This turns out to be not that easy. In general this is a non-trivial issue, is inefficient, and has several pitfalls such as numerical stability. So standard numerical integration techniques do not help out at this point. We will however completely avoid to solve the integral equation system. How? We will reduce the problem of computing $Pr(s \models \Diamond^{\leqslant t} G)$ to an alternative problem for which well-known efficient techniques exist—transient probabilities. Let us go back one step, and consider our original problem statement: compute $Pr(s \models \Diamond^{\leqslant d} G)$ in CTMC $\mathcal{C}$. Observe that once a path reaches G within d time units, then the remaining states (and residence times) along the path is not important. What is of importance is that G was reached before the deadline d. This simple observation allows for a modification of the CTMC. It suggests to make all states in G *absorbing*, i.e., we delete all outgoing transitions from all states in G and replace

them by a self-loop. (It is not difficult to see that all states in G then can also be collapsed into a single state, but let us not consider this optimization here. In addition, the reader should bear in mind, that by making this transformation certain parts of the CTMC might become unreachable and thus do not have to be considered any further. This simplifies matters.) For the four-state Markov chain, depicted in the figure (left) below again, and $G = \{ 1 \}$, we obtain the CTMC on the right: Let $\mathcal{C}[G]$ denote the CTMC $\mathcal{C}$ in which all

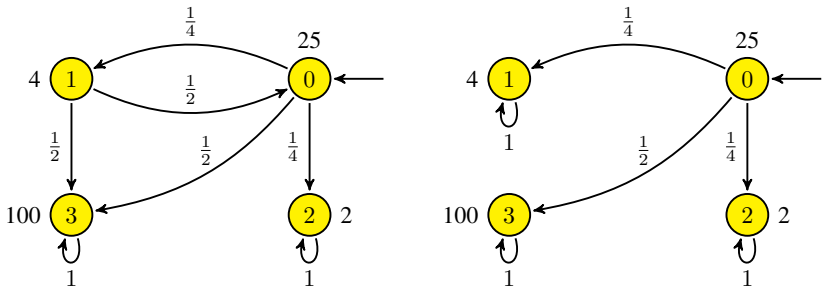

Figure 10. A CTMC (left) and its variant (right) in which the states in $G = \{ 1 \}$ are made absorbing.

states in G are made absorbing. Then it now follows:

$$\underbrace{Pr(s \models \Diamond^{\leqslant d} G)}_{\text{timed reachability in } \mathcal{C}} = \underbrace{Pr(s \models \Diamond^{=d} G)}_{\text{transient probability in } \mathcal{C}[G]} = \sum_{s' \in G} p_{s'}(d) \text{ where } \underline{p}(0) = \mathbf{1}_s.$$

Timed reachability probability = transient probability distribution in adapted CTMC.

10. Observing Markov Chains by Timed Automata

Recall that we have used (deterministic) automata to observe the paths of a DTMC. Such automata basically check which paths are acceptable and which ones are not. We have shown that the likelihood of all paths that are accepted by such automata can be obtained by determining reachability probabilities (of certain bottom SCCs) in a product construction of the Markov chain and the automaton. As argued before, one can take the same approach for CTMCs, as the finite-state or Rabin automata do not constrain the timing of the system model. Such observers only constrain the order in which states are to be visited, but have no influence on the delays that occur. We therefore consider an extension of (deterministic) automata that can constrain the timing. These are (deterministic) *timed automata*. What is a a timed automaton? Basically, it is a finite-state automaton that is equipped with clocks. Clocks are continuous variables, i.e., they take real values. Opposed to variables in a computer program whose value is changed by means of assignments, the value of clocks evolve implicitly. The only allowed assignment is to set a clock to zero. Clocks can thus be reset, but not assigned an arbitrary value. All clocks have initially the value zero. Boolean conditions on clocks, however,

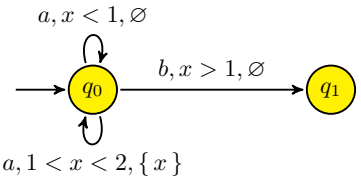

Figure 11.: A timed automaton that on reading a when $1 < x$ and $x < 2$ resets x, and only is able to read b when x exceeds one.

