Overview

3 Motivation

2 What are discrete-time Markov chains?

3 Reachability probabilities

4 Qualitative reachability and all that

5 Verifying probabilistic CTL

6 Expressiveness of probabilistic CTL

7 Probabilistic bisimulation

8 Verifying $\omega$-regular properties

Probabilities help

- When analysing system performance and dependability
  - to quantify arrivals, waiting times, time between failure, QoS, ...

- When modelling unreliable and unpredictable system behavior
  - to quantify message loss, processor failure
  - to quantify unpredictable delays, express soft deadlines, ...

- When building protocols for networked embedded systems
  - randomized algorithms

- When problems are undecidable deterministically
  - repeated reachability of lossy channel systems, ...
Illustrative example: Security

Security: Crowds protocol

A protocol for *anonymous web browsing* (variants: mCrowds, BT-Crowds)
- Hide user’s communication by *random routing* within a crowd
  - sender selects a crowd member randomly using a uniform distribution
  - selected router flips a biased coin:
    - with probability $1 - p$: direct delivery to final destination
    - otherwise: select a next router randomly (uniformly)
  - once a routing path has been established, use it until crowd changes
- Rebuild routing paths on crowd changes
- Property: Crowds protocol ensures “probable innocence”:
  - probability real sender is discovered $< \frac{1}{2}$ if $N \geq p \cdot (c+1)$
  - where $N$ is crowd’s size and $c$ is number of corrupt crowd members

Illustrative example: Leader election

Distributed system: Leader election

A round-based protocol in a synchronous ring of $N > 2$ nodes
- the nodes proceed in a *lock-step* fashion
  - each slot = 1 message is read + 1 state change + 1 message is sent
  - this synchronous computation yields a discrete-time Markov chain
- Each round starts by each node choosing a uniform id $\in \{1, \ldots, K\}$
- Nodes pass their selected id around the ring
- If there is a unique id, the node with the maximum unique id is leader
- If not, start another round and try again . . .

Properties of leader election

Almost surely eventually a leader will be elected

$P = 1$ (◊ leader elected)

With probability at least 0.8, a leader is elected within $k$ steps

$P \geq 0.8$ (◊ $\leq k$ leader elected)
Verifying Continuous-Time Markov Chains

Motivation

What is probabilistic model checking?

Joost-Pieter Katoen

Probabilistic models

<table>
<thead>
<tr>
<th></th>
<th>Nondeterminism</th>
<th>Nondeterminism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete time</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Markov decision process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete-time Markov chain (DTMC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous time</td>
<td>CTMC</td>
<td>CTMDP</td>
</tr>
</tbody>
</table>

Other models: probabilistic variants of (priced) timed automata, or hybrid automata

Probability theory is simple, isn’t it?

In no other branch of mathematics is it so easy to make mistakes as in probability theory

Geometric distribution

Let $X$ be a discrete random variable, natural $k > 0$ and $0 < p \leq 1$. The mass function of a geometric distribution is given by:

$$Pr\{X = k\} = (1 - p)^{k-1} \cdot p$$

We have $E[X] = \frac{1}{p}$ and $Var[X] = \frac{1 - p}{p^2}$ and cdf $Pr\{X \leq k\} = 1 - (1 - p)^k$.

Memoryless property

**Theorem**

1. For any random variable $X$ with a geometric distribution:

   $$Pr(X = k + m \mid X > m) = Pr(X = k) \quad \text{for any} \quad m \in T, k \geq 1$$

   This is called the memoryless property, and $X$ is a memoryless r.v.

2. Any discrete random variable which is memoryless is geometrically distributed.

Markov property

A discrete-time stochastic process $\{X(t) \mid t \in T\}$ over state space $\{d_0, d_1, \ldots\}$ is a Markov process if for any $t_0 < t_1 < \ldots < t_n < t_{n+1}$:

$$Pr\{X(t_{n+1}) = d_{n+1} \mid X(t_0) = d_0, X(t_1) = d_1, \ldots, X(t_n) = d_n\} = Pr\{X(t_{n+1}) = d_{n+1} \mid X(t_n) = d_n\}$$

The distribution of $X(t_{n+1})$, given the values $X(t_0)$ through $X(t_n)$, only depends on the current state $X(t_n)$.

The next state of the stochastic process only depends on the current state, and not on states assumed previously. This is the Markov property.
Invariance to time-shifts

Time homogeneity

Markov process \( \{ X(t) \mid t \in T \} \) is \textit{time-homogeneous} iff for any \( t' < t \):
\[
Pr\{ X(t) = d \mid X(t') = d' \} = Pr\{ X(t - t') = d \mid X(0) = d' \}.
\]
A time-homogeneous stochastic process is invariant to time shifts.

Discrete-time Markov chain

A \textit{discrete-time Markov chain} (DTMC) is a time-homogeneous Markov process with discrete parameter \( T \) and discrete state space \( S \).

Transition probability matrix

Let \( P \) be a function with \( P(s_i, s_j) = p(s_i, s_j) \). For finite state space \( S \), function \( P \) is called the \textit{transition probability matrix} of the DTMC with state space \( S \).

Properties

1. \( P \) is a (right) \textit{stochastic} matrix, i.e., it is a square matrix, all its elements are in \([0, 1]\), and each row sum equals one.
2. \( P \) has an eigenvalue of one, and all its eigenvalues are at most one.
3. For all \( n \in \mathbb{N} \), \( P^n \) is a stochastic matrix.

Transition probabilities

The \textit{(one-step) transition probability} from \( s \in S \) to \( s' \in S \) at epoch \( n \in \mathbb{N} \) is given by:
\[
p^{(n)}(s, s') = Pr\{ X_{n+1} = s' \mid X_n = s \} = Pr\{ X_1 = s' \mid X_0 = s \}
\]
where the last equality is due to time-homogeneity.

Since \( p^{(n)}(\cdot) = p^{(k)}(\cdot) \), the superscript \( (n) \) is omitted, and we write \( p(\cdot) \).
Simulating a die by a fair coin [Knuth & Yao]

Heads = “go left”; tails = “go right”. Does this DTMC adequately model a fair six-sided die?

Craps

- Roll two dice and bet
- Come-out roll (“pass line” wager):
  - outcome 7 or 11: win
  - outcome 2, 3, or 12: lose (“craps”)
  - any other outcome: roll again (outcome is “point”)
- Repeat until 7 or the “point” is thrown:
  - outcome 7: lose (“seven-out”)
  - outcome the point: win
  - any other outcome: roll again

A DTMC model of Craps

- Come-out roll:
  - 7 or 11: win
  - 2, 3, or 12: lose
  - else: roll again
- Next roll(s):
  - 7: lose
  - point: win
  - else: roll again
State residence time distribution

Let $T_s$ be the number of epochs of DTMC $\mathcal{D}$ to stay in state $s$:

$$
Pr\{T_s = 1\} = 1 - P(s, s) \\
Pr\{T_s = 2\} = P(s, s) \cdot (1 - P(s, s)) \\
\ldots \ldots \\
Pr\{T_s = n\} = P(s, s)^{n-1} \cdot (1 - P(s, s))
$$

So, the state residence times in a DTMC obey a geometric distribution.

The expected number of time steps to stay in state $s$ equals $E[T_s] = \frac{1}{1-P(s, s)}$.

The variance of the residence time distribution is $\text{Var}[T_s] = \frac{P(s, s)}{(1-P(s, s))^2}$.

Recall that the geometric distribution is the only discrete probability distribution that possesses the memoryless property.

Determining $n$-step transition probabilities

$n$-step transition probabilities

The probability to move from $s$ to $s'$ in $n \in \mathbb{N}$ steps is inductively defined:

$$p_{s,s'}(0) = 1 \text{ if } s = s', \text{ and 0 otherwise,}$$

$$p_{s,s'}(1) = P(s, s'), \text{ and for } n > 1 \text{ by the Chapman-Kolmogorov equation:}$$

$$p_{s,s'}(n) = \sum_{s''} p_{s,s''}(l) \cdot p_{s'',s'}(n-l) \text{ for all } 0 < l < n$$

For $l = 1$ and $n > 0$ we obtain:

$$p_{s,s'}(n) = \sum_{s''} p_{s,s''}(1) \cdot p_{s'',s'}(n-1)$$

$$p^{(n)} = p^{(1)} \cdot p^{(n-1)} = p \cdot p^{(n-1)}$$ is the $n$-step transition probability matrix.

Repeating this scheme: $p^{(n)} = P \cdot P^{(n-1)} = \ldots = P^{n-1} \cdot P^{(1)} = P^n$.

Overview

1. Motivation
2. What are discrete-time Markov chains?
3. Reachability probabilities
4. Qualitative reachability and all that
5. Verifying probabilistic CTL
6. Expressiveness of probabilistic CTL
7. Probabilistic bisimulation
8. Verifying $\omega$-regular properties
Paths in a DTMC

State graph

The state graph of DTMC $D$ is a digraph $G = (V, E)$ with $V$ are the states of $D$, and $(s, s') \in E$ iff $P(s, s') > 0$.

Paths

Paths in $D$ are maximal (i.e., infinite) paths in its state graph. Thus, a path is an infinite sequence of states $s_0 s_1 s_2 \ldots$ with $P(s_i, s_{i+1}) > 0$ for all $i$.

