The What, Why, and How of Probabilistic Verification

Part 2: Algorithmic Foundations

Joost-Pieter Katoen

RWTH AACHEN UNIVERSITY

UNIVERSITY OF TWENTE.

CAV Invited Tutorial 2015, San Francisco

Overview

Algorithmic Foundations

- Markov Chains
- Markov Decision Processes
- Continuous-Time Markov Chains
- Continuous-Time Markov Decision Processes

Overview

Algorithmic Foundations

- Markov Chains
- Markov Decision Processes
- Continuous-Time Markov Chains
- Continuous-Time Markov Decision Processes

Discrete-Time Markov Chains

Discrete-time Markov chain

A DTMC $\mathcal{D}$ is a tuple $(S, \mathbf{P}, \iota_{init}, L)$ with:

- S is a finite non-empty set of states
- $\mathbf{P} : S \times S \to [0,1]$, transition probability function s.t. $\sum_{s'} \mathbf{P}(s, s') = 1$
- $\iota_{init} : S \to [0,1]$, the initial distribution with $\sum_{s \in S} \iota_{init}(s) = 1$
- $L : S \to 2^{AP}$, the labeling function, assigning to state s, the set $L(s)$ of atomic propositions in AP that are valid in s.

Initial states

- $\iota_{init}(s)$ is the probability that DTMC $\mathcal{D}$ starts in state s
- the set $\{ s \in S \mid \iota_{init}(s) > 0 \}$ are the possible initial states.

Reachability Probabilities

Problem statement

Consider a MC with finite state space S, $s \in S$ and $G \subseteq S$.

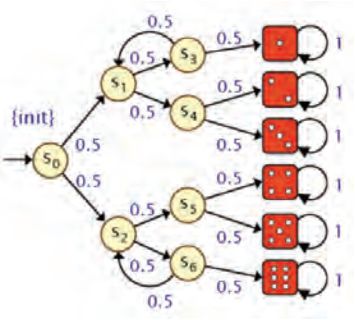
Aim: determine $\Pr(s \models \Diamond G) = \Pr_s\{\, \pi \in Paths(s) \mid \pi \models \Diamond G \,\}$

Characterisation of reachability probabilities

- Let variable $x_s = \Pr(s \models \Diamond G)$ for any state s
  - if G is not reachable from s, then $x_s = 0$
  - if $s \in G$ then $x_s = 1$
- For any state $s \in Pre^*(G) \setminus G$:

$$x_s = \underbrace{\sum_{t \in S \setminus G} \mathbf{P}(s,t) \cdot x_t}_{\text{reach } G \text{ via } t \in S \setminus G} + \underbrace{\sum_{u \in G} \mathbf{P}(s,u)}_{\text{reach } G \text{ in one step}}$$

Reachability Probabilities: Knuth-Yao's Die



- Consider the event $\Diamond 4$
- Using the previous slide we obtain:

$$x_1 = x_2 = x_3 = x_5 = x_6 = 0 \text{ and } x_4 = 1$$

$$x_{s_1} = x_{s_3} = x_{s_4} = 0$$

$$x_{s_0} = \frac{1}{2}x_{s_1} + \frac{1}{2}x_{s_2}$$

$$x_{s_2} = \frac{1}{2}x_{s_5} + \frac{1}{2}x_{s_6}$$

$$x_{s_5} = \frac{1}{2}x_5 + \frac{1}{2}x_4$$

$$x_{s_6} = \frac{1}{2}x_{s_2} + \frac{1}{2}x_6$$

- Gaussian elimination yields:

$$x_{s_5} = \frac{1}{2},\ x_{s_2} = \frac{1}{3},\ x_{s_6} = \frac{1}{6},\ \text{and } \boxed{x_{s_0} = \frac{1}{6}}$$

Unique Solution of Linear Equation System

Reachability probabilities as linear equation system

- Let $S_? = Pre^*(G) \setminus G$, the states that can reach G by > 0 steps
- $\mathbf{A} = \big(\mathbf{P}(s,t) \big)_{s,t \in S_?}$, the transition probabilities in $S_?$
- $\mathbf{b} = \big(b_s \big)_{s \in S_?}$, the probs to reach G in 1 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:

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

where $\mathbf{I}$ is the identity matrix of cardinality $|S_?| \times |S_?|$.

Long-Run Behaviour

Long-run theorem

Almost surely any finite Markov chain eventually reaches a terminal SCC and visits all its states infinitely often.

Reachability Probabilities are Pivotal

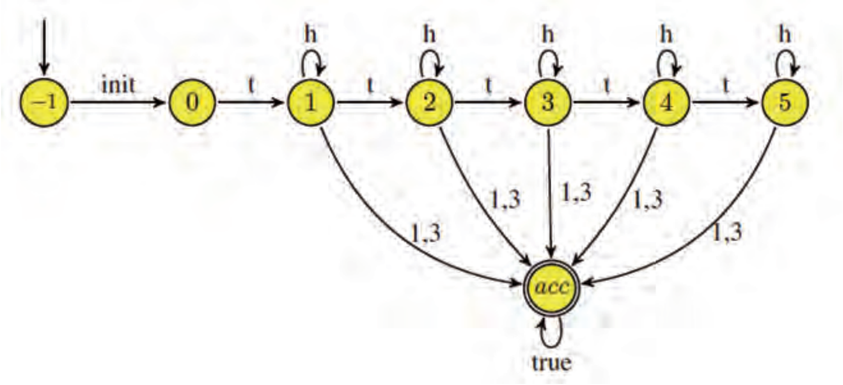
- Repeated reachability $\Pr(s \models \Box \Diamond G)$:
  1. Determine the terminal SCCs of the Markov chain
  2. Consider those that contain at least one G state
  3. Determine the probability to reach one of them from s

- Probabilistic CTL model checking
  1. Recursive descent on parse tree using reach probabilities at nodes
  2. Reduce until-modalities to reachability problem
  3. Yields a polynomial-time algorithm in model and formula

- LTL formulas $\Pr(s \models \varphi)$:
  1. Transform φ into a deterministic (Rabin) automaton
  2. Take the product of the Markov chain and this automaton
  3. Determine the reach-probability of an accepting terminal SCC from s

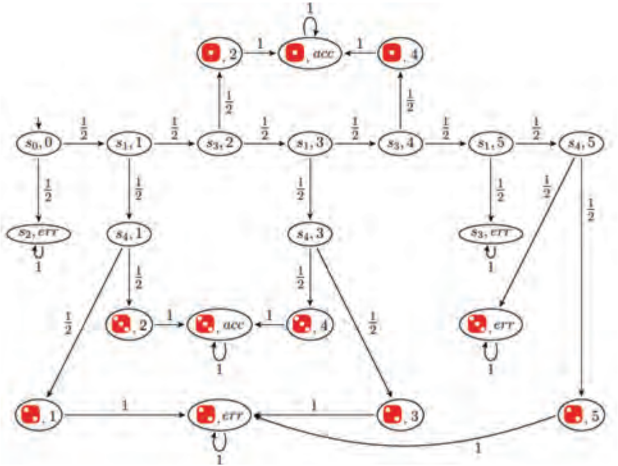
Consider LTL model checking a bit more in detail.

