Discrete-Time and Continuous-Time Markov Chains

Christel Baier
Technische Universität Dresden

Probabilistic Model Checking for Discrete-Time Markovian Models

Christel Baier
Technische Universität Dresden

Probability elsewhere

Probability elsewhere

- randomized algorithms [Rabin 1960]
  symmetry breaking, finderprint techniques,
  random choice of waiting times or IP addresses, ...
- stochastic control theory [Bellman 1957]
  operations research
- performance modeling [Markov, Erlang, Kolm., ∼ 1900]
- biological systems
- resilient systems

  ⋮

Probability elsewhere

- randomized algorithms [Rabin 1960]
  symmetry breaking, finderprint techniques,
  random choice of waiting times or IP addresses, ...
- stochastic control theory [Bellman 1957]
  operations research
- performance modeling [Markov, Erlang, Kolm., ~ 1900]
- biological systems
- resilient systems

discrete or continuous-time Markovian models

memoryless property: future system behavior depends only on the current state, but not on the past

Probabilistic models

Probabilistic models

	purely probabilistic	probabilistic and nondeterministic
discrete time		
continuous time		

Probabilistic models

	purely probabilistic	probabilistic and nondeterministic
discrete time	discrete-time Markov chain (DTMC)	Markov decision process (MDP)
continuous time		

Probabilistic models

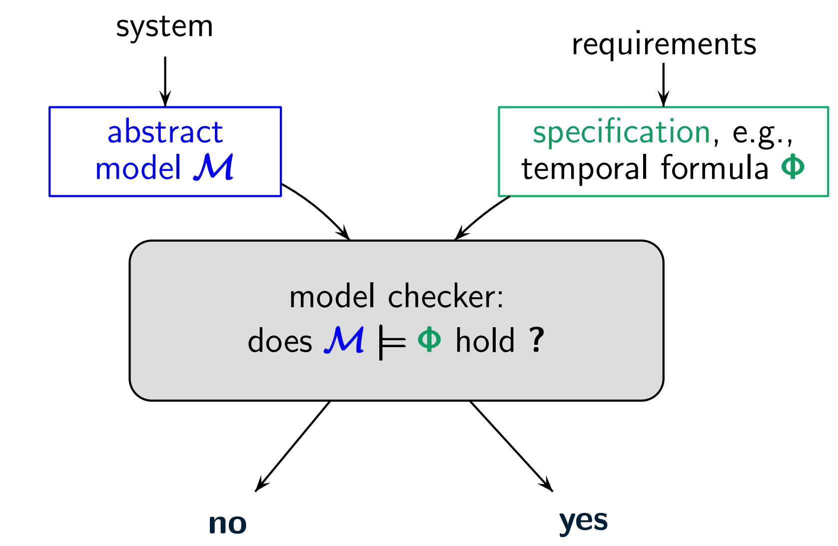
	purely probabilistic	probabilistic and nondeterministic
discrete time	discrete-time Markov chain (DTMC)	Markov decision process (MDP)
continuous time	continuous-time Markov chain (CTMC)	various models

↑ tutorial by *Marta Kwiatkowska*

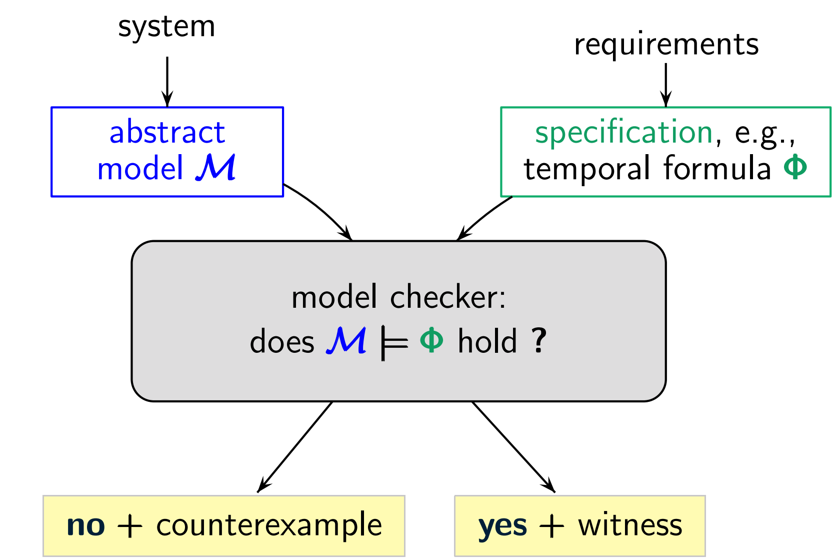
Probabilistic models

	purely probabilistic	probabilistic and nondeterministic
discrete time	discrete-time Markov chain (DTMC)	Markov decision process (MDP)
continuous time	continuous-time Markov chain (CTMC)	continuous-time MDP interactive Markov chains probabilistic timed automata stochastic automata ⋮

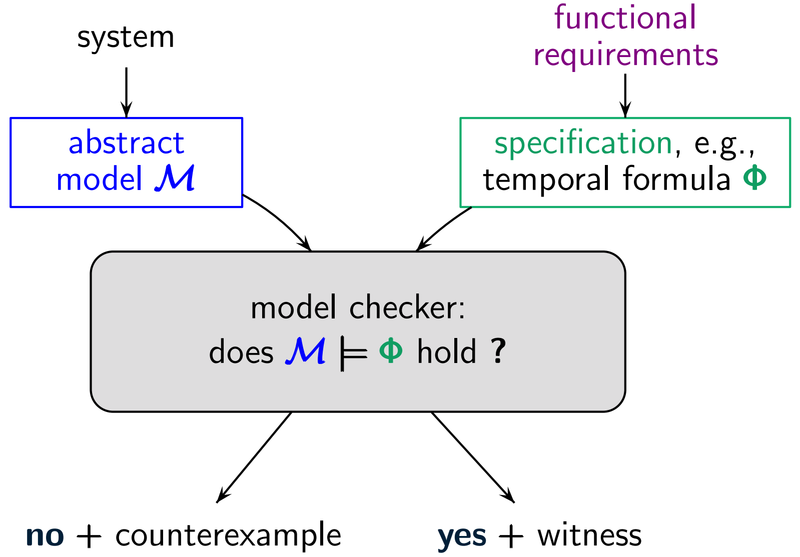
Model checking



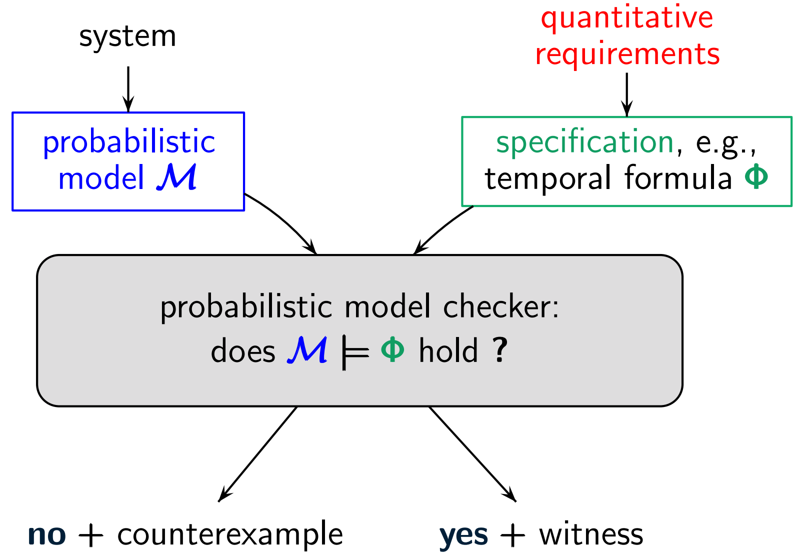
Model checking



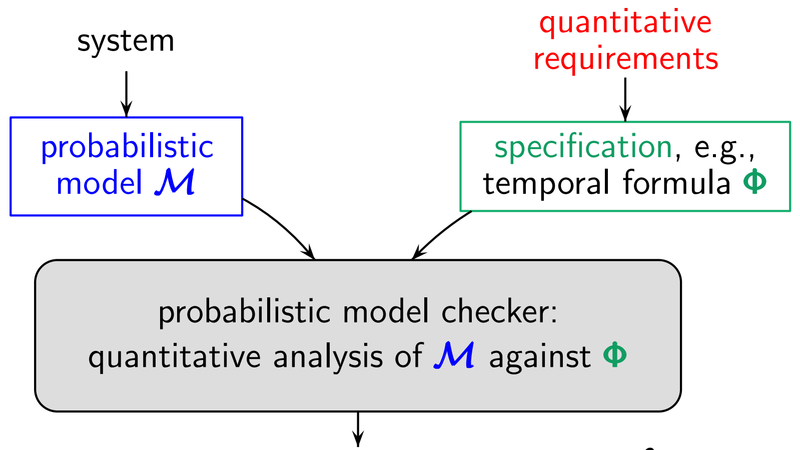
Model checking



Probabilistic model checking

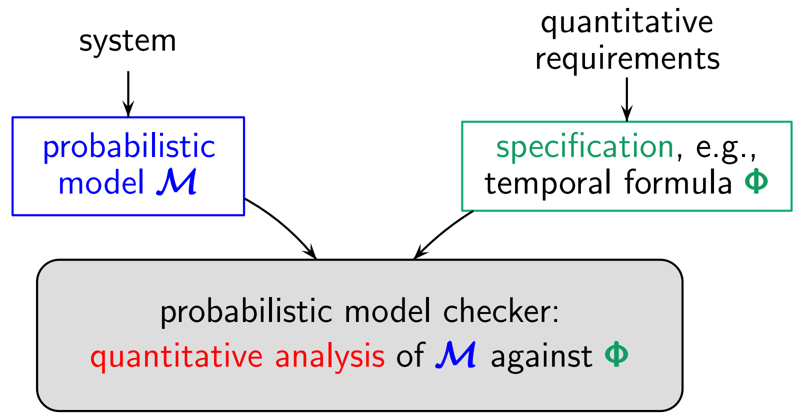


Probabilistic model checking



probability for "bad behaviors" is $< \mathbf{10^{-6}}$
probability for "good behaviors" is **1**
expected costs for ....

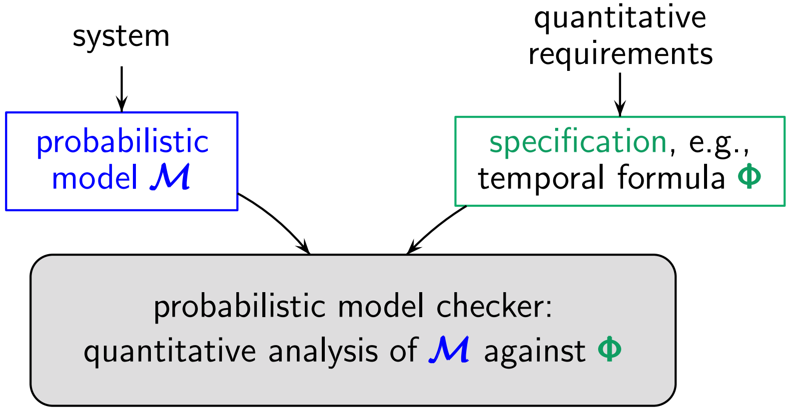
Probabilistic model checking



quantitative analysis relies on a combination of

- model checking techniques
- known concepts for probabilistic models

Probabilistic model checking



logical approach $\leadsto$ unambiguous specifications

model checking $\leadsto$ automatic reasoning

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions ⟵
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness

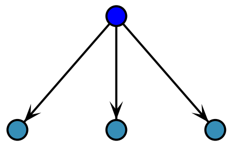
Markov chains

... transition systems with probabilistic distributions for the successor states

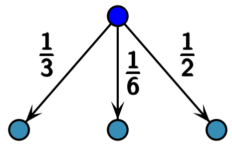
Markov chains

DTMC-05

... transition systems with probabilistic distributions for the successor states



transition system: nondeterministic branching



Markov chain: probabilistic branching

(discrete-time)

Discrete-time Markov chain (DTMC)

DTMC-07

$\mathcal{M} = (S, P, \ldots)$

Discrete-time Markov chain (DTMC)

DTMC-07

$\mathcal{M} = (S, P, \ldots)$

- state space S

Discrete-time Markov chain (DTMC)

DTMC-07

$\mathcal{M} = (S, P, \ldots)$

- state space S ⟵ here: finite

Discrete-time Markov chain (DTMC)

$\mathcal{M} = (S, P, \ldots)$

- state space S ⟵ here: finite
- transition probability function $P : S \times S \to [0,1]$

  s.t. $\sum_{s' \in S} P(s, s') = 1$

Discrete-time Markov chain (DTMC)

$\mathcal{M} = (S, P, \ldots)$

- state space S ⟵ here: finite
- transition probability function $P : S \times S \to [0,1]$

  s.t. $\sum_{s' \in S} P(s, s') = 1$

discrete-time or time-abstract:
probability for the step $s \longrightarrow s'$

Discrete-time Markov chain (DTMC)

$\mathcal{M} = (S, P, AP, L, \ldots)$

- state space S
- transition probability function $P : S \times S \to [0,1]$

  s.t. $\sum_{s' \in S} P(s, s') = 1$
- AP set of atomic propositions
- labeling function $L : S \to 2^{AP}$

Discrete-time Markov chain (DTMC)

$\mathcal{M} = (S, P, AP, L, \ldots)$ ⟵

+ initial distribution
+ reward/cost function
+ ...