Let $\text{Paths}(D)$ denote the set of paths in $D$, and $\text{Paths}^*(D)$ the set of finite prefixes thereof.

Direct successors and predecessors

$\text{Post}(s) = \{ s' \in S \mid P(s, s') > 0 \}$ and $\text{Pre}(s) = \{ s' \in S \mid P(s', s) > 0 \}$ are the set of direct successors and predecessors of $s$ respectively. $\text{Post}^*(s)$ and $\text{Pre}^*(s)$ are the reflexive and transitive closure of $\text{Post}$ and $\text{Pre}$.

Measurable space

Sample space

A sample space $\Omega$ of a chance experiment is a set of elements that have a 1-to-1 relationship to the possible outcomes of the experiment.

$\sigma$-algebra

A $\sigma$-algebra is a pair $(\Omega, F)$ with $\Omega \neq \emptyset$ and $F \subseteq 2^\Omega$ a collection of subsets of sample space $\Omega$ such that:

1. $\Omega \in F$
2. $A \in F \Rightarrow \Omega - A \in F$ complement
3. $(\forall i \geq 0. A_i \in F) \Rightarrow \bigcup_{i \geq 0} A_i \in F$ countable union

The elements in $F$ of a $\sigma$-algebra $(\Omega, F)$ are called events. The pair $(\Omega, F)$ is called a measurable space.

Probability space

A probability space $\mathcal{P}$ is a structure $(\Omega, F, Pr)$ with:

1. $(\Omega, F)$ is a $\sigma$-algebra, and
2. $Pr : F \to [0, 1]$ is a probability measure, i.e.:
   1. $Pr(\Omega) = 1$, i.e., $\Omega$ is the certain event
   2. $Pr\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} Pr(A_i)$ for any $A_i \in F$ with $A_i \cap A_j = \emptyset$ for $i \neq j$,

The elements in $F$ of a probability space $(\Omega, F, Pr)$ are called measurable events.
Paths and probabilities

To reason quantitatively about the behavior of a DTMC, we need to define a probability space over its paths.

Intuition

For a given state $s$ in DTMC $D$:
- Sample space := set of all infinite paths starting in $s$
- Events := sets of infinite paths starting in $s$
- Basic events := cylinder sets
- Cylinder set of finite path $\hat{\pi} :=$ set of all infinite continuations of $\hat{\pi}$

Probability measure on DTMCs

Cylinder set

The cylinder set of finite path $\hat{\pi} = s_0 s_1 \ldots s_n \in Paths^* (D)$ is defined by:

$$Cyl(\hat{\pi}) = \{ \pi \in Paths (D) | \hat{\pi} \text{ is a prefix of } \pi \}$$

The cylinder set spanned by finite path $\hat{\pi}$ thus consists of all infinite paths that have prefix $\hat{\pi}$. Cylinder sets serve as basic events of the smallest $\sigma$-algebra on $Paths(D)$.

$\sigma$-algebra of a DTMC

The $\sigma$-algebra associated with DTMC $D$ is the smallest $\sigma$-algebra that contains all cylinder sets $Cyl(\hat{\pi})$ where $\hat{\pi}$ ranges over all finite path fragments in $D$.

Some events of interest

Let DTMC $D$ with (possibly infinite) state space $S$.

(Simple) reachability

Eventually reach a state in $G \subseteq S$. Formally:

$$\Diamond G = \{ \pi \in Paths (D) | \exists i \in \mathbb{N}. \pi[i] \in G \}$$

Invariance, i.e., always stay in state in $G$:

$$\Box G = \{ \pi \in Paths (D) | \forall i \in \mathbb{N}. \pi[i] \in G \} = \overline{\Diamond G}.$$

Constrained reachability

Or "reach-avoid" properties where states in $F \subseteq S$ are forbidden:

$$\overline{F} \cup G = \{ \pi \in Paths (D) | \exists i \in \mathbb{N}. \pi[i] \in G \wedge \forall j < i. \pi[j] \notin F \}$$
### Repeated events of interest

#### Repeated reachability

Repeatedly visit a state in $G$; formally:

$$\square \Diamond G = \{ \pi \in \text{Paths}(D) \mid \forall i \in \mathbb{N}. \exists j \geq i. \pi[j] \in G \}$$

#### Persistence

Eventually reach in a state in $G$ and always stay there; formally:

$$\Diamond \square G = \{ \pi \in \text{Paths}(D) \mid \exists i \in \mathbb{N}. \forall j \geq i. \pi[j] \in G \}$$

---

### Measurability

**Measurability theorem**

Events $\Diamond G$, $\square G$, $\mathcal{F} U G$, $\square \Diamond G$ and $\Diamond \square G$ are measurable on any DTMC.

**Proof:**

To show this, every event will be expressed as allowed operations (complement and/or countable unions) of the events — our cylinder sets! — in the $\sigma$-algebra on infinite paths in a DTMC.

Note that $\square G = \Diamond G$ and $\square \square G = \Diamond \Diamond G$.

It remains to prove the measurability for the remaining three cases.

---

### Reachability probabilities: Knuth’s die

Consider the event $\Diamond 4$.

Using the previous theorem we obtain:

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

This yields:

$$P(s_0 s_2 s_4) + P(s_0 s_2 s_4 s_2) s_4 + \ldots$$

Or:

$$\sum_{k=0}^{\infty} P(s_0 s_2 s_4)^k s_4$$

Or:

$$\frac{1}{8} \sum_{k=0}^{\infty} \left(\frac{1}{4}\right)^k$$

Geometric series: $\frac{1}{8} \left(1 - \frac{1}{4}\right) = \frac{1}{8} \frac{3}{4} = \frac{1}{6}$

There is however a simpler way to obtain reachability probabilities!
Reachability probabilities in finite DTMCs

Problem statement
Let $D$ be a DTMC with finite state space $S$, $s \in S$ and $G \subseteq S$.
Aim: determine $Pr(s \models G) = Pr_s(\Diamond G) = Pr_s\{ \pi \in \text{Paths}(s) \mid \pi \models G \}$
where $Pr_s$ is the probability measure in $D$ with single initial state $s$.

Characterisation of reachability probabilities

- Let variable $x_s = Pr(s \models G)$ 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 $s \in \text{Pre}^*(G) \setminus G$:
  \[
  x_s = \sum_{t \in S \setminus G} P(s, t) \cdot x_t + \sum_{u \in G} P(s, u) \]
  \[
  \text{reach } G \text{ via } t \in S \setminus G \quad \text{and } \quad \text{reach } G \text{ in one step}
  \]

Reachability probabilities as linear equation system

- Let $S_I = \text{Pre}^*(G) \setminus G$, the states that can reach $G$ by $\geq 0$ steps
- $A = (P(s, t))_{s, t \in S_I}$, the transition probabilities in $S_I$
- $b = (b_s)_{s \in S_I}$, the probs to reach $G$ in 1 step, i.e., $b_s = \sum_{u \in G} P(s, u)$

Then: $x = (x_s)_{s \in S_I}$, with $x_s = Pr(s \models G)$ is the unique solution of:

\[
 x = A \cdot x + b \quad \text{or} \quad (I - A) x = b
\]
where $I$ is the identity matrix of cardinality $|S_I| \times |S_I|$. 

Reachability probabilities: Knuth’s die

- Consider the event $\Diamond 4$
- Using the previous characterisation we obtain:
  \[
  x_1 = x_2 = x_3 = x_5 = x_6 = 0 \quad \text{and} \quad x_4 = 1
  \]
  \[
  x_1 = x_3 = x_5 = 0 \\
  x_2 = \frac{1}{2} x_4 \\
  x_3 = \frac{1}{2} x_5 + \frac{1}{2} x_6 \\
  x_4 = \frac{1}{4} x_7 + \frac{1}{4} x_4 \\
  x_5 = \frac{1}{2} x_6 + \frac{1}{2} x_6
  \]
- Gaussian elimination yields:
  \[
  x_3 = \frac{1}{3}, \quad x_2 = \frac{1}{3}, \quad x_6 = \frac{1}{6}, \quad \text{and} \quad x_4 = \frac{1}{8}
  \]
Constrained reachability probabilities

Problem statement
Let $\mathcal{D}$ be a DTMC with finite state space $S$, $s \in S$ and $F, G \subseteq S$.
Aim: $Pr(s \models F \cup G) = Pr_s(F \cup G) = Pr_s\{ \pi \in \text{Paths}(s) \mid \pi \models F \cup G \}$
where $Pr_s$ is the probability measure in $\mathcal{D}$ with single initial state $s$.

Characterisation of constrained reachability probabilities

- Let variable $x_s = Pr(s \models F \cup G)$ for any state $s$
  - if $G$ is not reachable from $s$ via $F$, then $x_s = 0$
  - if $s \in G$ then $x_s = 1$
- For any state $s \in (\text{Pre}^*(G) \cap \overline{F}) \setminus G$:
  $$x_s = \sum_{t \in S \setminus G} P(s, t) \cdot x_t + \sum_{u \in G} P(s, u)$$

Iteratively computing reachability probabilities

Theorem
The vector $x = \left( Pr(s \models F \cup G) \right)_{s \in S}$ is the unique solution of:
$$y = A \cdot y + b$$
with $A$ and $b$ as defined before.
Furthermore, let:
$$x^{(0)} = 0 \quad \text{and} \quad x^{(i+1)} = A \cdot x^{(i)} + b \quad \text{for} \quad 0 \leq i.$$ 

Then:
1. $x^{(n)}(s) = Pr(s \models F \cup U \leq n G)$ for $s \in S$
2. $x^{(0)} \leq x^{(1)} \leq x^{(2)} \leq \ldots \leq x$
3. $x = \lim_{n \to \infty} x^{(n)}$
where $F \cup U \leq n G$ contains those paths that reach $G$ via $F$ within $n$ steps.