Property of Knuth-Yao's Algorithm



After initial tails, yield 1 or 3 but with at most five times tails in total

Product Markov Chain



Reachability probability of terminal SCC with $(\cdot, q_{acc})$ is $\frac{1}{8} + \frac{1}{8} + \frac{1}{32} + \frac{1}{32} = \frac{5}{16}$.

What About LTL?

Let φ be an LTL formula whose infinite sequences are $[\![\varphi]\!]$.

LTL is ω-regular

$[\![\varphi]\!]$ is an ω-regular language.

LTL is DRA-definable

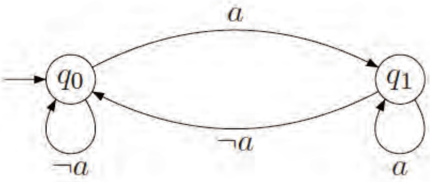
There exists a DRA $\mathcal{A}$ such that $\mathcal{L}_\omega(\mathcal{A}) = [\![\varphi]\!]$ where the number of states in $\mathcal{A}$ lies in $2^{2^{|\varphi|}}$.

A DRA is a finite automaton with acceptance sets $\{ (L_1, K_1), \ldots, (L_n, K_n) \}$ with $L_i, K_i \subseteq Q$.

Deterministic Rabin automaton: Example

Acceptance condition

A run of a word in Σ^ω on a DRA is accepting iff $\bigvee_{0<i\leqslant n} (\Diamond\Box \neg L_i \wedge \Box\Diamond K_i)$.



For $\mathcal{F} = \{ (L, K) \}$ with $L = \{ q_0 \}$ and $K = \{ q_1 \}$, this DRA accepts $\Diamond\Box\, a$

There does not exist a deterministic Büchi automaton for $\Diamond\Box\, a$.

Verifying ω-Regular Objectives = Reachability

Verifying DRA objectives theorem

Let $\mathcal{D}$ be a finite DTMC with state s, $\mathcal{A}$ a DRA with n acceptance sets. Then:

$$\underbrace{\Pr(s \models \mathcal{A})}_{\text{in } \mathcal{D}} = \underbrace{\Pr\Big(\langle s, q_s \rangle \models \Diamond U \Big)}_{\text{in } \mathcal{D} \otimes \mathcal{A}} \quad \text{with} \quad \underbrace{q_s = \delta(q_0, L(s))}_{\mathcal{A} \text{ after reading } L(s)}$$

where U is the union of all accepting terminal SCCs in $\mathcal{D} \otimes \mathcal{A}$.

Terminal SCC $T \subseteq S \times Q$ is accepting iff for some $0 < i \leqslant n$ it contains no L_i-state and some K_i-state.

Satisfaction probabilities for ω-regular properties in DTMC $\mathcal{D}$ = reachability probabilities for certain terminal SCCs in $\mathcal{D} \otimes \mathcal{A}$.

A graph analysis and solving systems of linear equations suffice.

All You Need to Know About Probabilistic CTL

- Qualitative PCTL only allow the probability bounds > 0 and $= 1$.
- There is no CTL formula that is equivalent to $\mathbb{P}_{=1}(\Diamond a)$.
- There is no PCTL formula that is equivalent to $\forall \Box\, a$.
- These results do not apply to finite DTMCs.
- $\mathbb{P}_{=1}(\Diamond a)$ and $\forall \Diamond\, a$ are equivalent under fairness.
- Repeated reachability probabilities are PCTL definable.

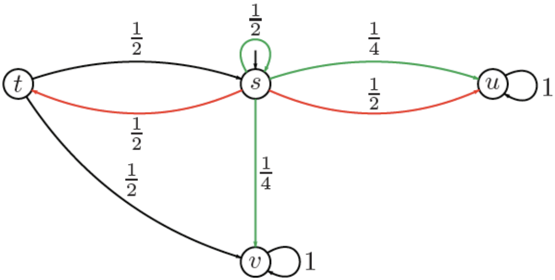
Take-home messages

Qualitative PCTL and CTL have incomparable expressiveness. Qualitative and fair CTL are equally expressive. Repeated reachability and persistence probabilities are PCTL definable. Their qualitative counterparts are not expressible in CTL.

Overview

Non-determinism: MDP

An MDP is a DTMC in which in any state a non-deterministic choice between probability distributions exists.



Set of enabled distributions (= colors) in state s is $Act(s) = \{ \alpha, \beta \}$ where

- $\mathbf{P}(s, \alpha, s) = \frac{1}{2}$, $\mathbf{P}(s, \alpha, t) = 0$ and $\mathbf{P}(s, \alpha, u) = \mathbf{P}(s, \alpha, v) = \frac{1}{4}$
- $\mathbf{P}(s, \beta, s) = \mathbf{P}(s, \beta, v) = 0$, and $\mathbf{P}(s, \beta, t) = \mathbf{P}(s, \beta, u) = \frac{1}{2}$

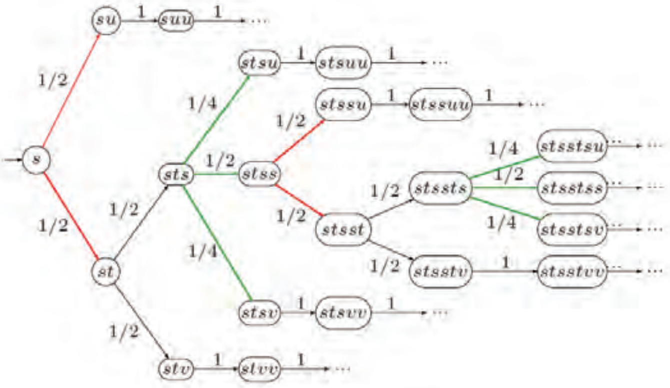
Policies

To solve MDPs, non-determinism is resolved by an oracle, called a policy.