can be used as guard of transitions. A transition from state q to q' equipped with guard $x > 1 \wedge x < 2$ for clock x, can only be taken whenever being in state q, the value of clock x lies in the interval $(1, 2)$. Transitions in timed automata are labelled with three elements: an input symbol, a guard, and a set of clocks. The operational meaning of a transition from q to q' labelled with the triple (a, b, X) is that when the guard b holds in state q, and the next input symbol is a, then the timed automaton can move to state q' while resetting all clocks in the set X. The timed automaton in Figure 11, e.g., resets the clock x when reading input symbol a in state q_0 whenever $x > 1$ and $x < 2$. If it reads input symbol a whenever $x < 1$, clock x is not reset. The timed automaton can switch from state q_0 to q_1 on reading input b whenever $x > 1$. This change to state q_1 can be postponed with at least one time unit by resetting x before it has reached the value two. It is important to realize that the underlying state space of a timed automaton is infinite, due to the fact that clocks are real-valued. A configuration of a timed automaton is a pair consisting of a control state, and a function assigning a value to each clock. Example configurations for the example timed automaton are, for instance, $(q_0, x = 0)$, $(q_0, x = \sqrt{2})$, and $(q_0, x = e)$. A timed automaton accepts timed words, i.e., alternating sequences of input symbols and time instants. For the sake of simplicity, we will focus on timed automata that accept finite timed words. These timed automata are the timed analogue of finite automata.

Let us illustrate the usage of timed automata as observers of CTMC by means of an example. Consider a robot randomly moving in some area. It starts in some grid cell (A, say) and has to reach cell B within 10 time units, cf. Figure 12 (left). (For simplicity, all cells on the map are equally-sized, but this is not a restriction.) The robot randomly moves through the cells, and resides in an area for an exponentially distributed amount of time. The robot may pass through all cells to reach B, but should not stay longer than 2 time units in any gray area, i.e., any area consisting of a number of adjacent gray cells. We are interested in the probability that B is reached within 10 time units without violating the time constraint on the maximal allowed residence time in gray areas. The specification "reach B from A within 10 time units while residing in any gray area for at most 2 time units" is naturally modelled by a deterministic timed automaton, cf. Figure 12 (right). Clock x controls the timing constraint on the residence times of the gray cells (assumed to be labelled with g), while clock y controls the global time constraint to reach cell B (labelled with b). In state q_0, the robot traverses non-gray cells, in q_1 gray cells, and in accepting state q_2 it has reached the goal cell B. For simplicity, we use logical formulas to indicate labels; e.g., the label $\neg g \wedge \neg b$ is a shorthand for any label that differs from g and b. In state q_0, traversing a cell that is neither a gray cell, nor a B cell is possible without any constraint. This is modelled by the self-loop at state q_0. If a B-cell is entered within the time bound 10, the timed automaton moves to state q_2 and accepts. On entering a gray area, clock x is set. This happens in the transition from state q_0 to q_1. If in state q_1, the target is reached without violating any of the time constraints, the timed automaton moves to q_2 and accepts. On entering a next gray cell, x is left unchanged. This is modelled by the self-loop at state q_1. On leaving the gray area, the automaton moves to q_0 without affecting clock x. If in states q_0 or q_1 clock y exceeds the deadline 10, the automaton moves to a reject state ragardless of the next

input symbol. In order not to blur the picture, this reject state is not depicted. Remark that the timed automaton is deterministic: in any state with a given clock value for x and y it is uniquely determined for a given input symbol what will be the next state.

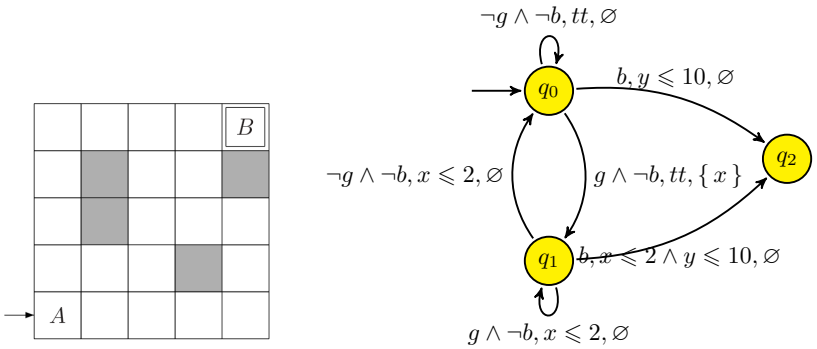

Figure 12. A grid structure on which the robot randomly moves (left), and a timed property automaton (right) specifying that B should be reached from A in less than 10 time units while visiting any block of adjacent gray cells for at most two time units.

Recall that states in CTMCs —like in Kripke structures— are labeled with sets of atomic propositions. These labels have not been of importance so far, but now are of relevance. The *timed trace* of a timed path $s_0\, t_0\, s_1\, t_1\, s_2\, t_2 \ldots$ is the infinite alternating sequence $L(s_0)\, t_0\, L(s_1)\, t_1\, L(s_2)\, t_2 \ldots$, where $L(s_i)$ is the set of propositions attached to state s_i. A timed trace is thus obtained from a timed path by a point-wise projection of each state in the path to its set of labels. The idea is now that timed traces obtained from paths of the CTMC are fed as input to the timed automaton. That is, the observer automaton reads the timed traces as input words. As a DTA is deterministic, it is for any time instant and input symbol uniquely determined what the next state of the DTA is. This entails that the alphabet of the timed automaton consists of sets of atomic propositions. We now need to fix when an observer timed automaton accepts a timed trace, and when it does not. This is determined by the acceptance condition of the automaton. As we focus on finite acceptance conditions, the timed automaton accepts timed words. (The setting can equally well be applied to timed automata that accepts infinite timed words; for details, see [CHKM11].) It thus is equipped with a set of accepting states, and accepts once one of such states is read.