Remark
Iterative algorithms to compute $x$

There are various algorithms to compute $x = \lim_{n \to \infty} x^{(n)}$ where:
$$x^{(0)} = 0 \quad \text{and} \quad x^{(i+1)} = A \cdot x^{(i)} + b \quad \text{for} \quad 0 \leq i.$$ 

The Power method computes vectors $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ and aborts if:
$$\max_{s \in S} |x^{(n+1)}(s) - x^{(n)}(s)| < \varepsilon \quad \text{for some small tolerance} \quad \varepsilon$$
This technique guarantees convergence.
Alternative iterative techniques: e.g., Jacobi or Gauss-Seidel, successive overrelaxation (SOR).

Example: Knuth’s die

- Let $G = \{ 1, 2, 3, 4, 5, 6 \}$
- Then $Pr(s_0 \models \diamond G) = 1$
- And $Pr(s_0 \models \diamond \leq k G)$ for $k \in \mathbb{N}$ is given by:
Reachability probability = transient probabilities

Aim
Compute \( Pr(\Diamond^n G) \) in DTMC \( D \). Observe that once a path \( \pi \) reaches \( G \), then the remaining behaviour along \( \pi \) is not important. This suggests to make all states in \( G \) absorbing.

\[
Pr(\Diamond^n G) = Pr(\Diamond^n G) = \pi_{init} \cdot P^n_G = \Theta^n_D[G]
\]
reachability in \( D \)  
reachability in \( D[G] \) in \( D[G] \)

Lemma
\[
Pr(F \cup^n G) = Pr(\Diamond^n G) = \pi_{init} \cdot P^n_{F \cup G} = \Theta^n_D[F \cup G]
\]
reachability in \( D \)  
reachability in \( D[F \cup G] \) in \( D[F \cup G] \)

Overview
- Motivation
- What are discrete-time Markov chains?
- Reachability probabilities
- Qualitative reachability and all that
- Verifying probabilistic CTL
- Expressiveness of probabilistic CTL
- Probabilistic bisimulation
- Verifying \( \omega \)-regular properties

Constrained reachability = transient probabilities

Aim
Compute \( Pr(\bar{F} \cup^n G) \) in DTMC \( D \). Observe (as before) that once a path \( \pi \) reaches \( G \) via \( F \), then the remaining behaviour along \( \pi \) is not important. Now also observe that once \( s \in F \setminus G \) is reached, then the remaining behaviour along \( \pi \) is not important. This suggests to make all states in \( G \) and \( F \setminus G \) absorbing.

Overview
- Qualitative properties

Qualitative properties
- Quantitative properties
Comparing the probability of an event such as \( \square G \), \( \Diamond G \) and \( \square \Diamond G \) with a threshold \( \sim p \) with \( p \in (0,1) \) and \( \sim \) a binary comparison operator \( (\,\;=,\;\langle,\;\leq,\;\rangle,\;>\,\;\) yields a quantitative property.

Example quantitative properties
\[
Pr(s \models \Diamond G) > \frac{1}{2} \quad \text{or} \quad Pr(s \models \Diamond^n G) \leq \frac{7}{8}
\]

Qualitative properties
Comparing the probability of an event such as \( \square G \), \( \Diamond G \) and \( \square \Diamond G \) with a threshold \( > 0 \) or \( \models = 1 \) yields a qualitative property. Any event \( E \) with \( Pr(E) = 1 \) is called almost surely.

Example qualitative properties
\[
Pr(s \models \Diamond G) > 0 \quad \text{or} \quad Pr(s \models \Diamond^n G) = 1
\]
Verifying qualitative properties

**Remark**
In the following we will concentrate on almost sure events, i.e., events \( E \) with \( \Pr(E) = 1 \). This suffices, as \( \Pr(E) > 0 \) if and only if not \( \Pr(\overline{E}) = 1 \).

---

**Graph notions**

Let \( D = (S, P, \iota_{\text{init}}, AP, L) \) be a (possibly infinite) DTMC.

**Strongly connected component**

- \( T \subseteq S \) is strongly connected if for any \( s, t \in T \), states \( s \) and \( t \in T \) are mutually reachable via edges in \( T \).
- \( T \) is a strongly connected component (SCC) of \( D \) if it is strongly connected and no proper superset of \( T \) is strongly connected.
- SCC \( T \) is a bottom SCC (BSCC) if no state outside \( T \) is reachable from \( T \), i.e., for any state \( s \in T \), \( P(s, T) = \sum_{t \in T} P(s, t) = 1 \).
- Let \( \text{BSCC}(D) \) denote the set of BSCCs of DTMC \( D \).

---

**Evolution of an example DTMC**

Which states have a probability \( > 0 \) when repeating this on the long run?

---

On the long run

The probability mass on the long run is only left in BSCCs.
Measurability

Lemma
For any state \( s \) in (possibly infinite) DTMC \( D \):
\[
\{ \pi \in \text{Paths}(s) \mid \inf(\pi) \in \text{BSCC}(D) \} \text{ is measurable}
\]
where \( \inf(\pi) \) is the set of states that are visited infinitely often along \( \pi \).

Proof:
1. For BSCC \( T \), \( \{ \pi \in \text{Paths}(s) \mid \inf(\pi) = T \} \) is measurable as:
\[
\{ \pi \in \text{Paths}(s) \mid \inf(\pi) = T \} = \bigcap_{t \in T} \square t \cap \lozenge t.
\]
2. As BSCC(\( D \)) is countable, we have:
\[
\{ \pi \in \text{Paths}(s) \mid \inf(\pi) \in \text{BSCC}(D) \} = \bigcup_{T \in \text{BSCC}(D)} \bigcap_{t \in T} \square t \land \lozenge t.
\]

Almost sure reachability

Recall: an absorbing state in a DTMC is a state with a self-loop with probability one.

Almost sure reachability theorem
For finite DTMC with state space \( S \), \( s \in S \) and \( G \subseteq S \) a set of absorbing states:
\[
Pr(s \models \lozenge G) = 1 \iff s \in S \setminus \text{Pre}^*(S \setminus \text{Pre}^*(G)).
\]
Note: \( S \setminus \text{Pre}^*(S \setminus \text{Pre}^*(G)) \) are states that cannot reach states from which \( G \) cannot be reached.

Proof:
Show that both sides of the equivalence are equivalent to \( \text{Post}^*(t) \cap G \neq \emptyset \) for each state \( t \in \text{Post}^*(s) \). Rather straightforward.

Fundamental result

Long-run theorem
For each state \( s \) of a finite Markov chain \( D \):
\[
Pr_s(\{ \pi \in \text{Paths}(s) \mid \inf(\pi) \in \text{BSCC}(M) \} = 1.
\]

Intuition
Almost surely any finite DTMC eventually reaches a BSCC and visits all its states infinitely often.

Computing almost sure reachability properties

Aim:
For finite DTMC \( D \) and \( G \subseteq S \), determine \( \{ s \in S \mid Pr(s \models \lozenge G) = 1 \} \).

Algorithm
1. Make all states in \( G \) absorbing yielding \( D[G] \).
2. Determine \( S \setminus \text{Pre}^*(S \setminus \text{Pre}^*(G)) \) by a graph analysis:
   2.1 do a backward search from \( G \) in \( D[G] \) to determine \( \text{Pre}^*(G) \).
   2.2 followed by a backward search from \( S \setminus \text{Pre}^*(G) \) in \( D[G] \).

This yields a time complexity which is linear in the size of the DTMC \( D \).
Repeated reachability

**Almost sure repeated reachability theorem**

For finite DTMC with state space $S$, $G \subseteq S$, and $s \in S$:

$$\Pr(s \models □♢ G) = 1 \text{ iff for each BSCC } T \subseteq \text{Post}^*(s). T \cap G \neq \emptyset.$$ 

**Proof:**

Immediate consequence of the long-run theorem.

**Almost sure persistence**

**Almost sure persistence theorem**

For finite DTMC with state space $S$, $G \subseteq S$, and $s \in S$:

$$\Pr(s \models ♢□ G) = 1 \text{ if and only if } T \subseteq G \text{ for any BSCC } T \subseteq \text{Post}^*(s)$$

**Example:**

$B = \{s_3, s_4, s_5\}$

**A remark on infinite Markov chains**

**Graph analysis for infinite DTMCs does not suffice!**

Consider the following infinitely countable DTMC, known as random walk:

The value of rational probability $p$ does affect qualitative properties:

$$\Pr(s \models ◇ s_0) = \begin{cases} 1 & \text{if } p \leq \frac{1}{2} \\ < 1 & \text{if } p > \frac{1}{2} \end{cases}$$

$$\Pr(s \models □◇ s_0) = \begin{cases} 1 & \text{if } p \leq \frac{1}{2} \\ 0 & \text{if } p > \frac{1}{2} \end{cases}$$
Quantitative properties

Quantitative repeated reachability theorem

For finite DTMC with state space $S$, $G \subseteq S$, and $s \in S$:

$$Pr(s \models □♦G) = Pr(s \models ♦U)$$

where $U$ is the union of all BSCCs $T$ with $T \cap G \neq \emptyset$.