- state space S
- transition probability function $P : S \times S \to [0,1]$

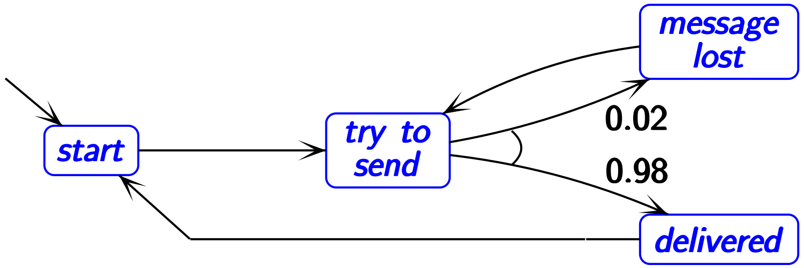
  s.t. $\sum_{s' \in S} P(s, s') = 1$
- AP set of atomic propositions
- labeling function $L : S \to 2^{AP}$
- $\mu : S \to [0,1]$ initial distribution
- $rew : S \to \mathbb{N}$ where $rew(s)$ is the reward earned per visit of state s

Example: DTMC for communication protocol

$\mathcal{M} = (S, P, AP, L, \ldots)$

- state space S
- transition probability function $P : S \times S \to [0,1]$

s.t. $\sum_{s' \in S} P(s, s') = 1$

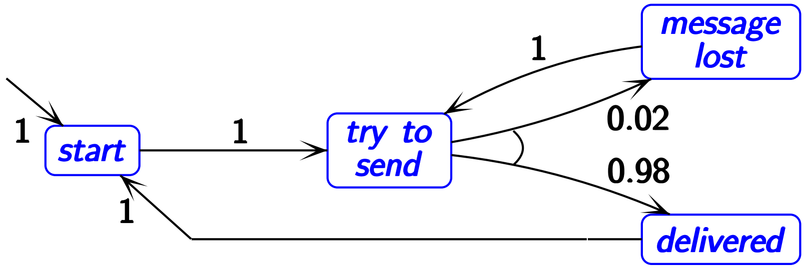


Example: DTMC for communication protocol

$\mathcal{M} = (S, P, AP, L, \ldots)$

- state space S
- transition probability function $P : S \times S \to [0,1]$

s.t. $\sum_{s' \in S} P(s, s') = 1$



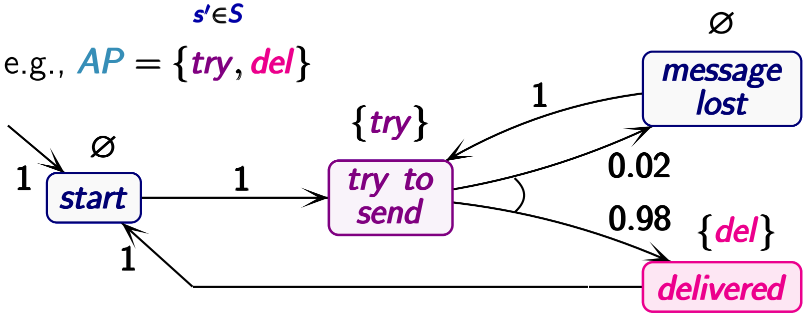
Example: DTMC for communication protocol

$\mathcal{M} = (S, P, AP, L, \ldots)$

- state space S
- transition probability function $P : S \times S \to [0,1]$

s.t. $\sum_{s' \in S} P(s, s') = 1$

e.g., $AP = \{try, del\}$



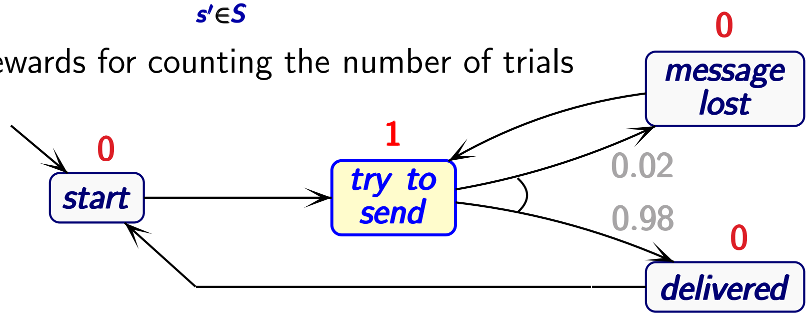
Example: DTMC for communication protocol

$\mathcal{M} = (S, P, AP, L, \ldots)$

- state space S
- transition probability function $P : S \times S \to [0,1]$

s.t. $\sum_{s' \in S} P(s, s') = 1$

rewards for counting the number of trials



Probability measure of a Markov chain

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

↑ initial distribution $\mu : S \to [0,1]$

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

↑ initial distribution $\mu : S \to [0,1]$

probability measure for (measurable) sets of paths:

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

↑ initial distribution $\mu : S \to [0,1]$

probability measure for (measurable) sets of paths:

consider the σ-algebra generated by cylinder sets

$\Delta(s_0\, s_1 \ldots s_n)$ = set of infinite paths $s_0\, s_1 \ldots s_n\, s_{n+1}\, s_{n+2}\, s_{n+3} \ldots$

↑ finite path

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

initial distribution $\mu : S \to [0,1]$

probability measure for (measurable) sets of paths:

consider the σ-algebra generated by cylinder sets

$\Delta(s_0\, s_1 \ldots s_n) \;=\;$ set of infinite paths ...

σ-algebra on universe $\mathcal{U}$: set $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

initial distribution $\mu : S \to [0,1]$

probability measure for (measurable) sets of paths:

consider the σ-algebra generated by cylinder sets

$\Delta(s_0\, s_1 \ldots s_n) \;=\;$ set of infinite paths ...

here: $\mathcal{U} \;=\;$ set of infinite paths $\subseteq S^\omega$

$\mathcal{V} \;=\;$ smallest subset of $2^{\mathcal{U}}$ that contains all cylinder sets and is closed under complement and countable unions

Probability measure of a Markov chain

$\mathcal{M} = (S, P, AP, L, \mu)$ Markov chain

initial distribution $\mu : S \to [0,1]$

probability measure for (measurable) sets of paths:

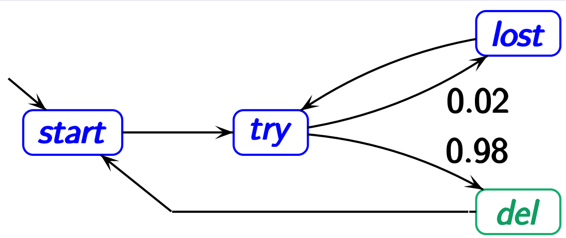
consider the σ-algebra generated by cylinder sets

$\Delta(s_0\, s_1 \ldots s_n) \;=\;$ set of infinite paths $s_0\, s_1 \ldots s_n\, s_{n+1}\, s_{n+2}\, s_{n+3} \ldots$

probability measure is given by:

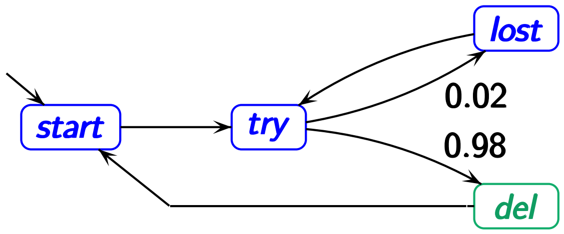
$$\Pr^{\mathcal{M}}\big(\,\Delta(s_0\, s_1 \ldots s_n)\,\big) \;=\; \mu(s_0) \cdot \prod_{1 \leqslant i \leqslant n} P(s_{i-1}, s_i)$$

Example: Markov chain



probability for delivering the message within **5** steps:

Example: Markov chain

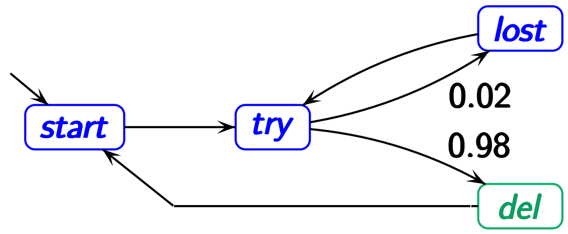


probability for delivering the message within **5** steps:

$$= \Pr^{\mathcal{M}}(\textit{start try del}) + \Pr^{\mathcal{M}}(\textit{start try lost try del})$$

notation: $\Pr^{\mathcal{M}}(s_0\, s_1 \ldots s_n) = \Pr^{\mathcal{M}}(\Delta(s_0\, s_1 \ldots s_n))$

Example: Markov chain



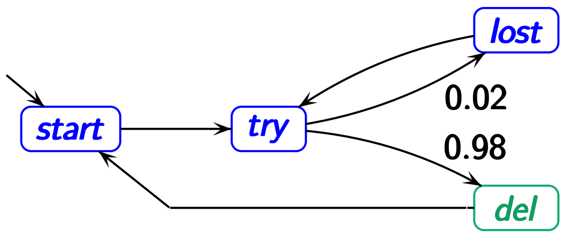
probability for delivering the message within **5** steps:

$$= \Pr^{\mathcal{M}}(\textit{start try del}) + \Pr^{\mathcal{M}}(\textit{start try lost try del})$$

$$= 0.98 + 0.02 \cdot 0.98 = 0.9996$$

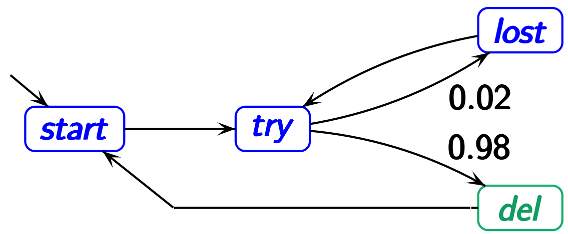
notation: $\Pr^{\mathcal{M}}(s_0\, s_1 \ldots s_n) = \Pr^{\mathcal{M}}(\Delta(s_0\, s_1 \ldots s_n))$

Example: Markov chain



probability for eventually delivering the message:

Example: Markov chain

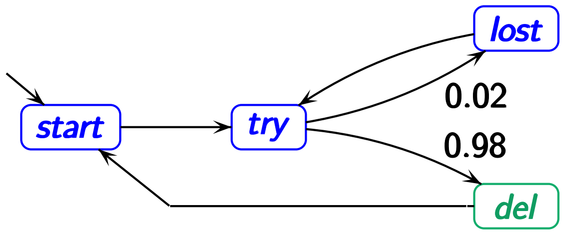


probability for eventually delivering the message:

$$= \sum_{n=0}^{\infty} \Pr^{\mathcal{M}}(\textit{start try } (\textit{lost try})^n \textit{ del})$$

Example: Markov chain

DTMC-20



probability for eventually delivering the message:

$$= \sum_{n=0}^{\infty} \Pr^{\mathcal{M}}(\mathit{start}\ \mathit{try}\ (\mathit{lost}\ \mathit{try})^n\ \mathit{del})$$

$$= \sum_{n=0}^{\infty} 0.02^n \cdot 0.98 \ = \ 1$$

Measurability of classical properties

DTMC-25

Measurability of classical properties

DTMC-25

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

Measurability of classical properties

DTMC-25

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i\in\mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

For DTMCs: $\mathcal{U}$ = set of infinite paths

$\mathcal{V}$ = σ-algebra generated by the cylinder sets

$$\Delta(s_0\, s_1 \ldots s_n) = \left\{ \begin{array}{l} \text{set of infinite paths } \pi \text{ of the form} \\ s_0\, s_1 \ldots s_n\, s_{n+1}\, s_{n+2}\, s_{n+3} \ldots \end{array} \right.$$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i\in\mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

step-bounded reachability: "visit G within n steps"

$$\Diamond^{\leqslant n} G = \bigcup_{0\leqslant i\leqslant n} \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

where $s_i \in G$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i\in\mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

step-bounded reachability: "visit G within n steps"

$$\Diamond^{\leqslant n} G = \bigcup_{0\leqslant i\leqslant n} \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

where $s_i \in G$ and $s_0, \ldots, s_{i-1} \notin G$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i\in\mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

step-bounded reachability: "visit G within n steps"

$$\Diamond^{\leqslant n} G = \bigcup_{0\leqslant i\leqslant n} \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

$$\Pr^{\mathcal{M}}(\Diamond^{\leqslant n} G) = \sum_{0\leqslant i\leqslant n} \sum_{s_0,\ldots,s_i} \Pr^{\mathcal{M}}(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

unbounded reachability: "visit G eventually"

$$\Diamond G \;=\; \bigcup_{i \in \mathbb{N}} \; \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

where $s_i \in G$ and $s_0, \ldots, s_{i-1} \notin G$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

unbounded reachability: "visit G eventually"

$$\Diamond G \;=\; \bigcup_{i \in \mathbb{N}} \; \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

$$\Pr^{\mathcal{M}}(\Diamond G) \;=\; \sum_{i \in \mathbb{N}} \; \sum_{s_0,\ldots,s_i} \Pr^{\mathcal{M}}(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

repeated reachability: "visit G infinitely often"

$$\Box\Diamond G \;=\; \bigcap_{n \in \mathbb{N}} \; \bigcup_{i \geqslant n} \; \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

where $s_i \in G$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U}, \mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

repeated reachability: "visit G infinitely often"

$$\Box\Diamond G \;=\; \bigcap_{n \in \mathbb{N}} \; \bigcup_{i \geqslant n} \; \bigcup_{s_0,\ldots,s_i} \Delta(s_0\, s_1 \ldots s_{i-1}\, s_i)$$

where $s_i \in G$, but possibly $s_j \in G$ for some $j < i$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

persistence: "from some moment on always G"

$$\Diamond\Box G \;=\; Paths^{\mathcal{M}} \setminus \Box\Diamond\neg G$$

Measurability of classical properties

A σ-algebra is a pair $(\mathcal{U},\mathcal{V})$ where $\mathcal{U}$ is a set and $\mathcal{V} \subseteq 2^{\mathcal{U}}$ s.t.