Policy

A policy for MDP M is a function $\mathfrak{S}$ that for a give finite sequence of states through $\mathcal{M}$ yields an action (= color) to take next.

MDP + Policy = Markov Chain



Induced DTMC for a policy that alternates between selecting red and green starting with red.

Reachability Probabilities

Let $\mathcal{M}$ be an MDP with state space S and $\mathfrak{S}$ be a policy on $\mathcal{M}$. The reachability probability of $G \subseteq S$ from state $s \in S$ under policy $\mathfrak{S}$ is:

$$\overset{\mathfrak{S}}{\Pr}(s \models \Diamond G) \;=\; \underset{s}{\overset{\mathcal{M}_{\mathfrak{S}}}{\Pr}} \{ \, \pi \in Paths(s) \mid \pi \models \Diamond G \, \}$$

Maximal and minimal reachability probabilities

The minimal reachability probability of $G \subseteq S$ from $s \in S$ is:

$$\overset{\min}{\Pr}(s \models \Diamond G) = \inf_{\mathfrak{S}} \overset{\mathfrak{S}}{\Pr}(s \models \Diamond G)$$

In a similar way, the maximal reachability probability of $G \subseteq S$ is:

$$\overset{\max}{\Pr}(s \models \Diamond G) \;=\; \sup_{\mathfrak{S}} \overset{\mathfrak{S}}{\Pr}(s \models \Diamond G).$$

Equation System for Max-Reach Probabilities

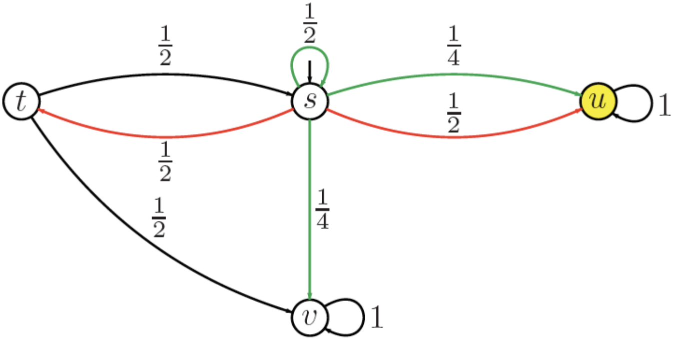
Let $\mathcal{M}$ be a finite MDP with state space S, $s \in S$ and $G \subseteq S$. The vector $(x_s)_{s \in S}$ with $x_s = \Pr^{\max}(s \models \Diamond G)$ yields the unique solution of the following equation system:

- If $s \in G$, then $x_s = 1$.
- If $s \not\models \exists \Diamond G$, then $x_s = 0$.
- If $s \models \exists \Diamond G$ and $s \notin G$, then

$$x_s \;=\; \max \left\{ \sum_{t \in S} \mathbf{P}(s, \alpha, t) \cdot x_t \;\middle|\; \alpha \in Act(s) \right\}$$

This is an instance of the Bellman equation for dynamic programming.

Example



equation system for reachability objective $\Diamond\{ u \}$ is:

$$x_u = 1 \text{ and } x_v = 0$$

$$x_s = \max\{ \tfrac{1}{2}x_s + \tfrac{1}{4}x_u + \tfrac{1}{4}x_v, \tfrac{1}{2}x_u + \tfrac{1}{2}x_t \} \quad \text{and} \quad x_t = \tfrac{1}{2}x_s + \tfrac{1}{2}x_v$$

Positional Policies Suffice for Reachability

A positional policy selects the next action only based on the current state.

Existence of optimal positional policies

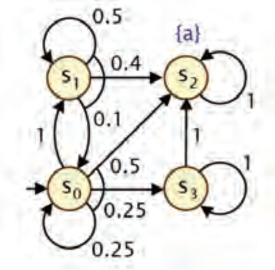
Let $\mathcal{M}$ be a finite MDP with state space S, and $G \subseteq S$. There exists a positional policy $\mathfrak{S}$ such that for any $s \in S$ it holds:

$$\overset{\mathfrak{S}}{\Pr}(s \models \Diamond G) = \overset{\max}{\Pr}(s \models \Diamond G).$$

A similar result holds for minimal reachability probabilities.

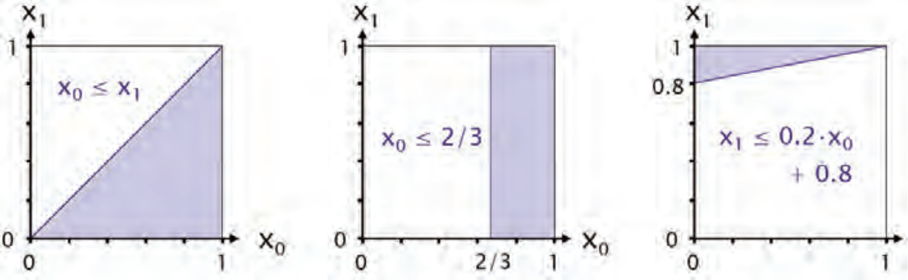
Techniques to obtain these policies: value or policy iteration, linear programming

Example Using Linear Programming

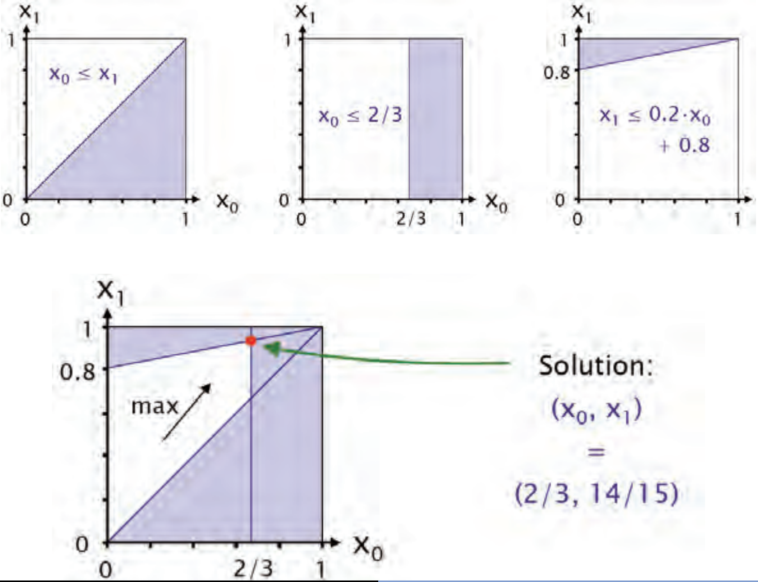


- $G = \{ s_2 \}, S^{\min}_{=0} = \{ s_3 \}, S \setminus (G \cup S^{\min}_{=0}) = \{ s_0, s_1 \}$.
- Maximise $x_0 + x_1$ subject to the constraints:

$$
\begin{aligned}
x_0 &\leqslant x_1 \\
x_0 &\leqslant \frac{2}{3} \\
x_1 &\leqslant \frac{2}{5} \cdot x_0 + \frac{4}{5}
\end{aligned}
$$



Example Linear Programming



Solution:

(x_0, x_1)

$=$

$(2/3, 14/15)$

Reachability Probabilities are Pivotal

Long-run theorem

Almost surely any finite MDP eventually reaches a terminal end-component and visits all its states infinitely often.

- Repeated reachability $\Pr^{\max}(s \models \Box \Diamond G)$:
  1. Determine the terminal end-components of the MDP
  2. Consider those that contain at least one G state
  3. Determine the maximal probability to reach one of them from s

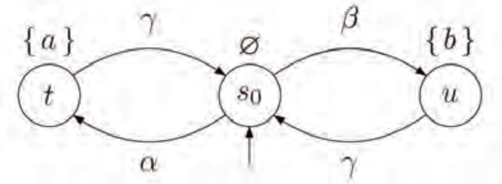
Reachability Probabilities are Pivotal

Probabilistic CTL model checking

1. The probabilistic operator $\mathbb{P}_J(\cdot)$ imposes probability bounds for all policies
   - Checking $s \models \mathbb{P}_{>p}(\varphi)$? amounts to $\Pr^{\min}(s \models \varphi) > p$?
   - Checking $s \models \mathbb{P}_{<p}(\varphi)$ amounts $\Pr^{\max}(s \models \varphi) < p$.
2. Recursive descent on parse tree using reach probabilities at nodes
3. Pre-determine states satisfying until-modalities with = 0 or = 1
4. Reduce until-modalities to reachability problem
5. Yields a polynomial-time algorithm in model and formula
6. This is generalisable to treating fair policies

What About LTL?

Consider the MDP:



Positional policy $\mathfrak{S}_\alpha$ always chooses α in state s_0
Positional policy $\mathfrak{S}_\beta$ always chooses β in state s_0. Then:

$$\Pr_{\mathfrak{S}_\alpha}(s_0 \models \Diamond a \wedge \Diamond b) = \Pr_{\mathfrak{S}_\beta}(s_0 \models \Diamond a \wedge \Diamond b) = 0.$$

Now consider the policy $\mathfrak{S}_{\alpha\beta}$ which alternates between selecting α and β. Then:

$$\Pr_{\mathfrak{S}_{\alpha\beta}}(s_0 \models \Diamond a \wedge \Diamond b) = 1.$$

Overview

Algorithmic Foundations

- Markov Chains
- Markov Decision Processes
- Continuous-Time Markov Chains
- Continuous-Time Markov Decision Processes

Random Timing

Negative Exponential Distribution

Density of exponential distribution

The density of an *exponentially distributed* r.v. Y with *rate* $\lambda \in \mathbb{R}_{>0}$ is:

$$f_Y(x) = \lambda \cdot e^{-\lambda \cdot x} \quad \text{for } x > 0 \quad \text{and } f_Y(x) = 0 \text{ otherwise}$$

The cumulative distribution of r.v. Y with rate $\lambda \in \mathbb{R}_{>0}$ is:

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

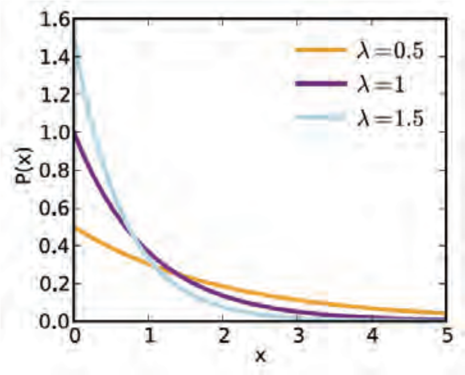
The rate $\lambda \in \mathbb{R}_{>0}$ uniquely determines an exponential distribution.

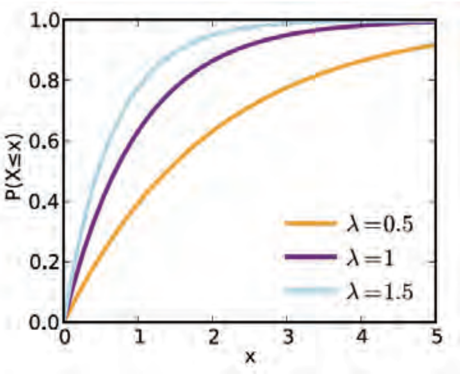
Variance and expectation

Let r.v. Y be exponentially distributed with rate $\lambda \in \mathbb{R}_{>0}$. Then:

Expectation $E[Y] = \frac{1}{\lambda}$ and variance $Var[Y] = \frac{1}{\lambda^2}$

Exponential Distribution Functions

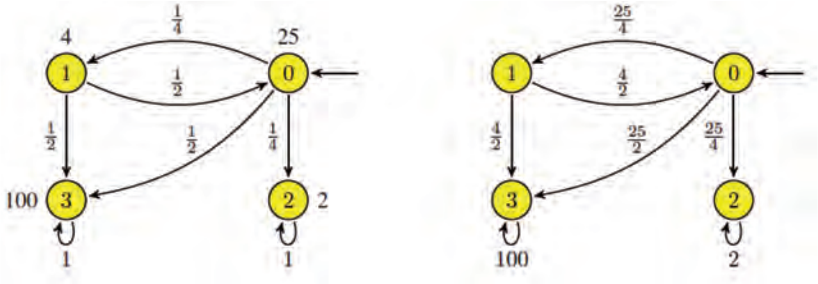




The higher the rate λ, the faster the cdf approaches 1.