Quantitative repeated reachability theorem

For finite DTMC with state space $S$, $G \subseteq S$, and $s \in S$:

$$Pr(s \models ♦□G) = Pr(s \models ♦U)$$

where $U$ is the union of all BSCCs $T$ with $T \subseteq G$.

Remark

Thus probabilities for □♦$G$ and ♦□$G$ are reduced to reachability probabilities. These can be computed by solving a linear equation system.

Summary

- Executions of a DTMC are strongly fair with respect to all probabilistic choices.
- A finite DTMC almost surely ends up in a BSCC on the long run.
- Almost sure reachability = double backward search.
- Almost sure □♦$G$ and ♦□$G$ properties can be checked by BSCC analysis and reachability.
- Probabilities for □♦$G$ and ♦□$G$ reduce to reachability probabilities.

Take-home message

For finite DTMCs, qualitative properties do only depend on their state graph and not on the transition probabilities! For infinite DTMCs, this does not hold.

Overview

1. Motivation
2. What are discrete-time Markov chains?
3. Reachability probabilities
4. Qualitative reachability and all that
5. Verifying probabilistic CTL
6. Expressiveness of probabilistic CTL
7. Probabilistic bisimulation
8. Verifying ω-regular properties

Probabilistic Computation Tree Logic

- PCTL is a language for formally specifying properties over DTMCs.
- It is a branching-time temporal logic based on CTL.
- Formula interpretation is Boolean, i.e., a state satisfies a formula or not.
- The main operator is $P_J(\varphi)$
  - where $\varphi$ constrains the set of paths and $J$ is a threshold on the probability.
  - it is the probabilistic counterpart of $\exists$ and $\forall$ path-quantifiers in CTL.
PCTL syntax

[Hansson & Jonsson, 1994]

Probabilistic Computation Tree Logic: Syntax

PCTL consists of state- and path-formulas.

- **PCTL state formulas** over the set $AP$ obey the grammar:

  \[ Φ ::= \text{true} \mid a \mid Φ_1 \land Φ_2 \mid \neg Φ \mid P_J(ϕ) \]

  where $a \in AP$, $ϕ$ is a path formula and $J \subseteq [0,1]$, $J \neq \emptyset$ is a non-empty interval.

- **PCTL path formulae** are formed according to the following grammar:

  \[ ϕ ::= \circ Φ \mid Φ_1 U Φ_2 \mid Φ_1 U^{\leq n} Φ_2 \]

  where $Φ$, $Φ_1$, and $Φ_2$ are state formulae and $n \in \mathbb{N}$.

Abbreviate $P_{[0,0.5]}(ϕ)$ by $P_{\leq 0.5}(ϕ)$ and $P_{[0,1]}(ϕ)$ by $P_{>0}(ϕ)$.

Intuitive semantics

- $s ⟳ P_J(ϕ)$ if:
  - the probability of all paths starting in $s$ fulfilling $ϕ$ lies in $J$.
  - Example: $s ⟳ P_{\geq 0.5}(\Diamond a)$ if
    - the probability to reach an $a$-labeled state from $s$ exceeds $\frac{1}{2}$.
  - Formally:
    - $s ⟳ P_J(ϕ)$ if and only if $Pr_s\{ π ∈ Paths(s) \mid π \models ϕ \} \in J$.

Derived operators

\[ \Diamond Φ = \text{true} U Φ \]

\[ \Diamond^{\leq n} Φ = \text{true} U^{\leq n} Φ \]

\[ P_{\leq p}(□ Φ) = P_{≥ 1-p}(\Diamond \neg Φ) \]

\[ P_{(p,q)}(□^{\leq n} Φ) = P_{[1-q,1-p]}(\Diamond^{\leq n} \neg Φ) \]

Joost-Pieter Katoen
Verifying Continuous-Time Markov Chains

**Correctness of Knuth’s die**

\[
\begin{align*}
P_{\frac{1}{6}}(\Diamond 1) \land P_{\frac{1}{6}}(\Diamond 2) \land P_{\frac{1}{6}}(\Diamond 3) \land P_{\frac{1}{6}}(\Diamond 4) \land P_{\frac{1}{6}}(\Diamond 5) \land P_{\frac{1}{6}}(\Diamond 6)
\end{align*}
\]

**PCTL model checking**

**PCTL model checking problem**

Input: a finite DTMC \( D = (S, P, s_{\text{init}}, AP, L) \), state \( s \in S \), and PCTL state formula \( \Phi \)

Output: yes, if \( s \models \Phi \); no, otherwise.

**Basic algorithm**

In order to check whether \( s \models \Phi \) do:

1. Compute the satisfaction set \( \text{Sat}(\Phi) = \{ s \in S \mid s \models \Phi \} \).
2. This is done recursively by a bottom-up traversal of \( \Phi \)'s parse tree.
   - For each node, i.e., each subformula \( \Psi \) of \( \Phi \), determine \( \text{Sat}(\Psi) \).
   - Determine \( \text{Sat}(\Psi) \) as function of the satisfaction sets of its children: e.g., \( \text{Sat}(\Psi_1 \land \Psi_2) = \text{Sat}(\Psi_1) \cap \text{Sat}(\Psi_2) \) and \( \text{Sat}(\neg \Psi) = S \setminus \text{Sat}(\Psi) \).
3. Check whether state \( s \) belongs to \( \text{Sat}(\Phi) \).

**Core model checking algorithm**

**Probabilistic operator \( P \)**

In order to determine whether \( s \in \text{Sat}(P_J(\varphi)) \), the probability \( Pr(s \models \varphi) \) for the event specified by \( \varphi \) needs to be established. Then

\[
\text{Sat}(P_J(\varphi)) = \{ s \in S \mid Pr(s \models \varphi) \in J \}.
\]

Let us consider the computation of \( Pr(s \models \varphi) \) for all possible \( \varphi \).

**Measurability**

**PCTL measurability**

For any PCTL path formula \( \varphi \) and state \( s \) of DTMC \( D \), the set \( \{ \pi \in \text{Paths}(s) \mid \pi \models \varphi \} \) is measurable.

**Proof (sketch):**

Three cases:

1. \( \Box \Phi \):
   - cylinder sets constructed from paths of length one.
2. \( \Phi \cup^n \Psi \):
   - (finite number of) cylinder sets from paths of length at most \( n \).
3. \( \Phi \cup \Psi \):
   - countable union of paths satisfying \( \Phi \cup^n \Psi \) for all \( n \geq 0 \).
Verifying Continuous-Time Markov Chains

**The next-step operator**

Recall that: $s \models \mathbb{P}_J (\triangledown \Phi)$ if and only if $\Pr(s \models \triangledown \Phi) \in J$.

**Lemma**

\[ \Pr(s \models \triangledown \Phi) = \sum_{s' \in \text{Sat}(\Phi)} P(s, s') \]

**Algorithm**

Considering the above equation for all states simultaneously yields:

\[ (\Pr(s \models \triangledown \Phi))_{s \in S} = \mathbf{P} \cdot \mathbf{b}_\Phi \]

with $\mathbf{b}_\Phi$ the characteristic vector of $\text{Sat}(\Phi)$, i.e., $\mathbf{b}_\Phi(s) = 1$ iff $s \in \text{Sat}(\Phi)$.

Checking the next-step operator reduces to a single matrix-vector multiplication.

---

**Time complexity**

Let $|\Phi|$ be the size of $\Phi$, i.e., the number of logical and temporal operators in $\Phi$.

**Time complexity of PCTL model checking**

For finite DTMC $D$ and PCTL state-formula $\Phi$, the PCTL model-checking problem can be solved in time

\[ O(\text{poly(size}(D)) \cdot n_{\text{max}} \cdot |\Phi|) \]

where $n_{\text{max}} = \max \{ n \mid \Psi_1 \cup \leq^n \Psi_2 \text{ occurs in } \Phi \}$ with and $n_{\text{max}} = 1$ if $\Phi$ does not contain a bounded until-operator.

---

**Example**

Consider DTMC:

\[ \begin{array}{c}
S_0 \xrightarrow{\text{try}} S_1 \\
S_1 \xrightarrow{\text{fail}} S_2 \\
S_2 \xrightarrow{\text{succ}} S_3 \\
S_3 \xrightarrow{\text{try}} S_0
\end{array} \]

and PCTL-formula:

\[ \mathbb{P}_{\geq 0.9} (\triangledown (\neg \text{try} \lor \text{succ})) \]

1. $\text{Sat}(\neg \text{try} \lor \text{succ}) = (S \setminus \text{Sat}(\text{try})) \cup \text{Sat}(\text{succ}) = \{ s_0, s_2, s_3 \}$
2. We know: $(\Pr(s \models \triangledown \Phi))_{s \in S} = \mathbf{P} \cdot \mathbf{b}_\Phi$ where $\Phi = \neg \text{try} \lor \text{succ}$
3. Applying that to this example yields:

\[ (\Pr(s \models \triangledown \Phi))_{s \in S} = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0.01 & 0.01 & 0.98 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix} \cdot \begin{pmatrix}
1 \\
0 \\
0 \\
1
\end{pmatrix} = \begin{pmatrix}
0.99 \\
1 \\
1 \\
1
\end{pmatrix} \]

4. Thus: $\text{Sat}(\mathbb{P}_{\geq 0.9} (\triangledown (\neg \text{try} \lor \text{succ}))) = \{ s_1, s_2, s_3 \}$.

---

**Proof (sketch)**

1. For each node in the parse tree, a model-checking is performed; this yields a linear complexity in $|\Phi|$.
2. The worst-case operator is (unbounded) until.
   2.1 Determining $S_{\leq 0}$ and $S_{\leq 1}$ can be done in linear time.
   2.2 Direct methods to solve linear equation systems are in $\Theta(|S|^3)$.
3. Strictly speaking, $U^{\leq n}$ could be more expensive for large $n$.
   But it remains polynomial, and $n$ is small in practice.
Some practical verification times

- Command-line tool MRMC ran on a Pentium 4, 2.66 GHz, 1 GB RAM laptop.
- PCTL formula $\mathbb{P}_p(\Diamond \text{obs})$ where \text{obs} holds when the sender’s id is detected.