1. $\mathcal{U} \in \mathcal{V}$
2. if $T \in \mathcal{V}$ then $\mathcal{U} \setminus T \in \mathcal{V}$
3. if $T_i \in \mathcal{V}$ for $i \in \mathbb{N}$ then $\bigcup_{i \in \mathbb{N}} T_i \in \mathcal{V}$

The elements of $\mathcal{V}$ are called measurable events.

persistence: "from some moment on always G"

$$\Diamond\Box G \;=\; Paths^{\mathcal{M}} \setminus \Box\Diamond\neg G$$

$$\Pr^{\mathcal{M}}(\Diamond\Box G) \;=\; 1 - \Pr^{\mathcal{M}}(\Box\Diamond\neg G)$$

Stochastic process of a Markov chain

Stochastic process of a Markov chain

stochastic process (general definition):

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

Stochastic process of a Markov chain

stochastic process (general definition):

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain, e.g., $\mathbb{N}$ or $\mathbb{R}_{\geqslant 0}$
- S is a set (with fixed σ-algebra)
- $\mathcal{U}$ is a sample space (with fixed σ-algebra and probability measure)

Stochastic process of a Markov chain

stochastic process of DTMC $\mathcal{M} = (S, P, \ldots)$

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain $\longleftarrow$ $Time = \mathbb{N}$
- S is a set $\longleftarrow$ state space
- $\mathcal{U}$ is a sample space $\longleftarrow$ set of infinite paths

Stochastic process of a Markov chain

stochastic process of DTMC $\mathcal{M} = (S, P, \ldots)$

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain $\longleftarrow$ $Time = \mathbb{N}$
- S is a set $\longleftarrow$ state space
- $\mathcal{U}$ is a sample space $\longleftarrow$ set of infinite paths

---

If $t \in \mathbb{N}$ and $\pi = s_0\, s_1\, s_2 \ldots s_t \ldots$ then $X_t(\pi) = s_t$.

Stochastic process of a Markov chain

stochastic process of DTMC $\mathcal{M} = (S, P, \ldots)$

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain $\longleftarrow$ $Time = \mathbb{N}$
- S is a set $\longleftarrow$ state space
- $\mathcal{U}$ is a sample space $\longleftarrow$ set of infinite paths

---

If $t \in \mathbb{N}$ and $\pi = s_0\, s_1 \ldots s_{t-2}\, u\, s_t \ldots$ then $X_t(\pi) = s_t$.

Markov property:

$$\Pr^{\mathcal{M}}\big(X_t = s \,\big|\, X_{t-1} = u \big) \; =$$
$$\Pr^{\mathcal{M}}\big(X_t = s \,\big|\, X_{t-1} = u,\, X_{t-2} = s_{t-2},\, \ldots,\, X_0 = s_0 \big)$$

Stochastic process of a Markov chain

stochastic process of DTMC $\mathcal{M} = (S, P, \ldots)$

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain $\longleftarrow$ $Time = \mathbb{N}$
- S is a set $\longleftarrow$ state space
- $\mathcal{U}$ is a sample space $\longleftarrow$ set of infinite paths

---

If $t \in \mathbb{N}$ and $\pi = s_0\, s_1\, \ldots s_{t-2}\, u\, s_t \ldots$ then $X_t(\pi) = s_t$.

Markov property:

$$\Pr^{\mathcal{M}}\big(X_t = s \,\big|\, X_{t-1} = u\big) \;=\; P(u,s) \;=$$
$$\Pr^{\mathcal{M}}\big(X_t = s \,\big|\, X_{t-1} = u,\, X_{t-2} = s_{t-2},\, \ldots,\, X_0 = s_0\big)$$

Stochastic process of a Markov chain

stochastic process of DTMC $\mathcal{M} = (S, P, \ldots)$

family $(X_t)_{t \in Time}$ of random variables $X_t : \mathcal{U} \to S$

- $Time$ is a time domain $\longleftarrow$ $Time = \mathbb{N}$
- S is a set $\longleftarrow$ state space
- $\mathcal{U}$ is a sample space $\longleftarrow$ set of infinite paths

---

If $t \in \mathbb{N}$ and $\pi = s_0\, s_1\, \ldots s_{t-2}\, u\, s_t \ldots$ then $X_t(\pi) = s_t$.

Markov property:

$$\Pr^{\mathcal{M}}\big(X_t = s \,\big|\, X_{t-1} = u\big) \;=\; P(s,u) \;=$$
$$\Pr^{\mathcal{M}}\big(X_1 = s \,\big|\, X_0 = u\big) \qquad \text{time-homogeneous}$$

Transient and long-run distribution

Transient and long-run distribution

transient: ... refers to a fixed time point t

long-run: ... when time tends to infinity

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \Pr^{\mathcal{M}}(X_t = s) \end{aligned}$$

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \mu \cdot P^t \cdot id_s \end{aligned}$$

↑ initial distribution (row vector)

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \mu \cdot P^t \cdot id_s \end{aligned}$$

↑ t-th power of transition probability matrix $P^t = P^{t-1} \cdot P$

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \mu \cdot P^t \cdot id_s \end{aligned}$$

↑ unit vector $(0 \ldots 0, 1, 0, \ldots 0)^T$ representing Dirac distribution for state s

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr{}^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \mu \cdot P^t \cdot id_s \;=\; \mu_{t-1} \cdot P \cdot id_s \end{aligned}$$

unit vector $(0 \ldots 0, 1, 0, \ldots 0)^T$
representing Dirac distribution
for state s

Transient distribution

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC, $t \in \mathbb{N}$ and $s \in S$.

transient state probability:

$$\begin{aligned} \mu_t(s) &= \Pr{}^{\mathcal{M}}\{s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : s_t = s\} \\ &= \mu \cdot P^t \cdot id_s \;=\; \mu_{t-1} \cdot P \cdot id_s \end{aligned}$$

transient state distribution
for time point $t-1$

$\mu_0 = \mu$ initial distribution

$\mu_n = \mu_{n-1} \cdot P$ for $n \geqslant 1$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t \to \infty} \mu_t(s)$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t \to \infty} \mu_t(s)$

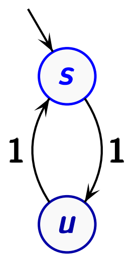
- limit may not exist

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\tilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

- limit may not exist



$\mu_{2t}(s) = 1$ ← even time points

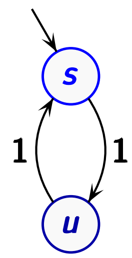
$\mu_{2t+1}(s) = 0$ ← odd time points

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\tilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ



$\mu_{2t}(s) = 1$ ← even time points

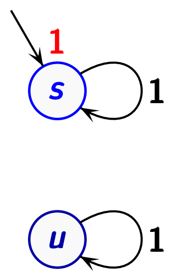
$\mu_{2t+1}(s) = 0$ ← odd time points

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\tilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ



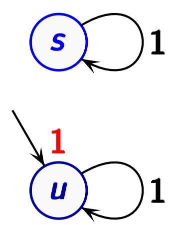
If $\mu(s) = 1$ then: $\tilde{\mu}(s) = 1$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\tilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ



If $\mu(s) = 1$ then: $\tilde{\mu}(s) = 1$

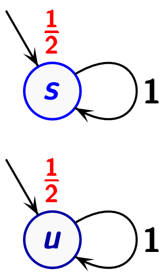
If $\mu(u) = 1$ then: $\tilde{\mu}(s) = 0$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ



If $\mu(s) = 1$ then: $\widetilde{\mu}(s) = 1$

If $\mu(u) = 1$ then: $\widetilde{\mu}(s) = 0$

If $\mu(s) = \mu(u) = \frac{1}{2}$ then:
$\widetilde{\mu}(s) = \frac{1}{2}$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ
- if existing for all states s then $\widetilde{\mu} = \widetilde{\mu} \cdot P$

↑ balance equation

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t\to\infty} \mu_t(s)$

- limit may not exist or depend on the initial distribution μ
- if existing for all states s then $\widetilde{\mu} = \widetilde{\mu} \cdot P$

↑ balance equation

unique solution with the side constraint

$$\sum_{s\in S} \widetilde{\mu}(s) = 1$$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim\limits_{t\to\infty} \mu_t(s)$

long-run fraction of being in state s (Cesàro limit):

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

long-run fraction of being in state s (Cesàro limit):

$$\theta(s) \;=\; \lim_{t\to\infty} \frac{1}{t+1} \cdot \sum_{i=0}^{t} \mu_i(s)$$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

long-run fraction of being in state s (Cesàro limit):

$$\theta(s) \;=\; \lim_{t\to\infty} \frac{1}{t+1} \cdot \sum_{i=0}^{t} \mu_i(s)$$

- Cesàro limit always exists

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

long-run fraction of being in state s (Cesàro limit):

$$\theta(s) \;=\; \lim_{t\to\infty} \frac{1}{t+1} \cdot \sum_{i=0}^{t} \mu_i(s)$$

- Cesàro limit always exists
- if the steady-state probabilities exists: $\widetilde{\mu}(s) = \theta(s)$

Long-run distributions

Let $\mathcal{M} = (S, P, \mu, \ldots)$ be a DTMC.

steady-state probability: $\widetilde{\mu}(s) = \lim_{t\to\infty} \mu_t(s)$

long-run fraction of being in state s (Cesàro limit):

$$\theta(s) \;=\; \lim_{t\to\infty} \frac{1}{t+1} \cdot \sum_{i=0}^{t} \mu_i(s)$$

- Cesàro limit always exists
- if the steady-state probabilities exists: $\widetilde{\mu}(s) = \theta(s)$
- if $\mathcal{M}$ is strongly connected: θ is computable via the balance equation $\theta = P \cdot \theta$ where $\sum_{s\in S} \theta(s) \;=\; 1$

Fundamental property of finite Markov chains

Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

A bottom strongly connected component will be reached and all its states visited infinitely often.

Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

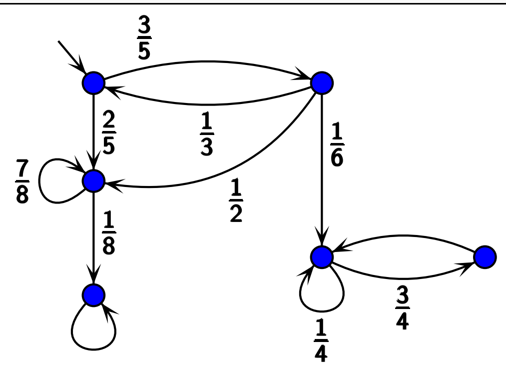
A bottom strongly connected component will be reached and all its states visited infinitely often.

$$Pr^{\mathcal{M}}\Big\{ s_0\, s_1\, s_2 \ldots \in Paths^{\mathcal{M}} : \text{there exists } i \geqslant 0 \text{ and a BSCC } C \text{ s.t. } \underbrace{\forall j \geqslant i.\ s_j \in C}_{\text{eventually forever } C} \ \wedge\ \underbrace{\forall s \in C\ \overset{\infty}{\exists} j.\ s_j = s}_{\text{visit each state in } C \text{ infinitely often}} \Big\} = 1$$

Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

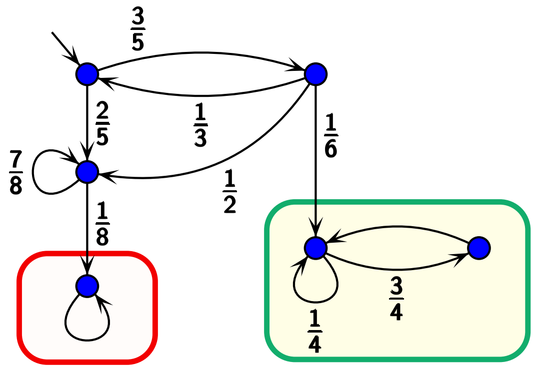
A bottom strongly connected component will be reached and all its states visited infinitely often.



Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

A bottom strongly connected component will be reached and all its states visited infinitely often.



2 BSCCs

Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

A bottom strongly connected component will be reached and all its states visited infinitely often.

long-run distribution:

$\theta(s) > 0$ iff s belongs to some BSCC

Fundamental property of finite Markov chains

Almost surely, i.e., with probability **1**:

A bottom strongly connected component will be reached and all its states visited infinitely often.

long-run distribution:

If s is a state of BSCC B then:

$$\theta(s) = \Pr^{\mathcal{M}}(\Diamond B) \cdot \theta^{B}(s)$$

probability to reach B — long-run probability for s inside B

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*) ⟵
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness

Probabilistic computation tree logic

Probabilistic computation tree logic

PCTL/PCTL* [Hansson/Jonsson 1994]

- probabilistic variants of CTL/CTL*
- contains a probabilistic operator $\mathbb{P}$
  to specify lower/upper probability bounds

Probabilistic computation tree logic

PCTL/PCTL* [Hansson/Jonsson 1994]

- probabilistic variants of CTL/CTL*
- contains a probabilistic operator $\mathbb{P}$
  to specify lower/upper probability bounds
- operators for expected costs, long-run averages, ...
  not considered here, but can be added

Syntax of PCTL*

state formulas:

$$\Phi ::= \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \quad \ldots$$

path formulas:

$$\varphi ::= \ldots$$

Syntax of PCTL*

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_{\mathrm{I}}(\varphi)$$

path formulas:

$$\varphi \ ::= \ \ldots$$

where $a \in AP$ is an atomic proposition

$\mathrm{I} \subseteq [0,1]$ is a probability interval

Syntax of PCTL*

PCTL-20

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_{\mathrm{I}}(\varphi)$$

path formulas:

$$\varphi \ ::= \ \ldots$$

where $a \in AP$ is an atomic proposition

$\mathrm{I} \subseteq [0,1]$ is a probability interval

qualitative properties: $\mathbb{P}_{>0}(\varphi)$ or $\mathbb{P}_{=1}(\varphi)$

quantitative properties: e.g., $\mathbb{P}_{>0.5}(\varphi)$ or $\mathbb{P}_{\leqslant 0.01}(\varphi)$

Syntax of PCTL* path formulas

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_{\mathrm{I}}(\varphi)$$

path formulas:

$$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \ \ldots$$

↑ state formula

Syntax of PCTL* path formulas

PCTL-22

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_{\mathrm{I}}(\varphi)$$

path formulas:

$$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \ \ldots$$

$\bigcirc \ \widehat{=}$ next

Syntax of PCTL* path formulas

state formulas:

$$\Phi \;::=\; \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

$\bigcirc \mathrel{\widehat{=}}$ next $\qquad \mathsf{U} \mathrel{\widehat{=}}$ until

Syntax of PCTL* path formulas

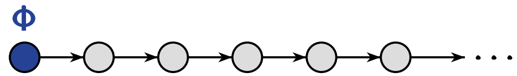
state formulas:

$$\Phi \;::=\; \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

state formula



Syntax of PCTL* path formulas

state formulas:

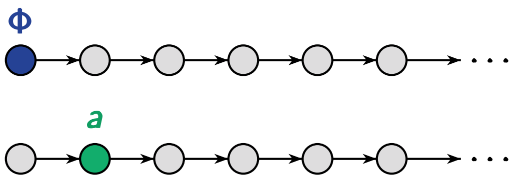
$$\Phi \;::=\; \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

state formula

next $\bigcirc a$



Syntax of PCTL* path formulas

state formulas:

$$\Phi \;::=\; \mathit{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

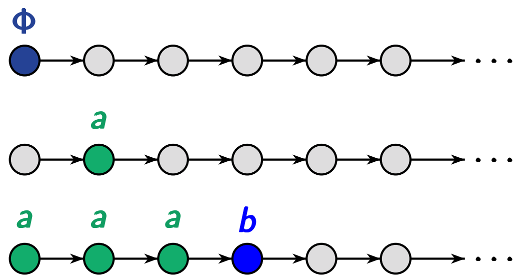
path formulas:

$$\varphi \;::=\; \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

state formula

next $\bigcirc a$

until $a \,\mathsf{U}\, b$



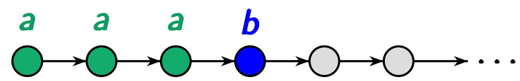
Derived path operators: eventually, always

Derived path operators: eventually, always

syntax of path formulas:

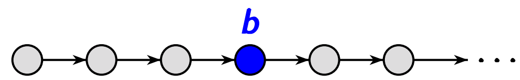
$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2$

until $a \mathbf{U} b$



eventually

$\Diamond b \stackrel{\text{def}}{=} \mathit{true} \, \mathbf{U} \, b$

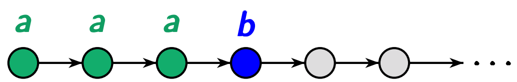


Derived path operators: eventually, always

syntax of path formulas:

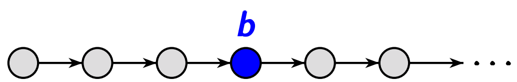
$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2$

until $a \mathbf{U} b$



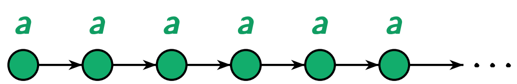
eventually

$\Diamond b \stackrel{\text{def}}{=} \mathit{true} \, \mathbf{U} \, b$



always

$\Box a \stackrel{\text{def}}{=} \neg\Diamond\neg a$



Semantics of PCTL*

Semantics of PCTL*

Let $\mathcal{M} = (S, P, AP, L)$ be a Markov chain.

Define by structural induction:

- a satisfaction relation $\models$ for states $s \in S$ and **PCTL*** state formulas
- a satisfaction relation $\models$ for infinite paths π in $\mathcal{M}$ and **PCTL*** path formulas

Semantics of PCTL*

$s \models \mathit{true}$

$s \models a$ iff $a \in L(s)$

$s \models \neg\Phi$ iff $s \not\models \Phi$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models \mathbb{P}_I(\varphi)$ iff $Pr^{\mathcal{M}}(s, \varphi) \in I$

Semantics of PCTL*

$s \models \mathit{true}$

$s \models a$ iff $a \in L(s)$

$s \models \neg\Phi$ iff $s \not\models \Phi$

$s \models \Phi_1 \wedge \Phi_2$ iff $s \models \Phi_1$ and $s \models \Phi_2$

$s \models \mathbb{P}_I(\varphi)$ iff $Pr^{\mathcal{M}}(s, \varphi) \in I$

↑ probability measure of the set of paths π with $\pi \models \varphi$

when s is viewed as the unique starting state

Semantics of PCTL* path formulas

let $\pi = s_0\, s_1\, s_2\, s_3 \ldots$ be an infinite path in $\mathcal{M}$

Semantics of PCTL* path formulas

let $\pi = s_0\, s_1\, s_2\, s_3 \ldots$ be an infinite path in $\mathcal{M}$

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & s_0 \models \Phi \\
\pi \models \neg\varphi & \text{iff} & \pi \not\models \varphi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \bigcirc\varphi & \text{iff} & s_1\, s_2\, s_3 \ldots \models \varphi \\
\pi \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \text{there exists } \ell \geqslant 0 \text{ such that} \\
 & & s_\ell\, s_{\ell+1}\, s_{\ell+2} \ldots \models \varphi_2 \\
 & & s_i\, s_{i+1}\, s_{i+2} \ldots \models \varphi_1 \quad \text{for } 0 \leqslant i < \ell
\end{array}
$$

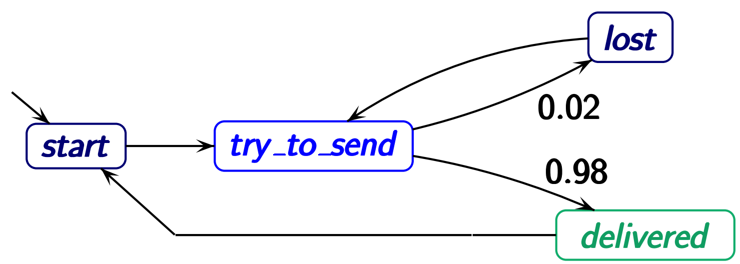
Examples for PCTL*-specifications

Examples for PCTL*-specifications

communication protocol:

$$\mathbb{P}_{=1}\big(\, \Box(\, \textit{try\_to\_send} \;\longrightarrow\; \mathbb{P}_{\geqslant 0.9}(\bigcirc \textit{delivered})\,)\,\big)$$

$$\mathbb{P}_{=1}\big(\, \Box(\, \textit{try\_to\_send} \;\longrightarrow\; \neg \textit{start} \,\mathsf{U}\, \textit{delivered}\,)\,\big)$$



Examples for PCTL*-specifications

communication protocol:

$$\mathbb{P}_{=1}\big(\, \Box(\, \textit{try\_to\_send} \;\longrightarrow\; \mathbb{P}_{\geqslant 0.9}(\bigcirc \textit{delivered})\,)\,\big)$$

$$\mathbb{P}_{=1}\big(\, \Box(\, \textit{try\_to\_send} \;\longrightarrow\; \neg \textit{start} \,\mathsf{U}\, \textit{delivered}\,)\,\big)$$

leader election protocol for n processes:

$$\mathbb{P}_{=1}\big(\, \Diamond\, \textit{leader\_elected}\,\big)$$

$$\mathbb{P}_{\geqslant 0.9}\big(\, \bigvee_{i \leqslant n} \bigcirc^i \textit{leader\_elected}\,\big)$$

PCTL* model checking for DTMC

PCTL-80

PCTL* model checking for DTMC

PCTL-80

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL* state formula Φ

task: check whether $s_0 \models \Phi$

PCTL* model checking for DTMC

PCTL-80

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL* state formula Φ

task: check whether $s_0 \models \Phi$

main procedure as for CTL*:

recursively compute the satisfaction sets

$$Sat(\Psi) = \{ s \in S : s \models \Psi \}$$

for all sub-state formulas Ψ of Φ

Recursive computation of the satisfaction sets

PCTL-85

Recursive computation of the satisfaction sets

$$\begin{aligned} Sat(true) &= S \quad \text{state space of } \mathcal{M} \\ Sat(a) &= \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) &= Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) &= S \setminus Sat(\Phi) \end{aligned}$$

Recursive computation of the satisfaction sets

$$\begin{aligned} Sat(true) &= S \quad \text{state space of } \mathcal{M} \\ Sat(a) &= \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) &= Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) &= S \setminus Sat(\Phi) \\ Sat(\mathbb{P}_I(\varphi)) &= \{s \in S : \Pr^{\mathcal{M}}(s, \varphi) \in I\} \end{aligned}$$

Recursive computation of the satisfaction sets

$$\begin{aligned} Sat(true) &= S \quad \text{state space of } \mathcal{M} \\ Sat(a) &= \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) &= Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) &= S \setminus Sat(\Phi) \\ Sat(\mathbb{P}_I(\varphi)) &= \{s \in S : \Pr^{\mathcal{M}}(s, \varphi) \in I\} \end{aligned}$$

special case: $\varphi = \Diamond\Phi$

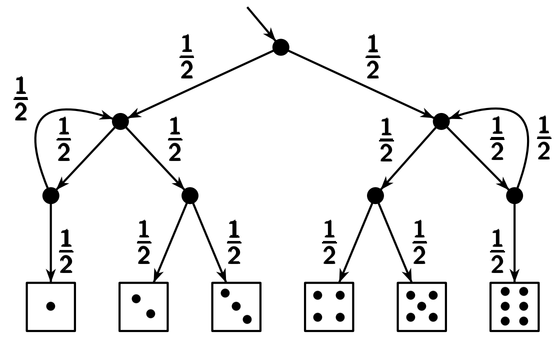
Recursive computation of the satisfaction sets

$$\begin{aligned} Sat(true) &= S \quad \text{state space of } \mathcal{M} \\ Sat(a) &= \{s \in S : a \in L(s)\} \\ Sat(\Phi_1 \wedge \Phi_2) &= Sat(\Phi_1) \cap Sat(\Phi_2) \\ Sat(\neg\Phi) &= S \setminus Sat(\Phi) \\ Sat(\mathbb{P}_I(\varphi)) &= \{s \in S : \Pr^{\mathcal{M}}(s, \varphi) \in I\} \end{aligned}$$