To summarize, the setting is as follows: the possible system behaviour is modelled by a finite-state CTMC $\mathcal{C}$, and the required system behaviour is given as a deterministic timed automaton (DTA, for short), $\mathcal{A}$ say. The problem of interest now is to determine the fraction of paths of the Markov chain that are accepted by the DTA. Stated differently, what is the probability mass of the set of timed paths generated by $\mathcal{C}$ which are accepted by $\mathcal{A}$? We denote this probability by $Pr(\mathcal{C} \models \mathcal{A})$. Formally, we have

$$Pr(\mathcal{C} \models \mathcal{A}) = Pr\{\, \sigma \text{ is a timed path of CTMC } \mathcal{C} \mid \mathit{trace}(\sigma) \text{ is accepted by DTA } \mathcal{A} \,\}.$$

It can be proven that the sets of paths accepted by a DTA is measurable, i.e., can be characterized in terms of (interval) cylinder sets. As a next step, we consider the product

automaton. As opposed to the discrete-time setting where a product of the Markov chain and the property automaton was taken, we here proceed in a slightly different manner.

The crux of the construction is to take a product of the CTMC $\mathcal{C}$ with an abstraction of the DTA $\mathcal{A}$. This abstraction is called the *zone graph* of the DTA $\mathcal{A}$, and is denoted $ZG(\mathcal{A})$. This zone graph is a finite-state automaton, in which states consist of a pair (q, Z) where q is a state of the DTA, and Z is a zone. What is a zone? A zone is a conjunction of constraints on clocks. Such constraints take the form $x-y < k, x-y \leqslant k$ (or similar with $>$, $=$, or $\geqslant$) for clocks x and y, and natural number k. Constraints such as $x < k$ are written as $x - \mathbf{0} < k$, where $\mathbf{0}$ is a clock that has constant value zero. It is implicitly assumed that clocks cannot be negative. A different way to look at zones is to consider them as convex polyhedra. In case there are only two clocks, a zone thus amounts to be a convex polygon. This geometrical interpretation of zones is fully equivalent to the description using conjunctions of clock constraints. An example zone for the DTA of the robot example is: $x = y \wedge x \leqslant 2 \wedge y \leqslant 10$, describing the line fragment $x = y$ for $0 \leqslant x, y \leqslant 2$. Another zone is $y - x \geqslant 5 \wedge y \leqslant 10 \wedge x > 2$. Clearly, a zone represents an infinite set of clock values. As argued before, a configuration of a timed automaton consists of the current state of the automaton together with the values of all clocks. This yields infinitely many, even uncountably many, possible configurations. Rather than keeping track of the precise values of all clocks, it suffices to know the range of each clock. That is to say, it suffices to keep track of the zone of the clocks involved. This yields (abstract) configurations of the form (q, Z). Indeed, the states of a zone graph act as (abstract) configurations of a timed automaton.

We now exploit a result from the theory of timed automata, stating that for every timed automaton $\mathcal{A}$, a zone graph $ZG(\mathcal{A})$ can be constructed that represents all its reachable (abstract) configurations. This result dates back to Alur and Dill [AD94] and has important repercussions. In particular, it means that there is a finite representation of all reachable configurations of $\mathcal{A}$. In fact, this finite representation is sound and complete with respect to reachability in $\mathcal{A}$. This means that it precisely represents all reachable configurations of $\mathcal{A}$ and nothig more.

Zone graph $ZG(\mathcal{A})$ is sound and complete w.r.t. reachability of DTA $\mathcal{A}$.

As the zone graph of the DTA $\mathcal{A}$ is basically a finite-state automaton, the product $\mathcal{C} \otimes \mathcal{A}$ of CTMC $\mathcal{C}$ and DTA $\mathcal{A}$ can be constructed along the same way as in the discrete-time setting. It then follows (without going into the technical details here):

$Pr(\mathcal{C} \models \mathcal{A})$ = reachability probability of "accepting" bottom SCCs in $\mathcal{C} \otimes ZG(\mathcal{A})$.