Summary

- PCTL is a variant of CTL with operator $\mathbb{P}_p(\varphi)$.
- Sets of paths fulfilling PCTL path-formula $\varphi$ are measurable.
- PCTL model checking is performed by a recursive descent over $\Phi$.
- The next operator amounts to a single matrix-vector multiplication.
- The bounded-until operator $\mathbb{U}^n$ amounts to $n$ matrix-vector multiplications.
- The until-operator amounts to solving a linear equation system.
- The worst-case time complexity is polynomial in the size of the DTMC and linear in the size of the formula.

Overview

1. Motivation
2. What are discrete-time Markov chains?
3. Reachability probabilities
4. Qualitative reachability and all that
5. Verifying probabilistic CTL
6. Expressiveness of probabilistic CTL
7. Probabilistic bisimulation
8. Verifying $\omega$-regular properties
Qualitative PCTL

State formulae in the qualitative fragment of PCTL (over \( AP \)):

\[
\begin{align*}
\Phi & ::= \text{true} \mid a \mid \Phi_1 \land \Phi_2 \mid \neg \Phi \mid P_{>0}(\varphi) \mid P_{=1}(\varphi)
\end{align*}
\]

where \( a \in AP \), and \( \varphi \) is a path formula formed according to the grammar:

\[
\varphi ::= \Diamond \Phi \mid \Phi_1 \cup \Phi_2.
\]

Examples

\( P_{=1}(\Diamond P_{>0}(\Diamond a)) \) and \( P_{<1}(P_{>0}(\Diamond a) \cup b) \) are qualitative PCTL formulas.

CTL versus qualitative PCTL

(1) \( P_{>0}(\Diamond a) \equiv \exists \Diamond a \) and (2) \( P_{=1}(\Box a) \equiv \forall \Box a \).

Proof:

(1) Consider the first statement.
   
   \[\Rightarrow\] Assume \( s \models P_{>0}(\Diamond a) \). By the PCTL semantics, \( Pr(s \models \Diamond a) > 0 \).
   
   Thus, \( \{ \pi \in Paths(s) \mid \pi \models \Diamond a \} \neq \emptyset \), and hence, \( s \models \exists \Diamond a \).
   
   \[\Leftarrow\] Assume \( s \models \exists \Diamond a \), i.e., there is a finite path \( \pi = s_0 s_1 \ldots s_n \) with \( s_0 = s \) and \( s_n \models a \). It follows that all paths in the cylinder set \( Cyl(\pi) \) fulfill \( \Diamond a \). Thus:
   
   \[ Pr(s \models \Diamond a) \geq Pr(Cyl(s_0 s_1 \ldots s_n)) = P(s_0 s_1 \ldots s_n) > 0. \]
   
   So, \( s \models P_{>0}(\Diamond a) \).

(2) The second statement follows by duality.

Equivalence of PCTL and CTL Formulae

The PCTL formula \( \Phi \) is equivalent to the CTL formula \( \Psi \), denoted \( \Phi \equiv \Psi \), if \( Sat(\Phi) = Sat(\Psi) \) for each DTMC \( D \).

Example

The simplest such cases are path formulae involving the next-step operator:

\[
\begin{align*}
P_{=1}(\Box a) & \equiv \forall \Box a \\
P_{>0}(\Diamond a) & \equiv \exists \Diamond a \\
P_{>0}(\Box a) & \equiv \exists \Box a \\
P_{=1}(\Box a) & \equiv \forall \Box a.
\end{align*}
\]

CTL versus qualitative PCTL

(1) \( P_{>0}(\Diamond a) \equiv \exists \Diamond a \) and (2) \( P_{=1}(\Box a) \equiv \forall \Box a \).

(3) \( P_{>0}(\Box a) \neq \exists \Box a \) and (4) \( P_{=1}(\Diamond a) \neq \forall \Diamond a \).

Example

Consider the second statement (4). Let \( s \) be a state in \( a \) (possibly infinite) DTMC. Then: \( s \models \forall \Diamond a \) implies \( s \models P_{=1}(\Diamond a) \). The reverse direction, however, does not hold. Consider the example DTMC:

\[
\begin{align*}
s & \models P_{=1}(\Diamond a) \text{ as the probability of path } s^\omega = 0. \text{ However, the path } s^\omega \text{ is possible and violates } \Diamond a. \text{ Thus, } s \not\models \forall \Diamond a.
\end{align*}
\]

Statement (3) follows by duality.
Almost-sure-reachability not in CTL

1. There is no CTL formula that is equivalent to $P = 1(◊a)$.
2. There is no CTL formula that is equivalent to $P > 0(□a)$.

Proof:
We provide the proof of 1.; 2. follows by duality: $P = 1(◊a) ≡ ¬P > 0(□¬a)$. By contraposition. Assume $Φ ≡ P = 1(◊a)$. Consider the infinite DTMC $D_p$:

```
1-p  s  p  1-p  p  1-p  1-p  ...
```

The value of $p$ does affect reachability: $Pr(s \models ◊s_0) = \begin{cases} 
1 & \text{if } p \leq \frac{1}{2} \\
< 1 & \text{if } p > \frac{1}{2}
\end{cases}$

Thus, in $D_\frac{1}{2}$ we have $s \models P = 1(◊s_0)$ for all states $s$, while in $D_\frac{3}{4}$, e.g., $s_1 \not\models P = 1(◊s_0)$. Hence: $s_1 \in Sat_{D_\frac{3}{4}}(P = 1(◊s_0))$ but $s_1 \not\in Sat_{D_\frac{1}{2}}(P = 1(◊s_0))$.

For CTL-formula $Φ$ — by assumption $Φ ≡ P = 1(◊a)$ — we have:

$$Sat_{D_\frac{3}{4}}(Φ) = Sat_{D_\frac{1}{2}}(Φ).$$

Hence, state $s_1$ either fulfills the CTL formula $Φ$ in both DTMCs or in none of them. This, however, contradicts $Φ ≡ P = 1(◊a)$.

∀◊ is not expressible in qualitative PCTL

1. There is no qualitative PCTL formula that is equivalent to $∀◊a$.
2. There is no qualitative PCTL formula that is equivalent to $∃□a$.

Fair CTL

Fair paths

In fair CTL, path formulas are interpreted over fair infinite paths, i.e., paths $π$ that satisfy

$$fair = \bigwedge_{s \in S} \bigwedge_{t \in Post(s)} (□◊s \rightarrow □◊t).$$

A path $π$ such that $π \models fair$ is called fair. Let $Paths_{fair}(s)$ be the set of fair paths starting in $s$.

Fair CTL semantics

The fair semantics of CTL is defined by the satisfaction $|=_{fair}$ which is defined as $|=_{fair}$ for the CTL semantics, except that:

$$s |=_{fair}∃ϕ \iff \text{there exists } π ∈ Paths_{fair}(s). π |=_{fair} ϕ$$

$$s |=_{fair}∀ϕ \iff \text{for all } π ∈ Paths_{fair}(s). π |=_{fair} ϕ.$$
Verifying Continuous-Time Markov Chains

Expressiveness of probabilistic CTL

Fairness theorem

Qualitative PCTL versus fair CTL theorem

Let \( s \) be an arbitrary state in a finite DTMC. Then:

\[
\begin{align*}
\mathcal{P} & \quad s \models \mathcal{P} = 1 (\Diamond a) \quad \text{iff} \quad \models_{\text{fair}} \forall \Diamond a \\
\mathcal{P} & \quad s \models \mathcal{P} > 0 (\Box a) \quad \text{iff} \quad \models_{\text{fair}} \exists \Box a \\
\mathcal{P} & \quad s \models \mathcal{P} = 1 (a U b) \quad \text{iff} \quad \models_{\text{fair}} \forall (a U b) \\
\mathcal{P} & \quad s \models \mathcal{P} > 0 (a U b) \quad \text{iff} \quad \models_{\text{fair}} \exists (a U b)
\end{align*}
\]

Comparable expressiveness

Qualitative PCTL and fair CTL are equally expressive.