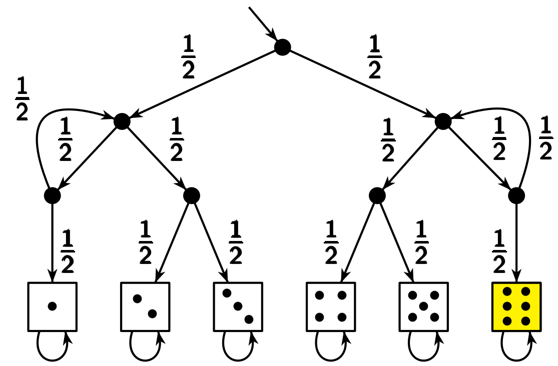
special case: $\varphi = \Diamond\Phi$

1. compute recursively $Sat(\Phi)$
2. compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond\Phi)$ by solving a linear equation system

Simulating a dice by a coin [Knuth]



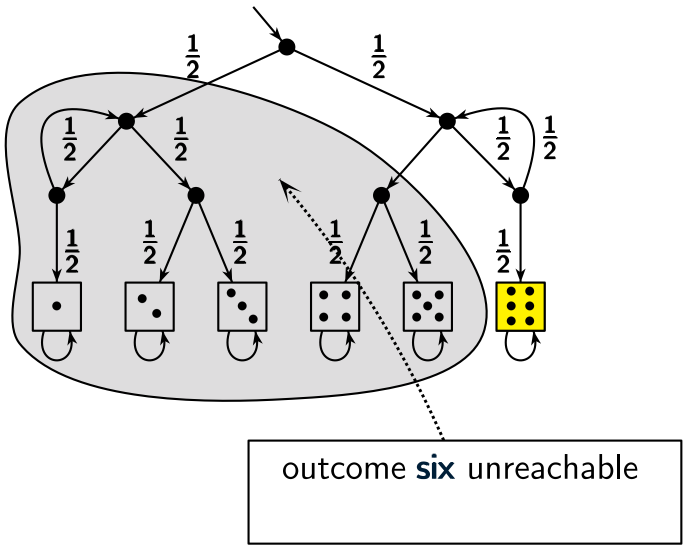
Simulating a dice by a coin [Knuth]



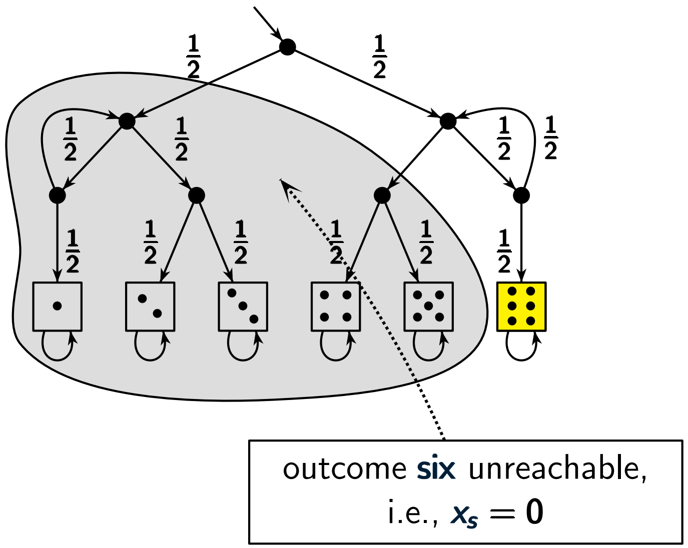
probability for the outcome **six**

$$\Pr^{\mathcal{M}}(\lozenge\, \mathsf{six}\,) = \textbf{?}$$

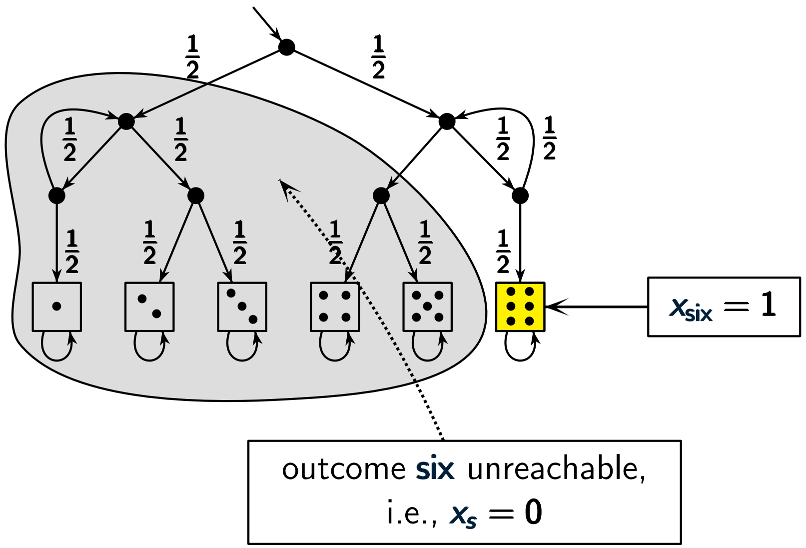
Simulating a dice by a coin [Knuth]



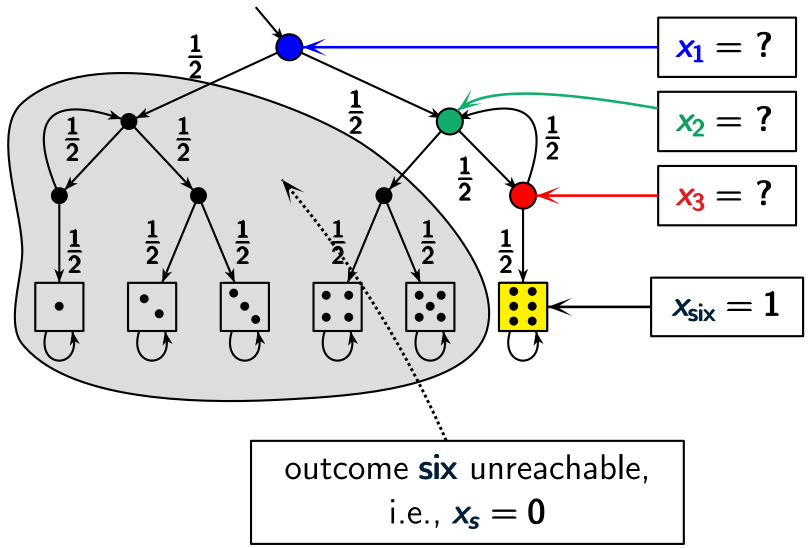
Simulating a dice by a coin [Knuth]



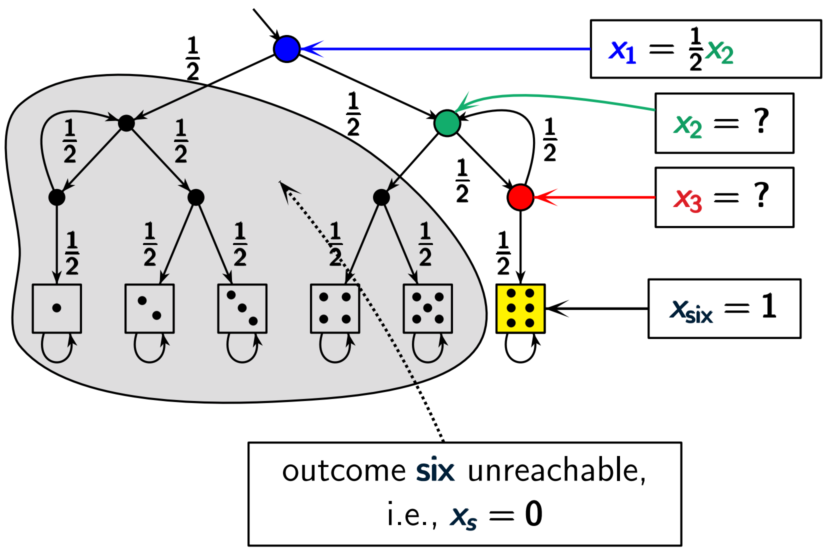
Simulating a dice by a coin [Knuth] PCTL-90



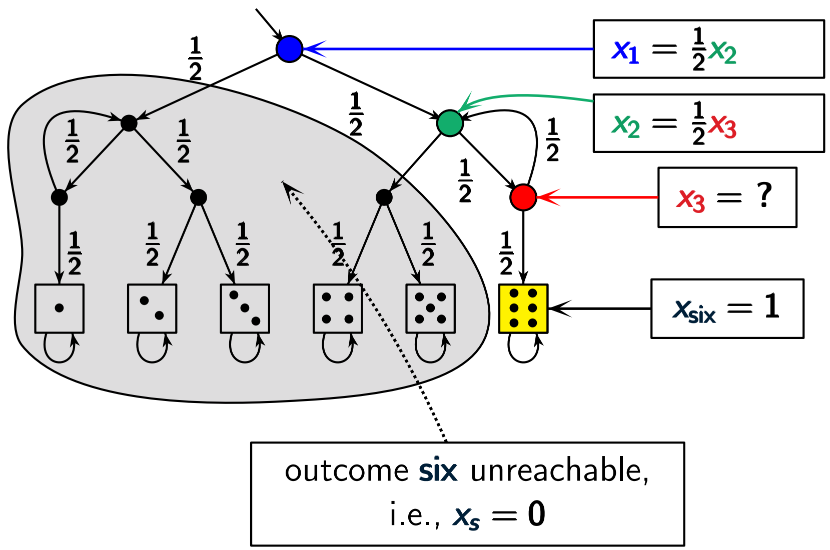
Simulating a dice by a coin [Knuth] PCTL-90



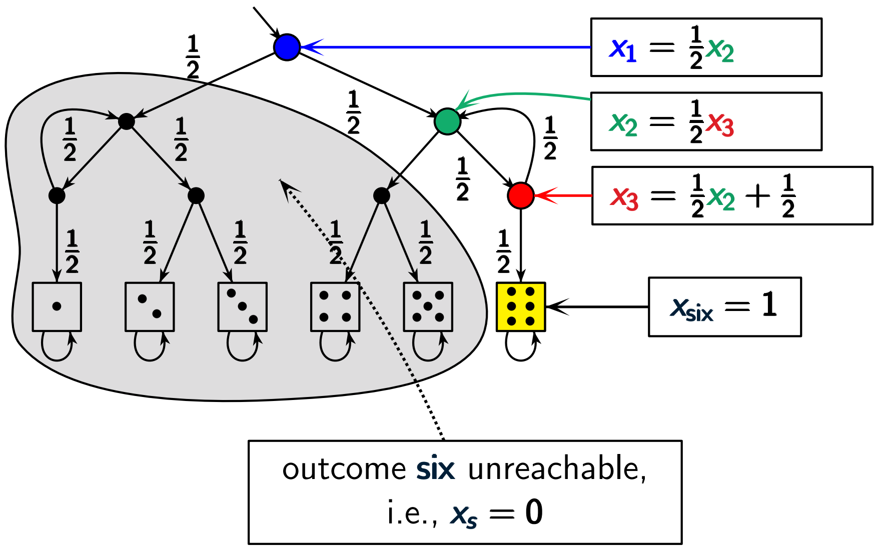
Simulating a dice by a coin [Knuth] PCTL-90



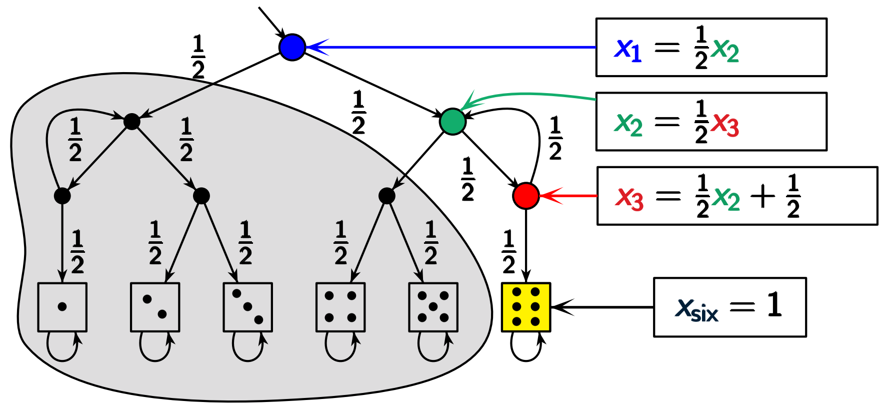
Simulating a dice by a coin [Knuth] PCTL-90



Simulating a dice by a coin [Knuth] PCTL-90

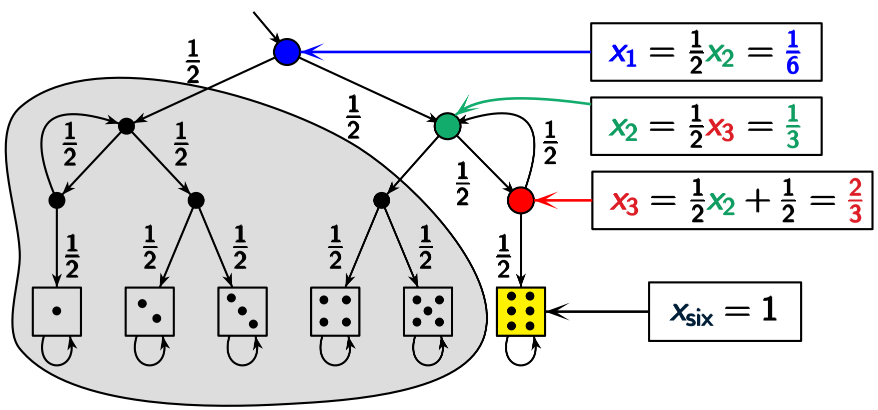


Simulating a dice by a coin [Knuth] PCTL-90



$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Simulating a dice by a coin [Knuth] PCTL-90



$Pr^{\mathcal{M}}(\Diamond \text{ six })$
$= x_1 = \frac{1}{6}$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2} \end{pmatrix}$$

Computing reachability probabilities PCTL-100

Computing reachability probabilities

given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

Computing reachability probabilities

given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

   $$S^0 = \{s \in S : x_s = 0\}$$
   $$S^1 = \{s \in S : x_s = 1\}$$

2. ...