In contrast to the discrete-time setting where the product $\mathcal{D} \times \mathcal{A}$ for DRA $\mathcal{A}$ is (again) a DTMC this does not apply to $\mathcal{C} \otimes ZG(\mathcal{A})$. This product is neither a CTMC nor a DTA. It turns out, that it is a stochastic process known as piecewise deterministic Markov process. It goes beyond the scope of this tutorial to explain the details of this more involved stochastic process; the interested reader is referred to [Dav93]. Reachability probabilities in such stochastic processes can be characterized by Volterra integral equation systems (of the second type). Their computation is involved. For DTA that contain a single clock, however, these reachability probabilities are unique solutions of linear equation systems where the coefficients are solutions of linear differential equation systems. Each differential equation system corresponds to performing a transient analysis on a CTMC, in

fact, as a sub-process of $\mathcal{C} \otimes \mathcal{A}$. In order to tackle this case, one first thus has to carry out a couple of transient analyses (one for each sub-process), and then to solve a linear equation system.

11. Epilogue

In this tutorial, we have attempted to give a gentle introduction to the foundations of model checking of Markov chains. As Markov chains are used in several areas ranging from reliability and dependability analysis to systems biology and psychology, an enormous potential application area of these techniques is available. A historical account of model checking of Markov chains has been given in [BHHK10]. That paper also contains a description of the main merits of model checking Markov chains over standard analysis techniques for Markov chains as used in classical performance evaluation. The emphasis of this tutorial has been to convey the main intuition behind the model-checking algorithms for Markov chains. More detailed treatments can be found in [BK08, Chapter 10] for DTMCs (as well as Markov decision processes) and in [BHHK03] [KNP07] for CTMCs. Various extensions of the presented approaches have been investigated in the literature. These include, amongst others, extensions to infinite-state Markov chains such as in probabilistic pushdown automata and recursive Markov chains [Ete13]. Here, the focus is mainly on decidability and theoretical complexity issues. Other important variations have been generalizations towards Markov reward models. In these models, states and/or transitions are equipped with real-valued rewards and are added along path fragments. Measures such as: what is the likelihood to reach a goal state without earning more than a certain reward, or what is the expected reward until reaching such state are quite common. Various model-checking algorithms have been developed for reward models; see [BHH$^+$13]. Various software tools have been realized to support Markov chain model checking; we mention PRISM [12] and MRMC [13]. Both tools support DTMCs and CTMCs and mainly concentrate on branching-time temporal logics, basically variants of CTL. PRISM is a symbolic model checker using a variant of binary decision diagrams; MRMC is an explicit state model checker. PRISM is a full-fledged tool aimed at direct usage by modellers supporting a modeling language and a GUI. MRMC uses a rudimentary textual matrix representation of the Markov chain and is easy to use as backend to existing modelling tools. The PRISM web page contains a large variety of case studies. A recent example of a serious attempt to exploit Markov chain model checking in the aerospace industry is reported in [BCK$^+$11]. Both aforementioned tools use numerical techniques. Statistical model checking is an alternative technique based on discrete-event simulation and has not been treated here.

Acknowledgement. The author thanks Friedrich Gretz, Dennis Guck, and Falak Sher for valuable feedback on an earlier version of this tutorial.

[12] www.prismmodelchecker.org
[13] www.mrmc-tool.org

References

[AD94] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[BCK+11] M. Bozzano, A. Cimatti, J.-P. Katoen, V. Y. Nguyen, T. Noll, and M. Roveri. Safety, dependability and performance analysis of extended AADL models. *The Computer Journal*, 54(5):754–775, 2011.

[BHH+13] C. Baier, E. M. Hahn, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Performability using model checking. *Mathematical Structures in Computer Science*, 2013. to appear.

[BHHK03] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[BHHK10] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Performance evaluation and model checking join forces. *Communications of the ACM*, 53(9):76–85, 2010.

[BK08] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[CHKM11] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model checking of continuous-time Markov chains against timed automata specifications. *Logical Methods in Computer Science*, 7(1–2):1–34, 2011.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[Dav93] M. H. A. Davis. *Markov Models and Optimization*. Chapman and Hall, 1993.

[Ete13] K. Etessami. Analysis of probabilistic processes and automata theory. In J. E. Pin, editor, *Automata: from Mathematics to Applications*. 2013. to appear.

[GM84] D. Gross and D. R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov chains. *Operations Research*, 32(2):343–361, 1984.

[Jen53] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skandinavisk Aktuarietidskrift*, 3:87–91, 1953.

[KNP07] M. Z. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In M. Bernardo et al., editor, *7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems (SFM)*, volume 4486 of *LNCS*, pages 220–270. Springer, 2007.

[KY76] D. E. Knuth and A. C. Yao. The complexity of nonuniform random number generation. In J.E. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 357–428. Academic Press, New York, 1976.

[MvL78] C. B. Moler and C. F. van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

[MvL03] C. B. Moler and C. F. van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45:3–49, 2003.