Almost sure repeated reachability

Almost sure repeated reachability is PCTL-definable

For finite DTMC \( D \), state \( s \in S \) and \( G \subseteq S \):

\[
\begin{align*}
\mathcal{P} & \quad s \models \mathcal{P} = 1 (\Box \mathcal{P} = 1 (\Diamond G)) \quad \text{iff} \quad Pr_{\pi} \{ \pi \in \text{Paths}(s) \mid \pi \models \Box \Diamond G \} = 1.
\end{align*}
\]

We abbreviate \( \mathcal{P} = 1 (\Box \mathcal{P} = 1 (\Diamond G)) \) by \( \mathcal{P} = 1 (\Box \Diamond G) \).

Remark:

For CTL, universal repeated reachability properties can be formalized by the combination of the modalities \( \forall \Box \) and \( \forall \Diamond \):

\[
\begin{align*}
\mathcal{P} & \quad s \models \forall \Box \forall \Diamond G \quad \text{iff} \quad \pi \models \Box \Diamond G \quad \text{for all} \quad \pi \in \text{Paths}(s).
\end{align*}
\]

Almost sure persistence

Almost sure persistence is PCTL-definable

For finite DTMC \( D \), state \( s \in S \) and \( G \subseteq S \):

\[
\begin{align*}
\mathcal{P} & \quad s \models \mathcal{P} = 1 (\Diamond \mathcal{P} = 1 (\Box G)) \quad \text{iff} \quad Pr_{\pi} \{ \pi \in \text{Paths}(s) \mid \pi \models \Diamond \Box G \} = 1.
\end{align*}
\]

We abbreviate \( \mathcal{P} = 1 (\Diamond \mathcal{P} = 1 (\Box G)) \) by \( \mathcal{P} = 1 (\Diamond \Box G) \).

Remark:

Note that \( \forall \Box \Diamond G \) is not CTL-definable. \( \Diamond \Box G \) is a well-known example formula in LTL that cannot be expressed in CTL. But by the above theorem it can be expressed in PCTL.

Repeated reachability probabilities

Repeated reachability probabilities are PCTL-definable

For finite DTMC \( D \), state \( s \in S \), \( G \subseteq S \) and interval \( J \subseteq [0,1] \) we have:

\[
\begin{align*}
\mathcal{P} & \quad s \models \mathcal{P} = 1 (\Diamond \mathcal{P} = 1 (\Box G)) \quad \text{iff} \quad Pr_{\pi} \{ \pi \models \Box G \} \in J.
\end{align*}
\]

Remark:

By the above theorem, \( \mathcal{P} > 0 (\Diamond \Box G) \) is PCTL definable. Note that \( \exists \Box \Diamond G \) is not CTL-definable (but definable in a combination of CTL and LTL, called CTL*).
Persistence probabilities are PCTL-definable

For finite DTMC $D$, state $s \in S$, $G \subseteq S$ and interval $J \subseteq [0,1]$ we have:

$$s \models J(\Diamond P_{=1}(\Box G)) \text{ if and only if } Pr(s \models \Diamond G) \in J.$$ 

Proof:

Left as an exercise. Hint: use the long run theorem.

---

**Overview**

- Motivation
- What are discrete-time Markov chains?
- Reachability probabilities
- Qualitative reachability and all that
- Verifying probabilistic CTL
- Expressiveness of probabilistic CTL
- Probabilistic bisimulation
- Verifying \(\omega\)-regular properties

**Probabilistic bisimulation: intuition**

**Intuition**

- Strong bisimulation is used to compare labeled transition systems.
- Strongly bisimilar states exhibit the same step-wise behaviour.
- Our aim: adapt bisimulation to discrete-time Markov chains.
- This yields a probabilistic variant of strong bisimulation.

- When do two DTMC states exhibit the same step-wise behaviour?
- Key: if their transition probability for each equivalence class coincides.

**Summary**

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

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.
**Probabilistic bisimulation**

[Larsen & Skou, 1989]

Let $D = (S, P, t_{init}, AP, L)$ be a DTMC and $R \subseteq S \times S$ an equivalence. Then: $R$ is a **probabilistic bisimulation** on $S$ if for any $(s, t) \in R$:

1. $L(s) = L(t)$, and
2. $P(s, C) = P(t, C)$ for all equivalence classes $C \in S/R$

where $P(s, C) = \sum_{s' \in C} P(s, s')$.

For states in $R$, the probability of moving by a single transition to some equivalence class is equal.

**Probabilistic bisimilarity**

Let $D$ be a DTMC and $s, t$ states in $D$. Then: $s \sim_P t$, denoted $s \sim_P t$, if there exists a probabilistic bisimulation $R$ with $(s, t) \in R$.

**Example**

![Example Diagram]

**Quotient under $\sim_P$**

**Quotient DTM under $\sim_P$**

For $D = (S, P, t_{init}, AP, L)$ and probabilistic bisimulation $\sim_P \subseteq S \times S$ let $D/\sim_P = (S', P', t'_{init}, AP, L')$, the **quotient** of $D$ under $\sim_P$ where:

- $S' = S/\sim_P = \{ [s]_{\sim_P} | s \in S \}$ with $[s]_{\sim_P} = \{ s' \in S | s \sim_P s' \}$
- $P'([s]_{\sim_P}, [s']_{\sim_P}) = P(s, [s']_{\sim_P})$
- $t'_{init}([s]_{\sim_P}) = \sum_{s' \in [s]_{\sim_P}} t_{init}(s)$
- $L'([s]_{\sim_P}) = L(s)$.

**Remarks**

The transition probability from $[s]_{\sim_P}$ to $[t]_{\sim_P}$ equals $P(s, [t]_{\sim_P})$. This is well-defined as $P(s, C) = P(s', C)$ for all $s \sim_P s'$ and all bisimulation equivalence classes $C$. As opposed to bisimulation on states in transition systems, any probabilistic bisimulation is an equivalence.
Craps

- **Come-out roll:**
  - 7 or 11: win
  - 2, 3, or 12: lose
  - else: roll again

- **Next roll(s):**
  - 7: lose
  - point: win
  - else: roll again

Quotient DTMC of Craps under $\sim_P$

Preservation of PCTL-formulas

**Bisimulation preserves PCTL**

Let $D$ be a DTMC and $s, t$ states in $D$. Then:

$$s \sim_P t \text{ if and only if } s \text{ and } t \text{ are PCTL-equivalent.}$$

**Remarks**

$s \sim_P t$ implies that

1. transient probabilities, reachability probabilities,
2. repeated reachability, persistence probabilities
3. all qualitative PCTL formulas

for $s$ and $t$ are equal.

If for PCTL-formula $\Phi$ we have $s \models \Phi$ but $t \not\models \Phi$, then it follows $s \not\sim_P t$. A single PCTL-formula suffices!

PCTL$^*$ syntax

**Probabilistic Computation Tree Logic: Syntax**

PCTL$^*$ consists of state- and path-formulas.

- **PCTL$^*$ state formulas** over the set $AP$ obey the grammar:

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \land \Phi_2 \mid \neg \Phi \mid \mathbb{P}_J(\varphi)$$

where $a \in AP$, $\varphi$ is a path formula and $J \subseteq [0, 1]$, $J \neq \emptyset$ is a non-empty interval.

- **PCTL$^*$ path formulae** are formed according to the following grammar:

$$\varphi ::= \Phi \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

where $\Phi$ is a state formula and $\varphi$, $\varphi_1$, and $\varphi_2$ are path formulae.
Bounded until in PCTL∗

Bounded until

Bounded until can be defined using the other operators:

\[ \varphi_1 U^\leq n \varphi_2 = \bigvee_{0 \leq i \leq n} \psi_i \]

where \( \psi_0 = \varphi_2 \) and \( \psi_{i+1} = \varphi_1 \land \Box \psi_i \) for \( i \geq 0 \).

Examples in PCTL∗ but not in PCTL

\[ P > \frac{1}{4}(\Box a U \Box b) \] and \[ P = 1(P > \frac{1}{2}(\Box \Diamond a) \lor P \leq \frac{1}{3}(\Diamond \Box b)) \].

Preservation of PCTL/PCTL∗-formulas

Bisimulation preserves PCTL∗

Let \( D \) be a DTMC and \( s, t \) states in \( D \). Then:

\[ s \sim_p t \]

if and only if \( s \) and \( t \) are PCTL∗-equivalent.

Remarks

1. Bisimulation thus preserves not only all PCTL but also all PCTL∗ formulas.
2. By the last two results it follows that PCTL- and PCTL∗-equivalence coincide. Thus any two states that satisfy the same PCTL formulas, satisfy the same PCTL∗ formulas.

PCTL− syntax

Simple Probabilistic Computation Tree Logic: Syntax

PCTL− only consists of state-formulas. These formulas over the set \( AP \) obey the grammar:

\[ \Phi ::= a \mid \Phi_1 \land \Phi_2 \mid P \leq p (\Box \Phi) \]

where \( a \in AP \) and \( p \) is a probability in \([0,1]\).

Remarks

This is a truly simple logic. It does not contain the until-operator. Negation is not present and cannot be expressed. Only upper bounds on probabilities.

The next theorem shows that PCTL−, PCTL∗- and PCTL−-equivalence coincide.

Preservation of PCTL

PCTL/PCTL∗ and Bisimulation Equivalence

Let \( D \) be a DTMC and \( s_1, s_2 \) states in \( D \). Then, the following statements are equivalent:

(a) \( s_1 \sim_p s_2 \).
(b) \( s_1 \) and \( s_2 \) are PCTL∗-equivalent, i.e., fulfill the same PCTL∗ formulas.
(c) \( s_1 \) and \( s_2 \) are PCTL-equivalent, i.e., fulfill the same PCTL formulas.
(d) \( s_1 \) and \( s_2 \) are PCTL−-equivalent, i.e., fulfill the same PCTL− formulas.