Continuous-Time Markov Chains

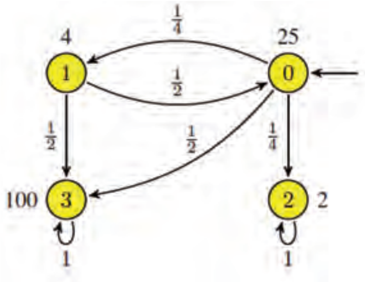
A CTMC is a DTMC with an *exit rate* function $r : S \to \mathbb{R}_{>0}$ where $r(s)$ is the rate of an exponential distribution.

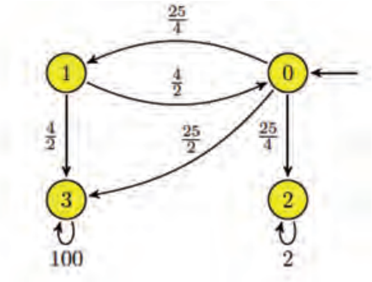


$r(0) = 25,\ r(1) = 4,\ r(2) = 2$ and $r(3) = 100$

A Classical Perspective

A CTMC is a DTMC where transition probability function $\mathbf{P}$ is replaced by a *transition rate* function $\mathbf{R}$. We have $\mathbf{R}(s, s') = \mathbf{P}(s, s') \cdot r(s)$.





$r(0) = 25$, $r(1) = 4$, $r(2) = 2$ and $r(3) = 100$

CTMC Semantics

State-to-state timed transition probability

The probability to *move* from non-absorbing s to s' in $[0, t]$ is:

$$\frac{\mathbf{R}(s, s')}{r(s)} \cdot \left(1 - e^{-r(s) \cdot t}\right).$$

Residence time distribution

The probability to *take some* outgoing transition from s in $[0, t]$ is:

$$\int_0^t r(s) \cdot e^{-r(s) \cdot x} \, dx \; = \; 1 - e^{-r(s) \cdot t}$$

Zenoness

Zeno theorem

In every CTMC, almost surely no Zeno runs occur.

In contrast to timed automata verification, Zeno runs thus pose no problem.

Timed Reachability Probabilities

Problem statement

Consider an MC with finite state space S, $s \in S$, $t \in \mathbb{R}_{\geqslant 0}$ and $G \subseteq S$.

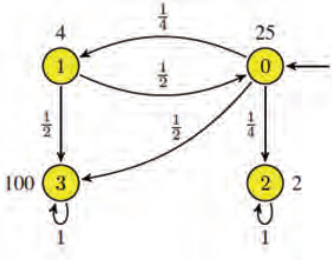
Aim: determine $\Pr(s \vDash \diamond^{\leqslant t} G)$.

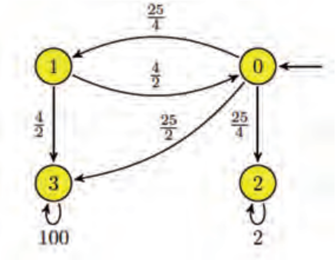
Characterisation of timed reachability probabilities

- Let function $x_s(t) = \Pr(s \vDash \diamond^{\leqslant t} G)$ for any state s
  - if G is not reachable from s, then $x_s(t) = 0$ for all t
  - if $s \in G$ then $x_s(t) = 1$ for all t
- For any state $s \in Pre^*(G) \setminus G$:

$$x_s(t) = \int_0^t \sum_{s' \in S} \underbrace{\mathbf{R}(s,s') \cdot e^{-r(s) \cdot x}}_{\substack{\text{probability to move to} \\ \text{state } s' \text{ at time } x}} \cdot \underbrace{x_{s'}(t-x)}_{\substack{\text{prob. to fulfill} \\ \diamond^{\leqslant t-x}\, G \text{ from } s'}} \, dx$$

Timed Reachability Probabilities





Integral equations for $\Diamond^{\leqslant 10}\, 2$:

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:

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

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

Timed Reachability Probabilities

Reachability probabilities

Can be obtained by solving a system of linear equations for which many efficient techniques exist.

Timed reachability probabilities

Can be obtained by solving a system of Volterra integral equations. This is in general non-trivial, inefficient, and has several pitfalls such as numerical stability.

Solution

Reduce the problem of computing $\Pr(s \models \Diamond^{\leqslant t} G)$ to an alternative problem for which well-known efficient techniques exist: computing transient probabilities.

Timed Reachability Probabilities = Transient Probabilities

Aim

Compute $\Pr(s \models \Diamond^{\leqslant t} G)$ in CTMC $\mathcal{C}$. Observe that once a path π reaches G within t time, then the remaining behaviour along π is not important. This suggests to make all states in G absorbing. This yields $\mathcal{C}[G]$

Theorem

$$\underbrace{\Pr(s \models \Diamond^{\leqslant t} G)}_{\text{timed reachability in } \mathcal{C}} = \underbrace{\Pr(s \models \Diamond^{= t} G)}_{\text{timed reachability in } \mathcal{C}[G]} = \underbrace{\vec{p}(t) \text{ with } \vec{p}(0) = \mathbf{1}_s}_{\text{transient prob. in } \mathcal{C}[G]}.$$

Transient probabilities can be efficiently computed as solutions of linear differential equations.

Computing Transient Probabilities

By solving a linear differential equation system

The transient probability vector $\underline{p}(t) = (p_{s_1}(t), \ldots, p_{s_k}(t))$ satisfies:

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

where $\mathbf{r}$ is the diagonal matrix of vector $\underline{r}$.

Solution using standard knowledge yields: $\underline{p}(t) \;=\; \underline{p}(0) \cdot e^{(\mathbf{R}-\mathbf{r}) \cdot t}$.

Computing the matrix exponential is a challenging numerical problem[1].

[1] Nineteen dubious ways of computing a matrix exponential (1978 and 2003)

Computing Transient Probabilities: Example

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

Uniformization

CTMC $\mathcal{C}$ is uniform if $r(s) = r$ for all $s \in S$ for some $r \in \mathbb{R}_{>0}$.