Computing reachability probabilities

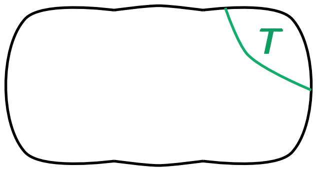
given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

   $$S^0 = \{s \in S : x_s = 0\}$$
   $$S^1 = \{s \in S : x_s = 1\}$$

2. ...



state space S

Computing reachability probabilities

given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

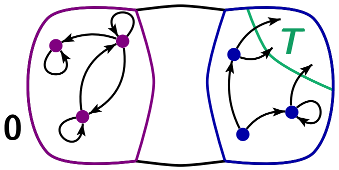
task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

   $$S^0 = \{s \in S : x_s = 0\}$$
   $$S^1 = \{s \in S : x_s = 1\}$$

2. ...

$x_s = 0$



state space S

$x_s = 1$

Computing reachability probabilities

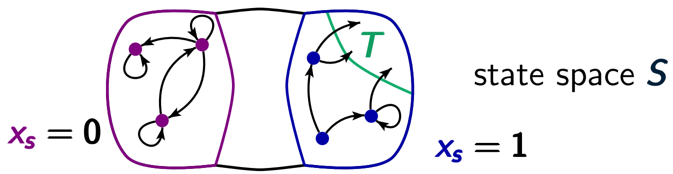
given: DTMC $\mathcal{M} = (S, P, \dots)$ and $T \subseteq S$

task: compute $x_s = Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\}$$

2. …



Computing reachability probabilities

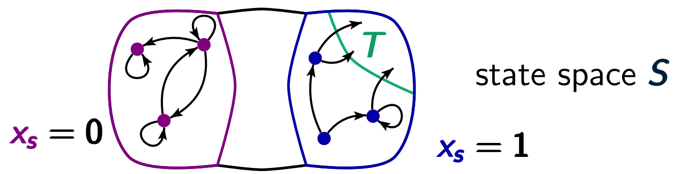
given: DTMC $\mathcal{M} = (S, P, \dots)$ and $T \subseteq S$

task: compute $x_s = Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\} = \{s : s \not\models \exists (\neg T) \, \mathsf{U} \, S^0\}$$

2. …



Computing reachability probabilities

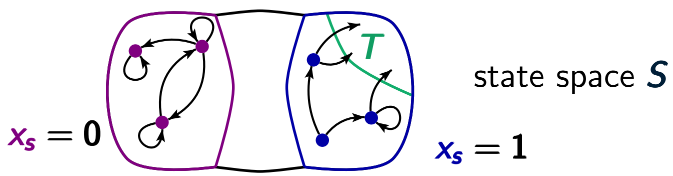
given: DTMC $\mathcal{M} = (S, P, \dots)$ and $T \subseteq S$

task: compute $x_s = Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1 ⟵ graph algorithms

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\} = \{s : s \not\models \exists (\neg T) \, \mathsf{U} \, S^0\}$$

2. …



Computing reachability probabilities

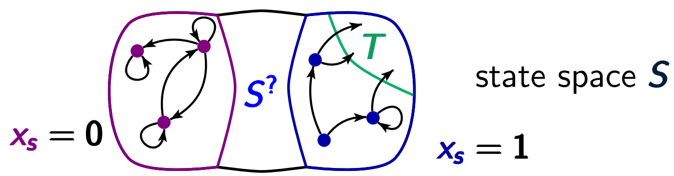
given: DTMC $\mathcal{M} = (S, P, \dots)$ and $T \subseteq S$

task: compute $x_s = Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1 ⟵ graph algorithms

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\} = \{s : s \not\models \exists (\neg T) \, \mathsf{U} \, S^0\}$$

2. compute x_s for $s \in S^? = S \setminus (S^0 \cup S^1)$



Computing reachability probabilities

given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

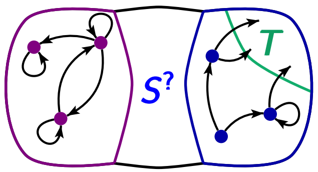
task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1 ⟵ graph algorithms

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\} = \{s : s \not\models \exists (\neg T) \,\mathsf{U}\, S^0\}$$

2. compute x_s for $s \in S^? = \{s : 0 < x_s < 1\}$

$x_s = 0$



state space S

$x_s = 1$

Computing reachability probabilities

given: DTMC $\mathcal{M} = (S, P, \ldots)$ and $T \subseteq S$

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S$

1. compute S^0 and S^1

$$S^0 = \{s \in S : x_s = 0\} = \{s : s \not\models \exists \Diamond T\}$$
$$S^1 = \{s \in S : x_s = 1\} = \{s : s \not\models \exists (\neg T) \,\mathsf{U}\, S^0\}$$

2. compute x_s for $s \in S^? = \{s : 0 < x_s < 1\}$

by solving a linear equation system

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \sum_{s' \in S^?} P(s, s') \cdot x_{s'} + P(s, S^1)$$

$$P(s, S^1) = \sum_{u \in S^1} P(s, u)$$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \sum_{s' \in S^?} P(s, s') \cdot x_{s'} + P(s, S^1)$$

probability for paths of the form

$s \, \underbrace{u_1 \, u_2 \ldots u_k}_{u_j \in S^1} \, t$ with $t \in T$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \underbrace{\sum_{s' \in S^?} P(s,s') \cdot x_{s'}} + P(s, S^1)$$

probability for paths of the form

$$\underbrace{s\ s_1\ s_2 \ldots s_m}_{s_i \in S^?}\ \underbrace{u_1\ u_2 \ldots u_k}_{u_j \in S^1}\ t \quad \text{with } t \in T$$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \sum_{s' \in S^?} P(s,s') \cdot x_{s'} + P(s, S^1)$$

$$x = A \cdot x + b$$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \sum_{s' \in S^?} P(s,s') \cdot x_{s'} + P(s, S^1)$$

$$x = A \cdot x + b$$

matrix $A = \big(P(s,s')\big)_{s,s' \in S^?}$

vectors $x = \big(x_s\big)_{s \in S^?}$

$b = \big(P(s,S^1)\big)_{s \in S^?}$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \Diamond T)$ for all $s \in S^?$

by solving the equation system:

$$x_s = \sum_{s' \in S^?} P(s,s') \cdot x_{s'} + P(s, S^1)$$

$$x = A \cdot x + b$$

iff

$$(\mathbf{I} - A) \cdot x = b$$

matrix $A = \big(P(s,s')\big)_{s,s' \in S^?}$

vectors $x = \big(x_s\big)_{s \in S^?}$

$b = \big(P(s,S^1)\big)_{s \in S^?}$

identity matrix $\mathbf{I}$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \lozenge T)$ for all $s \in S^?$

by solving the equation system:

$$x_s \;=\; \sum_{s' \in S^?} P(s,s') \cdot x_{s'} \;+\; P(s, S^1)$$

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

linear equation system with non-singular matrix $\mathbf{I} - A$

Computing reachability probabilities

task: compute $x_s = \Pr^{\mathcal{M}}(s, \lozenge T)$ for all $s \in S^?$

by solving the equation system:

$$x_s \;=\; \sum_{s' \in S^?} P(s,s') \cdot x_{s'} \;+\; P(s, S^1)$$

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

linear equation system with non-singular matrix $\mathbf{I} - A$

$\Downarrow$

unique solution

PCTL

sublogic of **PCTL*** where only path formulas of the form $\bigcirc\Phi$ and $\Phi_1 \,\mathsf{U}\, \Phi_2$ are allowed

PCTL

sublogic of **PCTL*** where only path formulas of the form $\bigcirc\Phi$ and $\Phi_1 \,\mathsf{U}\, \Phi_2$ are allowed

state formulas:

$$\Phi \;::=\; \mathit{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \bigcirc\Phi \;\big|\; \Phi_1 \,\mathsf{U}\, \Phi_2$$

PCTL

sublogic of **PCTL*** where only path formulas of the form $\bigcirc\Phi$ and $\Phi_1 \,\mathsf{U}\, \Phi_2$ are allowed

state formulas:

$$\Phi \;::=\; true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

$$\mathbb{P}_I(\Diamond\Phi) \stackrel{\text{def}}{=} \mathbb{P}_I(true \,\mathsf{U}\, \Phi)$$

PCTL
PCTL-150

sublogic of **PCTL*** where only path formulas of the form $\bigcirc\Phi$ and $\Phi_1 \,\mathsf{U}\, \Phi_2$ are allowed

state formulas:

$$\Phi \;::=\; true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \;::=\; \bigcirc\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2 \mid \Diamond\Phi \mid \Box\Phi$$

$$\mathbb{P}_I(\Diamond\Phi) \stackrel{\text{def}}{=} \mathbb{P}_I(true \,\mathsf{U}\, \Phi)$$

e.g., $\mathbb{P}_{<0.4}(\Box\Phi) \stackrel{\text{def}}{=} \mathbb{P}_{>0.6}(\Diamond\neg\Phi)$

note: $\Pr^{\mathcal{M}}(s, \Box\Phi) = 1 - \Pr^{\mathcal{M}}(s, \Diamond\neg\Phi)$

PCTL model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL formula Φ

task: check whether $s_0 \models \Phi$

recursive computation of $Sat(\Psi) = \{s \in S : s \models \Psi\}$ for all subformulas Ψ of Φ

- treatment of the propositional logic fragment: $\surd$
- treatment of the probability operator $\mathbb{P}_I(\varphi)$

  graph algorithms + matrix/vector operations

  ↑

  next: matrix/vector multiplication
  until: linear equation system

PCTL model checking
PCTL-170

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL formula Φ

task: check whether $s_0 \models \Phi$

recursive computation of $Sat(\Psi) = \{s \in S : s \models \Psi\}$ for all subformulas Ψ of Φ

- treatment of the propositional logic fragment: $\surd$
- treatment of the probability operator $\mathbb{P}_I(\varphi)$

  graph algorithms + matrix/vector operations

time complexity: $\mathcal{O}(\,\mathbf{poly}(\mathcal{M}) \cdot |\Phi|\,)$

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

recursive computation of $Sat(\Psi) = \{s \in S : s \models \Psi\}$
for all sub-state formulas Ψ of Φ

- treatment of the propositional logic fragment: $\surd$
- treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'
↑
path formula without
state formulas $\mathbb{P}_I(\ldots)$

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

recursive computation of $Sat(\Psi) = \{s \in S : s \models \Psi\}$
for all sub-state formulas Ψ of Φ

- treatment of the propositional logic fragment: $\surd$
- treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'

... automata-based approach for φ' ...