Proof:

1. (a) \( \implies \) (b): by structural induction on PCTL∗ formulas.
2. (b) \( \implies \) (c): trivial as PCTL is a sublogic of PCTL∗.
3. (c) \( \implies \) (d): trivial as PCTL− is a sublogic of PCTL.
4. (d) \( \implies \) (a): involved. First finite DTMCs, then for arbitrary DTMCs.
IEEE 802.11 group communication protocol

<table>
<thead>
<tr>
<th>OD</th>
<th>states</th>
<th>transitions</th>
<th>ver. time</th>
<th>blocks</th>
<th>total time</th>
<th>states</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1125</td>
<td>5369</td>
<td>122</td>
<td>71</td>
<td>13</td>
<td>15.9</td>
<td>9.00</td>
</tr>
<tr>
<td>12</td>
<td>37349</td>
<td>236313</td>
<td>7180</td>
<td>1821</td>
<td>642</td>
<td>20.5</td>
<td>11.2</td>
</tr>
<tr>
<td>20</td>
<td>231525</td>
<td>1590329</td>
<td>50133</td>
<td>10627</td>
<td>5431</td>
<td>21.8</td>
<td>9.2</td>
</tr>
<tr>
<td>28</td>
<td>804837</td>
<td>5750873</td>
<td>195086</td>
<td>35961</td>
<td>24716</td>
<td>22.4</td>
<td>7.9</td>
</tr>
<tr>
<td>36</td>
<td>207673</td>
<td>1518733</td>
<td>5103900</td>
<td>91391</td>
<td>77694</td>
<td>22.7</td>
<td>6.6</td>
</tr>
<tr>
<td>40</td>
<td>310145</td>
<td>22871849</td>
<td>7725041</td>
<td>135752</td>
<td>127489</td>
<td>22.9</td>
<td>6.1</td>
</tr>
</tbody>
</table>

Summary

- Bisimilar states have equal transition probabilities to all equivalence classes.
- $\sim_p$ is the coarsest probabilistic bisimulation.
- In a quotient DTMC all states are equivalence classes under $\sim_p$.
- Bisimulation, i.e., $\sim_p$, and PCTL-equivalence coincide.
- PCTL, PCTL$^*$ and PCTL$^-$-equivalence coincide.
- To show $s \not\sim_p t$, show $s \models \Phi$ and $t \not\models \Phi$ for $\Phi \in$ PCTL$^-$.
- Bisimulation may yield up to exponential savings in state space.

Take-home message

Probabilistic bisimulation coincides with a notion from the sixties, named (ordinary) lumpability.

Overview

1. Motivation
2. What are discrete-time Markov chains?
3. Reachability probabilities
4. Qualitative reachability and all that
5. Verifying probabilistic CTL
6. Expressiveness of probabilistic CTL
7. Probabilistic bisimulation
8. Verifying $\omega$-regular properties

Paths and traces

Paths

A path in DTMC $D$ is an infinite sequence of states $s_0 s_1 s_2 \ldots$ with $P(s_i, s_{i+1}) > 0$ for all $i$.

Let $\text{Paths}(D)$ denote the set of paths in $D$, and $\text{Paths}^*(D)$ the set of finite prefixes thereof.

Trace

The trace of path $\pi = s_0 s_1 s_2 \ldots$ is $\text{trace}(\pi) = L(s_0) L(s_1) L(s_2) \ldots$. The trace of finite path $\hat{\pi} = s_0 s_1 \ldots s_n$ is $\text{trace}(\hat{\pi}) = L(s_0) L(s_1) \ldots L(s_n)$.

The set of traces of a set $\Pi$ of paths: $\text{trace}(\Pi) = \{ \text{trace}(\pi) \mid \pi \in \Pi \}$. 
**LT properties**

**Linear-time property**

A **linear-time property** (LT property) over \(AP\) is a subset of \((2^{AP})^\omega\). An LT-property is thus a set of infinite traces over \(2^{AP}\).

**Intuition**

An LT-property gives the admissible behaviours of the DTMC at hand.

**Probability of LT properties**

The **probability** for DTMC \(D\) to exhibit a trace in \(P\) (over \(AP\)) is:

\[
P^D(P) = P^D\{\pi \in \text{Paths}(D) \mid \text{trace}(\pi) \in P\}.
\]

For state \(s\) in \(D\), let \(P(s \models P) = P^s\{\pi \in \text{Paths}(s) \mid \text{trace}(\pi) \in P\} \).

We will later identify a rich set of LT-properties—those that include all LTL formulas—for which \(\{\pi \in \text{Paths}(D) \mid \text{trace}(\pi) \in P\}\) is measurable.

---

**Safety properties**

**Safety property**

LT property \(P_{safe}\) over \(AP\) is a **safety property** if for all \(\sigma \in (2^{AP})^\omega \setminus P_{safe}\) there exists a finite prefix \(\hat{\sigma}\) of \(\sigma\) such that:

\[
P_{safe} \cap \{\sigma' \in (2^{AP})^\omega \mid \hat{\sigma} \text{ is a prefix of } \sigma'\} = \emptyset.
\]

Any such finite word \(\hat{\sigma}\) is called a **bad prefix** for \(P_{safe}\).

**Regular safety property**

A safety property is **regular** if its set of bad prefixes constitutes a regular language (over the alphabet \(2^{AP}\)). Thus, the bad prefixes of a regular safety property can be represented by a finite-state automaton.

---

**Product Markov chain**

**Product Markov chain**

Let \(D = (S, P, t_{\text{init}}, AP, L)\) be a DTMC and \(A = (Q, 2^{AP}, \delta, q_0, F)\) be a DFA. The **product** \(D \otimes A\) is the DTMC:

\[
D \otimes A = (S \times Q, P', t'_{\text{init}}, \{\text{accept}\}, L')
\]

where \(L'((s, q)) = \{\text{accept}\}\) if \(q \in F\) and \(L'((s, q)) = \emptyset\) otherwise, and

\[
t'_{\text{init}}((s, q)) = \begin{cases} t_{\text{init}}(s) & \text{if } q = \delta(q_0, L(s)) \\ 0 & \text{otherwise.} \end{cases}
\]

The transition probabilities in \(D \otimes A\) are given by:

\[
P'((s, q), (s', q')) = \begin{cases} P(s, s') & \text{if } q' = \delta(q, L(s')) \\ 0 & \text{otherwise.} \end{cases}
\]

These probabilities can be obtained by considering a product of DTMC \(D\) with DFA \(A\).
Product Markov chain

Remarks

- For each path \( \pi = s_0 s_1 s_2 \ldots \) in DTMC \( D \) there exists a unique run \( q_0 q_1 q_2 \ldots \) in DFA \( A \) for \( \text{trace}(\pi) = L(s_0) L(s_1) L(s_2) \ldots \), and \( \pi^+ = \langle s_0, q_1 \rangle \langle s_1, q_2 \rangle \langle s_2, q_3 \rangle \ldots \) is a path in \( D \otimes A \).

- The DFA \( A \) does not affect the probabilities, i.e., for each measurable set \( \Pi \) of paths in \( D \) and state \( s \):

\[
P_D^D(\Pi) = P_{D \otimes A}(s, q_0, L(s)) \{ \pi^+ | \pi \in \Pi \}.
\]

- For \( \Pi = \{ \pi \in \text{Paths}(s) | \text{trace}(\pi) \notin P_{\text{safe}} \} \), the set \( \Pi^+ \) is given by:

\[
\Pi^+ = \{ \pi^+ \in \text{Paths}(s, q_0, L(s)) | \pi^+ \models \Diamond \text{accept} \}.
\]

\( \omega \)-regular languages

Infinite repetition of languages

Let \( \Sigma \) be a finite alphabet. For language \( L \subseteq \Sigma^* \), let \( L^\omega \) be the set of words in \( \Sigma^* \cup \Sigma^\omega \) that arise from the infinite concatenation of (arbitrary) words in \( \Sigma \), i.e.,

\[
L^\omega = \{ w_1 w_2 w_3 \ldots | w_i \in L, i \geq 1 \}.
\]

The result is an \( \omega \)-language, i.e., \( L \subseteq \Sigma^\omega \), provided that \( L \subseteq \Sigma^+ \), i.e., \( \varepsilon \notin L \).

\( \omega \)-regular expression

An \( \omega \)-regular expression \( G \) over the \( \Sigma \) has the form: \( G = E_1 F_1^\omega + \ldots + E_n F_n^\omega \) where \( n \geq 1 \) and \( E_1, \ldots, E_n, F_1, \ldots, F_n \) are regular expressions over \( \Sigma \) such that \( \varepsilon \notin L(F_i) \), for all \( 1 \leq i \leq n \).

The semantics of \( G \) is defined by \( L_\omega(G) = L(E_1) L(F_1)^\omega \ldots \cup L(E_n) L(F_n)^\omega \) where \( L(E) \subseteq \Sigma^+ \) denotes the language (of finite words) induced by the regular expression \( E \).