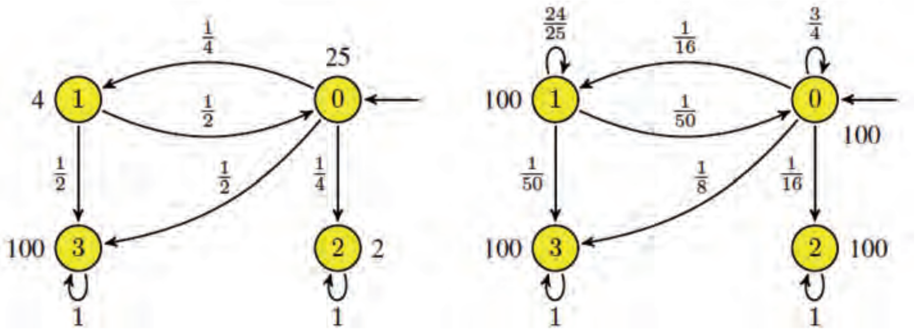
Uniformization [Gross and Miller, 1984]

Let $r \in \mathbb{R}_{>0}$ such that $r \geqslant max_{s \in S}\ r(s)$. Then $\overline{r}(\mathcal{C})$ is the CTMC $\mathcal{C}$ with two changes: $\overline{r}(s) = r$ for all $s \in S$, and:

$$\overline{\mathbf{P}}(s,s') = \frac{r(s)}{r} \cdot \mathbf{P}(s,s') \text{ if } s' \neq s \quad \text{and} \quad \overline{\mathbf{P}}(s,s) = \frac{r(s)}{r} \cdot \mathbf{P}(s,s) + 1 - \frac{r(s)}{r}.$$

$\overline{\mathbf{P}}$ is a stochastic matrix and $\overline{r}(\mathcal{C})$ is uniform.

Uniformization: Example



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

The Benefit of Uniformization

Transient probabilities of a CTMC and its uniformized CTMC coincide.

Thus: $\underbrace{\underline{p}(t) \;=\; \underline{p}(0)\cdot e^{(\mathbf{R}-\mathbf{r})\cdot t}}_{\text{transient probablity in } \mathcal{C}} \;=\; \underbrace{\underline{p}(0)\cdot e^{(\overline{\mathbf{R}}-\overline{\mathbf{r}})\cdot t}}_{\text{transient probablity in } \overline{r}(\mathcal{C})} \;=\; \underline{p}(0)\cdot e^{-r\cdot t}\cdot e^{r\cdot t\cdot\overline{\mathbf{P}}}$

Still a matrix exponential remains. Did we gain anything?

Yes. Since $\overline{\mathbf{P}}$ is stochastic, Taylor-Maclaurin yields $\sum_i \ldots \overline{\mathbf{P}}^i$.

Computing Poisson probabilities is done using tailored Fox-Glynn algorithm.

Other CTMC Properties

- Expected time and (unbounded) reachability objectives

  Can be characterised as solution of set of linear equations

- Long-run average objectives
  1. Determine the limiting distribution in any terminal SCC
  2. Take weighted sum with reachability probabilities of terminal SCCs

- Probabilistic timed CTL model checking

  recursive descent over parse tree; key is timed reachability

- Deterministic timed automata objectives
  1. Determine the Zone automaton of the timed automaton
  2. Take the product of the Markov chain and this Zone automaton[2]
  3. Determine the probability to reach an "accepting" zone

[2] This yields a piecewise deterministic Markov process.

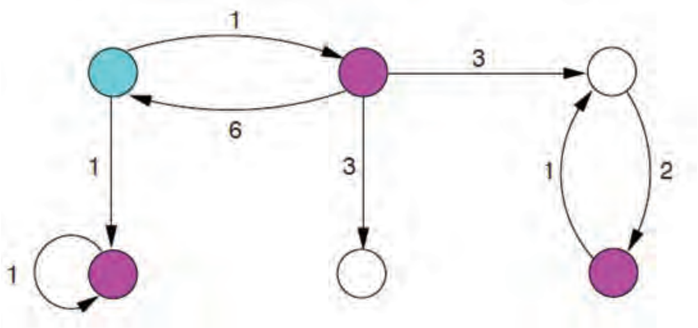
Long-Run Probabilities

$s \models \mathbb{L}_{\leqslant p}(\Phi)$ iff the probability to be in a Φ-state on the long run (when starting in s) is at most p.

Model-checking algorithm:

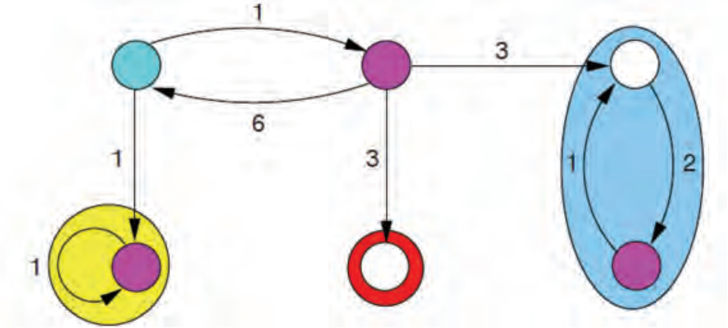
1. for each terminal SCC T:
   1.1 determine the steady-state probabilities $\pi^T(t)$ of each Φ-state t
   1.2 determine the reachability probability of T from s
2. check whether $\sum_T \left(\Pr(s \models \Diamond T) \cdot \sum_{t \in T \wedge t \models \Phi} \pi^T(t) \right) \leqslant p$

Long-Run: Example



Does the long-run fraction of time to be in a purple-state exceed 75%?

Long-Run: Example



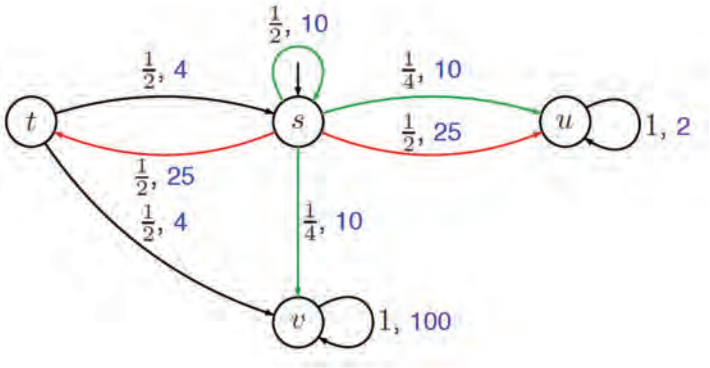
$$s \models \mathbb{L}_{>\frac{3}{4}}(purple) \quad \text{iff} \quad \Pr(s \models \Diamond yellow) \cdot \pi^{yellow}(purple) + \Pr(s \models \Diamond blue) \cdot \pi^{blue}(purple) > \frac{3}{4}$$

Overview

What is a CTMDP?