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'

by replacing each maximal state-subformula
of φ with a fresh atomic proposition

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'

by replacing each maximal state-subformula
of φ with a fresh atomic proposition

$$\Diamond\big(a \,\mathsf{U}\, \mathbb{P}_{\geqslant 0.7}(\Box\Diamond b) \ \wedge \Box\ \mathbb{P}_{<0.3}(\bigcirc\Box c)\ \big)$$

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'

by replacing each maximal state-subformula of φ with a fresh atomic proposition

$$\Diamond(a \, \mathsf{U} \, \mathbb{P}_{\geqslant 0.7}(\Box\Diamond b) \wedge \Box \, \mathbb{P}_{<0.3}(\bigcirc\Box c))$$

PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL*-state formula Φ
task: check whether $s_0 \models \Phi$

treatment of the probability operator $\mathbb{P}_I(\varphi)$

PCTL* path formula $\varphi \rightsquigarrow$ LTL formula φ'

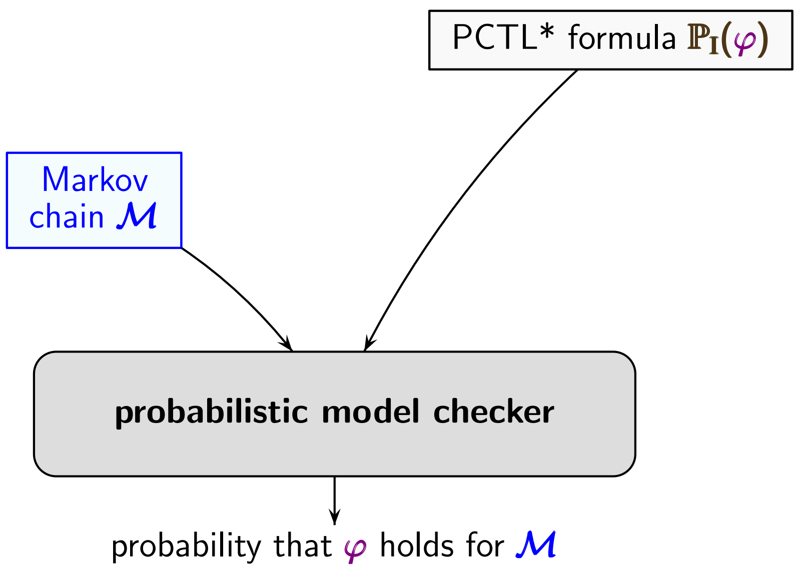
by replacing each maximal state-subformula of φ with a fresh atomic proposition

$$\Diamond(a \, \mathsf{U} \, \mathbb{P}_{\geqslant 0.7}(\Box\Diamond b) \wedge \Box \, \mathbb{P}_{<0.3}(\bigcirc\Box c))$$

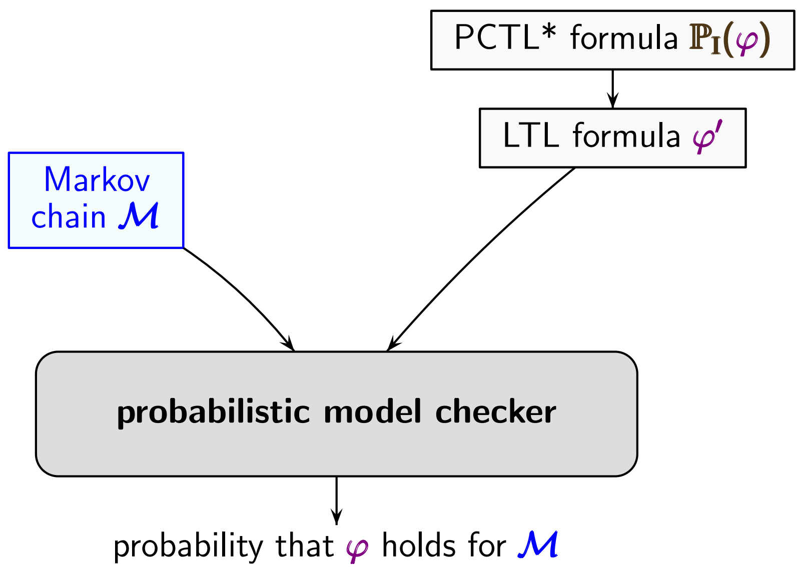
$$\Downarrow$$

$$\Diamond(a \, \mathsf{U} \, d \wedge \Box e)$$

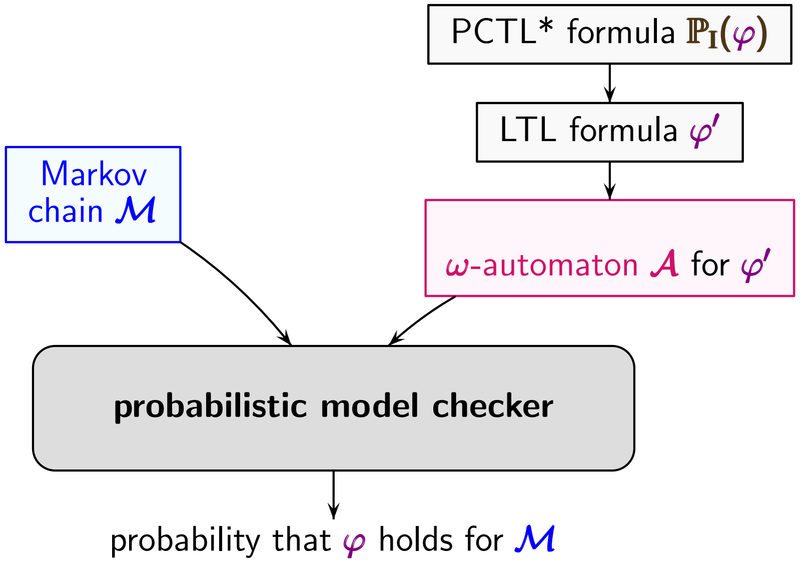
PCTL* model checking



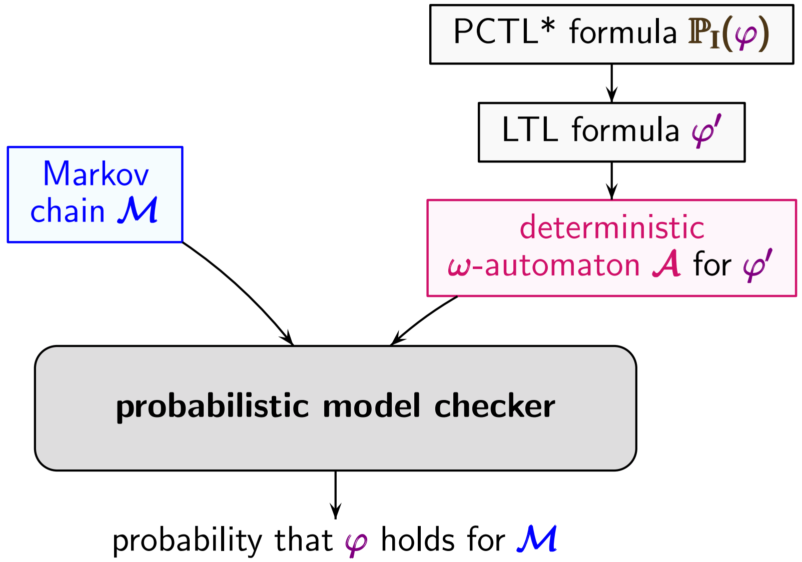
PCTL* model checking



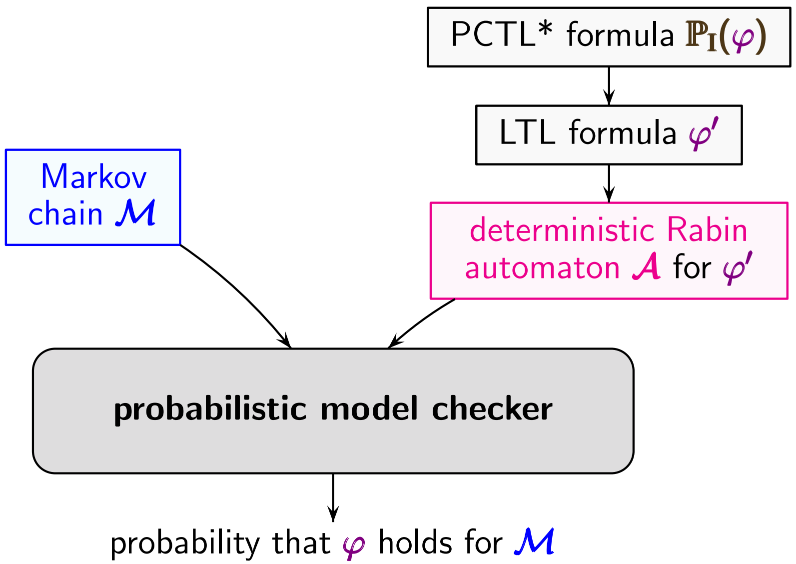
PCTL* model checking



PCTL* model checking



PCTL* model checking



Deterministic Rabin automata (DRA)

Deterministic Rabin automata (DRA)

A DRA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ where

- Q finite state space
- $q_0 \in Q$ initial state
- Σ alphabet
- $\delta : Q \times \Sigma \longrightarrow Q$ deterministic transition function

Deterministic Rabin automata (DRA)

A DRA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ where

- Q finite state space
- $q_0 \in Q$ initial state
- Σ alphabet
- $\delta : Q \times \Sigma \longrightarrow Q$ deterministic transition function
- acceptance condition Acc is a set of pairs (L, U) with $L, U \subseteq Q$

Deterministic Rabin automata (DRA)

A DRA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ where

- Q finite state space
- $q_0 \in Q$ initial state
- Σ alphabet
- $\delta : Q \times \Sigma \longrightarrow Q$ deterministic transition function
- acceptance condition Acc is a set of pairs (L, U) with $L, U \subseteq Q$, say $Acc = \{(L_1, U_1), \ldots, (L_k, U_k)\}$

Deterministic Rabin automata (DRA)

A DRA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ where

- Q finite state space
- $q_0 \in Q$ initial state
- Σ alphabet
- $\delta : Q \times \Sigma \longrightarrow Q$ deterministic transition function
- acceptance condition Acc is a set of pairs (L, U) with $L, U \subseteq Q$, say $Acc = \{(L_1, U_1), \ldots, (L_k, U_k)\}$

semantics of the acceptance condition:

$$\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \ \wedge \ \Box\Diamond U_i)$$

Accepted language of a DRA

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ be an DRA where

$$Acc = \{(L_1, U_1), \ldots, (L_k, U_k)\} \qquad L_i, U_i \subseteq Q$$

accepted language:

$$\mathcal{L}_\omega(\mathcal{A}) = \{ \sigma \in \Sigma^\omega : \text{the run for } \sigma \text{ in } \mathcal{A} \text{ fulfills } Acc \}$$

Accepted language of a DRA

PCTL-355

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ be an DRA where

$$Acc = \{(L_1, U_1), \ldots, (L_k, U_k)\} \qquad L_i, U_i \subseteq Q$$

accepted language:

$$\mathcal{L}_\omega(\mathcal{A}) = \{ \sigma \in \Sigma^\omega : \text{the run for } \sigma \text{ in } \mathcal{A} \text{ fulfills } Acc \}$$

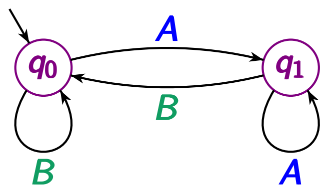
Let $\rho = q_0\, q_1\, q_2 \ldots$ is the run for some infinite word σ.

ρ fulfills Acc iff

$$\exists i \in \{1, \ldots, k\}.\ \inf(\rho) \cap L_i = \varnothing \ \wedge\ \inf(\rho) \cap U_i \neq \varnothing$$

where $\inf(\rho) = \{q \in Q : \overset{\infty}{\exists} \ell \in \mathbb{N}.\ q = q_\ell\}$

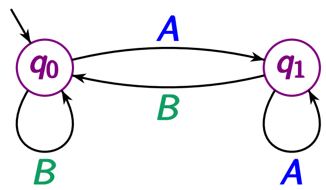
Example: DRA



$$Acc = \{(\{q_0\}, \{q_1\})\}$$

Example: DRA

PCTL-360



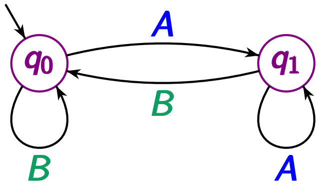
$$Acc = \{(\{q_0\}, \{q_1\})\}$$

$$\widehat{=}\ \Diamond\Box\neg q_0 \ \wedge\ \Box\Diamond q_1$$

$\Diamond\Box$ "eventually forever"

$\Box\Diamond$ "infinitely often"

Example: DRA



$Acc = \{(\{q_0\}, \{q_1\})\}$

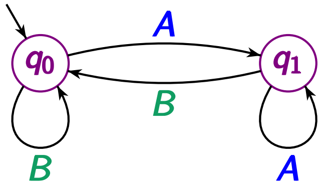
$\hat{=} \ \Diamond\Box\neg q_0 \ \wedge \ \Box\Diamond q_1$

accepted language: $(A + B)^* A^\omega$

$\Diamond\Box$ "eventually forever"

$\Box\Diamond$ "infinitely often"

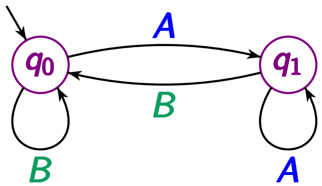
Example: DRA



$Acc = \{(\{q_0\}, \{q_1\})\}$

$\hat{=} \ \Diamond\Box\neg q_0 \ \wedge \ \Box\Diamond q_1$

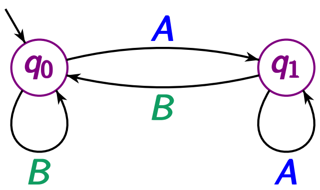
accepted language: $(A + B)^* A^\omega$



$Acc = \{(\varnothing, \{q_1\})\}$

$\Box\Diamond$ "infinitely often"