Example

Examples for \( \omega \)-regular expressions over the alphabet \( \Sigma = \{ A, B, C \} \) are:

\((A + B)^* A(AAB + C)^\omega \) or \( A(B + C)^* A^\omega + B(A + C)^\omega \).
**ω-regular properties**

<table>
<thead>
<tr>
<th>ω-regular property</th>
</tr>
</thead>
<tbody>
<tr>
<td>LT property $P$ over $AP$ is called <strong>ω-regular</strong> if $P = L_\omega(G)$ for some $\omega$-regular expression $G$ over the alphabet $2^{AP}$.</td>
</tr>
</tbody>
</table>

**Example**

Let $AP = \{ a, b \}$. Then some $ω$-regular properties over $AP$ are:

- always $a$, i.e., $(\{ a \} + \{ a, b \})^ω$.
- eventually $a$, i.e., $(\emptyset + \{ b \})^ω.(\{ a \} + \{ a, b \} ).(2^{AP})^ω$.
- infinitely often $a$, i.e., $((\emptyset + \{ b \})^ω.(\{ a \} + \{ a, b \} ))^ω$.
- from some moment on, always $a$, i.e., $(2^{AP})^ω.(\{ a \} + \{ a, b \} )^ω$.

---

**Deterministic Rabin automata**

<table>
<thead>
<tr>
<th>Deterministic Rabin automaton</th>
</tr>
</thead>
<tbody>
<tr>
<td>A <strong>deterministic Rabin automaton</strong> $(DRA) A = (Q, \Sigma, \delta, q_0, F)$ with</td>
</tr>
<tr>
<td>- $Q, q_0 \in Q_0$, $\Sigma$ is an alphabet, and $\delta : Q \times \Sigma \rightarrow Q$ as before</td>
</tr>
<tr>
<td>- $F = { (L_i, K_i) \mid 0 &lt; i \leq k }$ with $L_i, K_i \subseteq Q$, is a set of <strong>accept pairs</strong></td>
</tr>
</tbody>
</table>

A **run** for $σ = A_0 A_1 A_2 \ldots \in \Sigma^ω$ denotes an infinite sequence $q_0 q_1 q_2 \ldots$ of states in $A$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A} q_{i+1}$ for $i \geq 0$.

Run $q_0 q_1 q_2 \ldots$ is **accepting** if for some pair $(L_i, K_i)$, the states in $L_i$ are visited **finitely** often and the states in $K_i$ **infinitely** often. That is, an accepting run should satisfy

$$\bigvee_{0 < i \leq k} (\Diamond \Box L_i \land \Box \Diamond K_i).$$

---

**Verifying DRA properties**

<table>
<thead>
<tr>
<th>Product of a Markov chain and a DRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>The product of DTMC $D$ and DRA $A$ is defined as the product of a Markov chain and a DFA, except that the labeling is defined differently.</td>
</tr>
</tbody>
</table>

Let the acceptance condition of $A$ is $F = \{ (L_1, K_1), \ldots, (L_k, K_k) \}$. Then the sets $L_i, K_i$ serve as atomic propositions in $D \otimes A$. The labeling function $L'$ in $D \otimes A$ is the obvious one: if $H \in \{ L_1, \ldots, L_k, K_1, \ldots, K_k \}$, then $H \in L'((s, q))$ if and only if $q \in H$.

<table>
<thead>
<tr>
<th>Accepting BSCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A BSCC $T$ in $T \otimes A$ is <strong>accepting</strong> if and only if there exists some index $i \in { 1, \ldots, k }$ such that:</td>
</tr>
</tbody>
</table>

$$T \cap (S \times L_i) = \emptyset \quad \text{and} \quad T \cap (S \times K_i) \neq \emptyset.$$

Thus, once such an accepting BSCC $T$ is reached in $D \otimes A$, the acceptance criterion for the DRA $A$ is fulfilled almost surely.
Verifying DRA objectives

**Verifying DRA objectives theorem**

Let \( D \) be a finite DTMC, \( s \) a state in \( D \), \( A \) a DRA, and let \( U \) be the union of all accepting BSCCs in \( D \otimes A \). Then:

\[
P^D(s \models A) = P^{D \otimes A}(s, q_s) \models \bigtriangledown U \text{ where } q_s = \delta(q_0, L(s)).
\]

Thus: \( P^D(A) = \sum_{s \in S_{\text{init}}} P^{D \otimes A}(s, \delta(q_0, L(s))) \models \bigtriangledown U \). The computation of probabilities for satisfying \( \omega \)-regular properties boils down to computing the reachability probabilities for certain BSCCs in \( D \otimes A \). Again, a graph analysis and solving systems of linear equations suffice. The time complexity is polynomial in the size of \( D \) and \( A \).

---

Linear temporal logic

**Linear Temporal Logic: Syntax**

[LTL formulas](#) over the set \( AP \) obey the grammar:

\[
\varphi ::= a \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \bigtriangledown \varphi \mid \varphi_1 \mathbf{U} \varphi_2
\]

where \( a \in AP \) and \( \varphi_1 \) and \( \varphi_2 \) are LTL formulas.

---

Measurability

**Measurability theorem for \( \omega \)-regular properties** [Vardi 1985]

For any DTMC \( D \) and \( \omega \)-regular LT property \( P \), the set

\[
\{ \pi \in \text{Paths}(D) \mid \text{trace}(\pi) \in P \}
\]

is measurable.

**Proof (sketch)**

Represent \( P \) by a DRA \( A \) with accept sets \( \{L_1, K_1\}, \ldots, \{L_k, K_k\} \). Let \( \varphi_i = \bigtriangledown \square \neg L_i \land \bigtriangledown \square K_i \) and \( \Pi_i \) the set of paths satisfying \( \varphi_i \). Then \( \Pi = \Pi_1 \cup \ldots \cup \Pi_k \). In addition, \( \Pi_i = \Pi_i^{\bigtriangledown \square} \cap \Pi_i^{\square \bigtriangledown} \) where \( \Pi_i^{\bigtriangledown \square} \) is the set of paths \( \pi \) in \( D \) such that \( \pi^+ \models \bigtriangledown \square \neg L_i \), and \( \Pi_i^{\square \bigtriangledown} \) is the set of paths \( \pi \) in \( D \) such that \( \pi^+ \models \square \bigtriangledown K_i \). It remains to show that \( \Pi_i^{\bigtriangledown \square} \) and \( \Pi_i^{\square \bigtriangledown} \) are measurable. This goes along the same lines as proving that \( \bigtriangledown \square G \) and \( \square \bigtriangledown G \) are measurable.

---

LTL semantics

**LTL semantics**

The LT-property induced by LTL formula \( \varphi \) over \( AP \) is:

\[
\text{Words}(\varphi) = \left\{ \sigma \in (2^{AP})^\omega \mid \sigma \models \varphi \right\}, \text{ where } \models \text{ is the smallest relation s.t.:
}

- \( \sigma \models \text{true} \)
- \( \sigma \models a \) iff \( a \in A_0 \) (i.e., \( A_0 \models a \))
- \( \sigma \models \varphi_1 \land \varphi_2 \) iff \( \sigma \models \varphi_1 \) and \( \sigma \models \varphi_2 \)
- \( \sigma \models \neg \varphi \) iff \( \sigma \not\models \varphi \)
- \( \sigma \models \bigtriangledown \varphi \) iff \( \sigma^1 = A_1 A_2 A_3 \ldots \models \varphi \)
- \( \sigma \models \varphi_1 \mathbf{U} \varphi_2 \) iff \( \exists j \geq 0. \sigma^j \models \varphi_2 \) and \( \sigma^{j+i} \models \varphi_1 \), \( 0 \leq i < j \)

for \( \sigma = A_0 A_1 A_2 \ldots \) we have \( \sigma^i = A_i A_{i+1} A_{i+2} \ldots \) is the suffix of \( \sigma \) from index \( i \) on.
Some facts about LTL

**LTL is \( \omega \)-regular**
For any LTL formula \( \varphi \), the set \( \text{Words}(\varphi) \) is an \( \omega \)-regular language.

**LTL are DRA-definable**
For any LTL formula \( \varphi \), there exists a DRA \( A \) such that \( L_\omega = \text{Words}(\varphi) \) where the number of states in \( A \) lies in \( 2^{2^{\vert \varphi \vert}} \).

### Verifying a DTMC against LTL formulas

**Complexity of LTL model checking** [Vardi 1985]
The qualitative model-checking problem for finite DTMCs against LTL formula \( \varphi \) is PSPACE-complete, i.e., verifying whether \( \Pr(s \models \varphi) > 0 \) or \( \Pr(s \models \varphi) = 1 \) is PSPACE-complete.

Recall that the LTL model-checking problem for finite transition systems is also PSPACE-complete.

### Summary

**Summary**
- Verifying a DTMC \( D \) against a DFA \( A \), i.e., determining \( \Pr(D \models A) \), amounts to computing reachability probabilities of accept states in \( D \otimes A \).
- For DBA objectives, the probability of infinitely often visiting an accept state in \( D \otimes A \).
- DBA are strictly less powerful than \( \omega \)-regular languages.
- Deterministic Rabin automata are as expressive as \( \omega \)-regular languages.
- Verifying DTMC \( D \) against DRA \( A \) amounts to computing reachability probabilities of accepting BSCCs in \( D \otimes A \).

**Take-home message**
Model checking a DTMC against various automata models reduces to computing reachability probabilities in a product.