A CTMDP is an MDP plus an exit-rate function $E : S \times Act \to \mathbb{R}_{\geqslant 0}$.

Or, equivalently, a CTMDP is a CTMC with non-determinism.



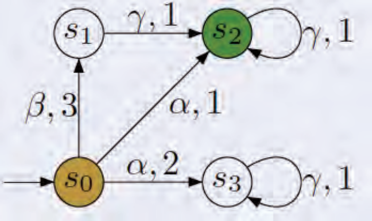
Maximal Timed Reachability

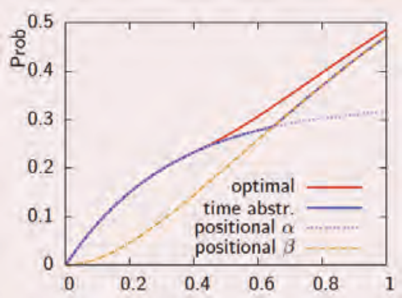
Characterisation of timed reachability probabilities

- Let function $x_s(t) = \Pr^{\max}(s \models \diamond^{\leqslant t} G)$ for any state s
  - if G is not reachable from s, then $x_s(t) = 0$ for all t
  - if $s \in G$ then $x_s(t) = 1$ for all t
- For any state $s \in Pre^*(G) \setminus G$:

$$x_s(t) = \max_{\alpha \in Act(s)} \int_0^t \sum_{s' \in S} \underbrace{\mathbf{R}(s,\alpha,s') \cdot e^{-r(s,\alpha)\cdot x}}_{\substack{\text{probability to move to}\\ \text{state } s' \text{ at time } x\\ \text{under action } \alpha}} \cdot \underbrace{x_{s'}(t-x)}_{\substack{\text{max. prob.}\\ \text{to fulfill } \diamond^{\leqslant t-x} G\\ \text{from } s'}} \, dx$$

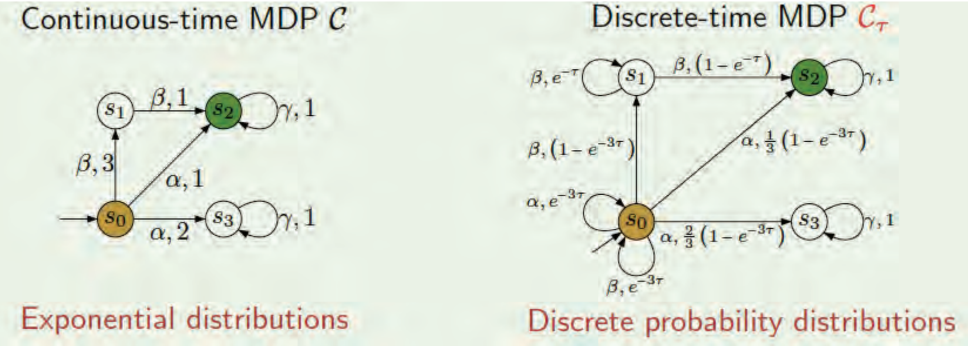
Timed Reachability Requires Timed Policies





- Timed positional policies are optimal; any simpler policy is inferior.
- If long time remains: choose β; if short time remains: choose α.
- Optimal for deadline 1: choose α if $1 - t_0 \leqslant \ln 3 - \ln 2$, otherwise β

Discretisation



$$\underbrace{\text{Reachability in } d \text{ time}}_{\text{in CTMDP}} \approx \underbrace{\text{Reachability in } \frac{d}{\tau} \text{ steps}}_{\text{in corresponding MDP}}$$

This analysis yields ϵ-optimal policies.

Bounding the Imprecision [Neuhäusser & Zhang, 2010]

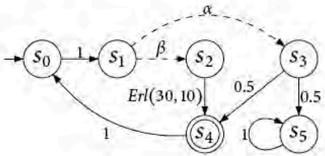
Let r be the CTMDP's maximal rate, deadline d with $d = k \cdot \tau$ for $k \in \mathbb{N}_{>0}$.

$$\overset{\max}{\Pr}(s \models \Diamond^{\leqslant k} G) \leqslant \underbrace{\overset{\max}{\Pr}(s \models \Diamond^{\leqslant d} G)}_{\text{timed reachability}} \leqslant \overset{\max}{\Pr}(s \models \Diamond^{\leqslant k} G) + \underbrace{\frac{(r \cdot \tau)^2}{2k}}_{\text{error}}$$

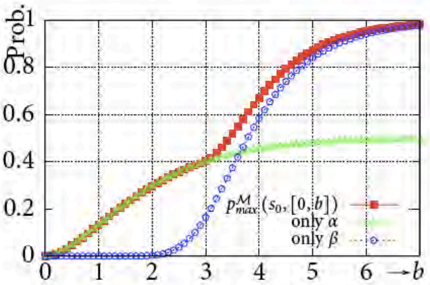
The maximal reachability probabilities can be computed by k value iterations.

For error ϵ, the step bound k lies in $\mathcal{O}\left((r \cdot d)^2 / \epsilon\right)$.

A Small Example



(a) The $Erl(30,10)$ model $\mathcal{M}$.

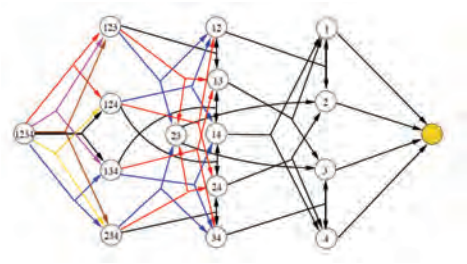


(b) Time-bounded reachability in $\mathcal{M}$.

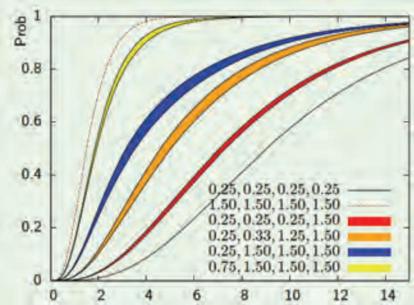
problem	states	ε	λ	b	prob.	time
$Erl(30,10)$	35	10^{-3}	10	4	0.672	50s
$Erl(30,10)$	35	10^{-3}	10	7	0.983	70s
$Erl(30,10)$	35	10^{-4}	10	4	0.6718	268s

(c) Computation times for different parameters.

Stochastic Scheduling Results



The stochastic scheduling problem



Minimal and maximal timed reachability (x-axis = d)

The SEPT policy turns out to be optimal for this example

Uniform CTMDPs

[Baier *et al.*, 2004]

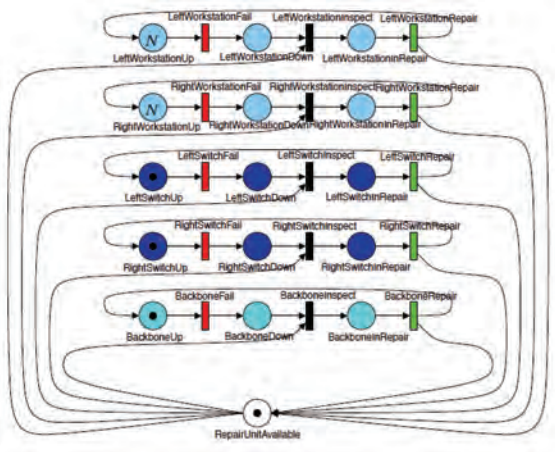
N	states	100 h	1000 h	5000 h	10000 h	50000 h
1	110	0s 0.00	0s 0.01	0s 0.04	0s 0.09	0s 0.36
4	818	0s 0.00	0s 0.02	0s 0.09	0s 0.18	1s 0.62
16	10130	0s 0.01	0s 0.08	1s 0.32	1s 0.54	4s 0.98
64	151058	0s 0.03	1s 0.23	5s 0.73	10s 0.93	46s 1.00
256	2373650	7s 0.05	40s 0.43	2m 46s 0.94	5m 16s 1.00	24m 31s 1.00

MRMC run times and probabilities for fault-tolerant GSPN workstation cluster.

Increrasing the uniformization rate improves optimality; in the limit, yields ϵ-optimality.[3]

[3] Hermanns *et al.* (2015) showed that this algorithm is mostly performing the best.

Stochastic Petri Net of Workstation Cluster



Average durations:

BackboneFail	5000h
LeftSwitchFail	4000h
RightSwitchFail	4000h
RightWSFail	500h
LeftWSFail	500h
BackboneRepair	8h
RightSwitchRepair	4h
LeftSwitchRepair	4h
RightWSRepair	0.5h
LeftWSRepair	0.5h

when several units have failed, repair unit is assigned nondeterministically.

Alternative Properties

- Expected time and reachability objectives

  can be solved by standard MDP algorithms for reachability
- Reward-bounded properties

  can be reduced to time-bounded reachability properties
  by exploiting the duality between progress of time and reward gain
- Long-run average properties

  can be reduced to long-run ratio objectives in MDPs
- Deterministic 1-clock timed automata

  can be reduced to reachability probabilities in PDP-decision processes

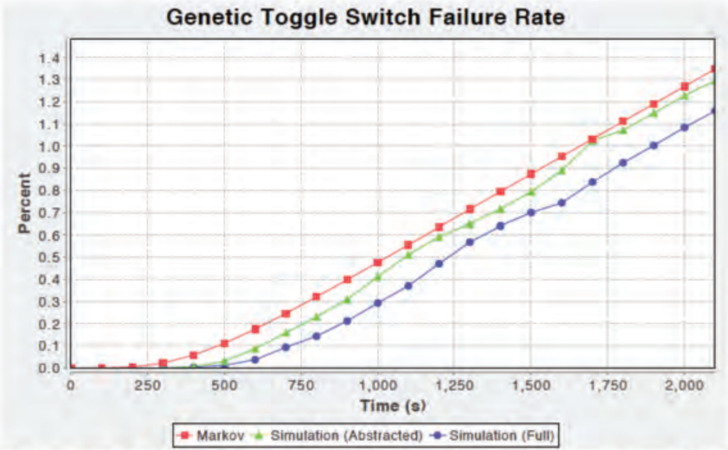
Probabilistic Model Checkers

- PRISM [Kwiatkowska, Parker *et al.*]
- MRMC [Katoen, Hermanns *et al.*]
- iscasMC [Zhang *et al.*]
- LiQuor [Baier *et al.*]
- iBioSim [Myers *et al.*]
- GreatSPN [Franceschinis *et al.*]
- SMART [Ciardo, Miner *et al.*]
- MarCie [Heiner *et al.*]
- PAT [Song, Dong *et al.*]
- SToRM (under development)
- ......

Statistical model checkers: Ymer, Vesta, UppAal, APMC, PlasmaLab, ...

Model Checking Times (CTMCs) [Madsen, Myers *et al.*, 2014]

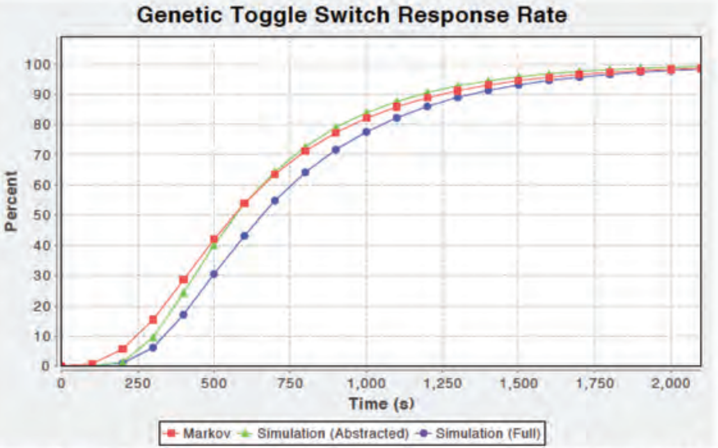
$$\mathtt{Pr}\{\Diamond^{[0,2100]}\,\mathrm{LacI} < 20 \wedge \mathrm{TetR} > 40\}$$



Simulation time: 43 min. (Full), 3 min. 15 sec. (Abstracted), <1 sec. (Markov).

Model Checking Times (CTMCs) [Madsen, Myers *et al.*, 2014]

$$\Pr\{\Diamond^{[0,2100]}\, \mathrm{LacI} < 20 \wedge \mathrm{TetR} > 40\}$$



Simulation time: 3 hours 12 min. (Full), 1 min. (Abstracted), 0.5 sec. (Markov).