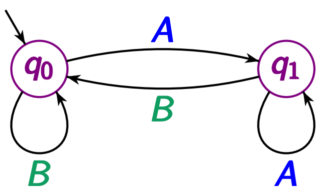
Example: DRA



$Acc = \{(\{q_0\}, \{q_1\})\}$

$\hat{=} \ \Diamond\Box\neg q_0 \ \wedge \ \Box\Diamond q_1$

accepted language: $(A + B)^* A^\omega$

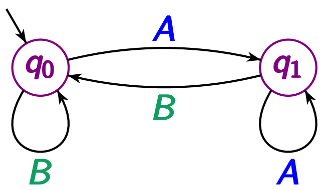


$Acc = \{(\varnothing, \{q_1\})\}$

$\hat{=} \ \Box\Diamond q_1$

$\Box\Diamond$ "infinitely often"

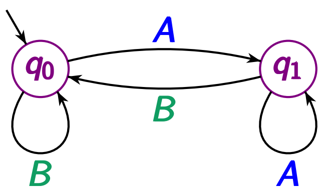
Example: DRA



$Acc = \{(\{q_0\}, \{q_1\})\}$

$\hat{=} \ \Diamond\Box\neg q_0 \ \wedge \ \Box\Diamond q_1$

accepted language: $(A + B)^* A^\omega$



$Acc = \{(\varnothing, \{q_1\})\}$

$\hat{=} \ \Box\Diamond q_1$

accepted language: $(B^* A)^\omega$

Fundamental result: LTL-2-DRA

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Example: $AP = \{a, b\}$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

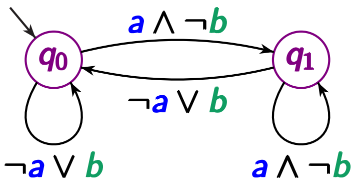
Example: $AP = \{a, b\} \rightsquigarrow \Sigma = \Big\{\varnothing, \{a\}, \{b\}, \{a, b\}\Big\}$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) \;=\; \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Example: $AP = \{a, b\} \;\rightsquigarrow\; \Sigma = \big\{\varnothing, \{a\}, \{b\}, \{a, b\}\big\}$



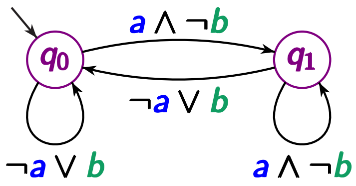
acceptance condition:
$\Diamond\Box\neg q_0 \;\wedge\; \Box\Diamond q_1$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) \;=\; \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Example: $AP = \{a, b\} \;\rightsquigarrow\; \Sigma = \big\{\varnothing, \{a\}, \{b\}, \{a, b\}\big\}$



acceptance condition:
$\Diamond\Box\neg q_0 \;\wedge\; \Box\Diamond q_1$

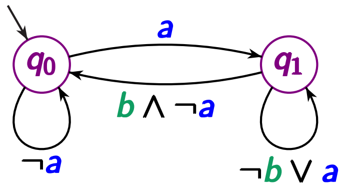
LTL formula $\Diamond\Box(a \wedge \neg b)$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) \;=\; \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Example: $AP = \{a, b\} \;\rightsquigarrow\; \Sigma = \big\{\varnothing, \{a\}, \{b\}, \{a, b\}\big\}$



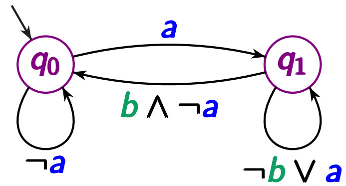
acceptance condition:
$\Diamond\Box\neg q_1 \;\wedge\; \Box\Diamond q_0$

Fundamental result: LTL-2-DRA

For each LTL formula φ over AP there exists a DRA $\mathcal{A}$ with the alphabet $\Sigma = 2^{AP}$ s.t.

$$\mathcal{L}_\omega(\mathcal{A}) \;=\; \{\sigma \in \Sigma^\omega : \sigma \models \varphi\}$$

Example: $AP = \{a, b\} \;\rightsquigarrow\; \Sigma = \big\{\varnothing, \{a\}, \{b\}, \{a, b\}\big\}$

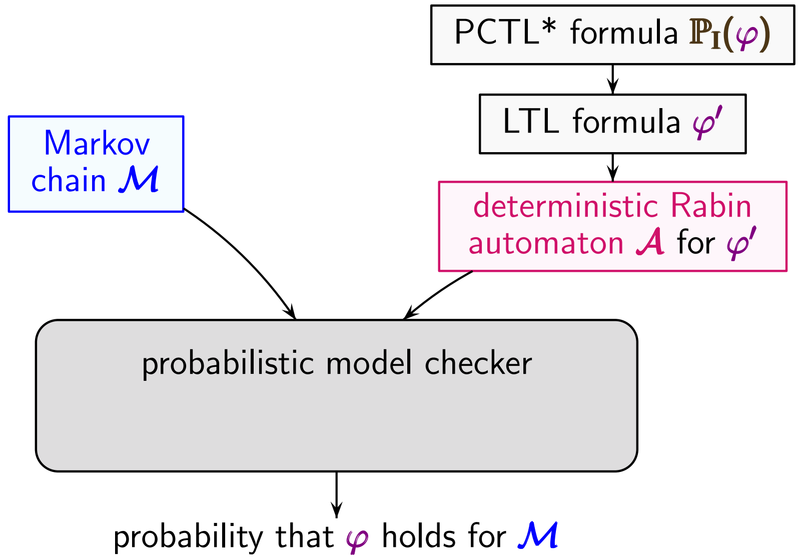


acceptance condition:
$\Diamond\Box\neg q_1 \;\wedge\; \Box\Diamond q_0$

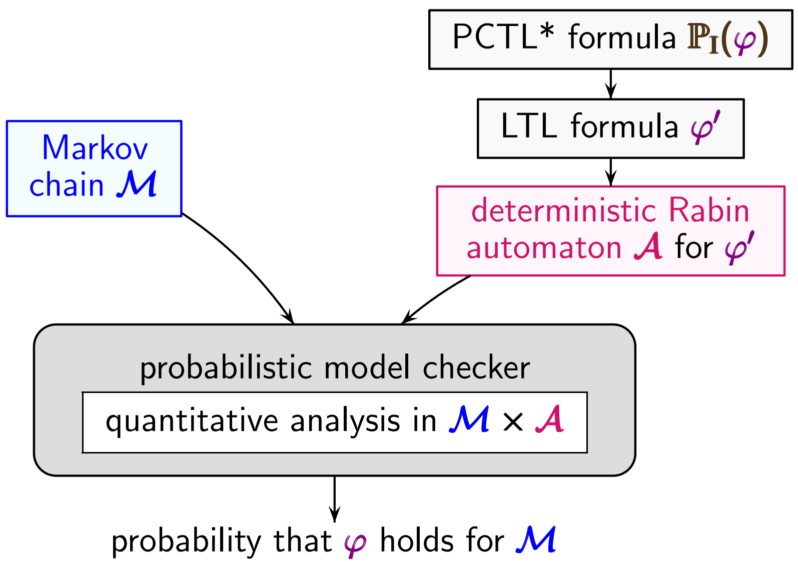
LTL formula
$\Box(a \rightarrow \Diamond(b \wedge \neg a)) \;\wedge\; \Diamond\Box\neg a$

PCTL* model checking

PCTL* model checking



PCTL* model checking



Product of a Markov chain and a DRA

Product of a Markov chain and a DRA

given: Markov chain $\mathcal{M} = (S, P, AP, L)$
DRA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, Acc)$
goal: define a Markov chain $\mathcal{M} \times \mathcal{A}$

Product of a Markov chain and a DRA

given: Markov chain $\mathcal{M} = (S, P, AP, L)$
DRA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, Acc)$
goal: define a Markov chain $\mathcal{M} \times \mathcal{A}$ s.t.

$$\Pr^{\mathcal{M}}(s, \mathcal{A}) = \Pr^{\mathcal{M}}\{\pi \in Paths(s) : trace(\pi) \in \mathcal{L}_\omega(\mathcal{A})\}$$

can be derived by a probabilistic reachability analysis in the product-chain $\mathcal{M} \times \mathcal{A}$

$$trace(s_0\, s_1\, s_2 \ldots) \;=\; L(s_0)\, L(s_1)\, L(s_2) \ldots \in \left(2^{AP}\right)^\omega$$

Product of a Markov chain and a DRA

given: Markov chain $\mathcal{M} = (S, P, AP, L)$
DRA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, Acc)$
idea: define a Markov chain $\mathcal{M} \times \mathcal{A}$ s.t. ...

path π in $\mathcal{M}$

$s_0 \to s_1 \to s_2 \to \ldots$

Product of a Markov chain and a DRA

given: Markov chain $\mathcal{M} = (S, P, AP, L)$
DRA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, Acc)$
idea: define a Markov chain $\mathcal{M} \times \mathcal{A}$ s.t. ...

path π in $\mathcal{M}$

$s_0 \to s_1 \to s_2 \to \ldots$

run for $trace(\pi)$ in $\mathcal{A}$

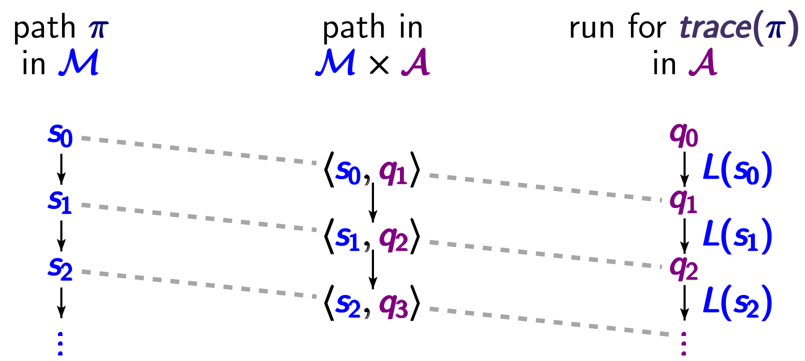
$q_0 \xrightarrow{L(s_0)} q_1 \xrightarrow{L(s_1)} q_2 \xrightarrow{L(s_2)} \ldots$

Product of a Markov chain and a DRA

given: Markov chain $\mathcal{M} = (S, P, AP, L)$

DRA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, Acc)$

idea: define a Markov chain $\mathcal{M} \times \mathcal{A}$ s.t. ...



Fundamental property of the product

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$

For each state s in $\mathcal{M}$, let $q_s = \delta(q_0, L(s))$.

successor state in $\mathcal{A}$ of the initial DRA-state q_0 for the input symbol $L(s) \in 2^{AP}$

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$$

For each state s in $\mathcal{M}$, let $q_s = \delta(q_0, L(s))$.

$Pr^{\mathcal{M}}(s, \mathcal{A})$

probability measure of all paths $\pi \in Paths^{\mathcal{M}}(s)$ such that $trace(\pi) \in \mathcal{L}_\omega(\mathcal{A})$

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$$

For each state s in $\mathcal{M}$, let $q_s = \delta(q_0, L(s))$.

$$Pr^{\mathcal{M}}(s, \mathcal{A}) = Pr^{\mathcal{M}\times\mathcal{A}}\Big(\langle s, q_s \rangle, \bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i) \Big)$$

probability measure of all paths π in the product s.t. $\pi|_{\mathcal{A}}$ satisfies the acceptance condition of $\mathcal{A}$

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$$

For each state s in $\mathcal{M}$, let $q_s = \delta(q_0, L(s))$.

$$\begin{aligned} Pr^{\mathcal{M}}(s, \mathcal{A}) &= Pr^{\mathcal{M}\times\mathcal{A}}\Big(\langle s, q_s \rangle, \bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i) \Big) \\ &= Pr^{\mathcal{M}\times\mathcal{A}}\big(\langle s, q_s \rangle, \Diamond\, accBSCC \big) \end{aligned}$$

Fundamental property of the product

given: Markov chain $\mathcal{M}$ and DRA $\mathcal{A}$ where

$$Acc = \{ (L_1, U_1), (L_2, U_2), \ldots, (L_k, U_k) \}$$

For each state s in $\mathcal{M}$, let $q_s = \delta(q_0, L(s))$.

$$Pr^{\mathcal{M}}(s, \mathcal{A}) = Pr^{\mathcal{M}\times\mathcal{A}}\big(\langle s, q_s \rangle, \Diamond\, accBSCC \big)$$

↑ union of accepting BSCCs in $\mathcal{M} \times \mathcal{A}$ i.e., BSCC C s.t.

$$\exists i \in \{1, \ldots, k\}.\ \ C \cap L_i = \varnothing \ \wedge\ C \cap U_i \neq \varnothing$$

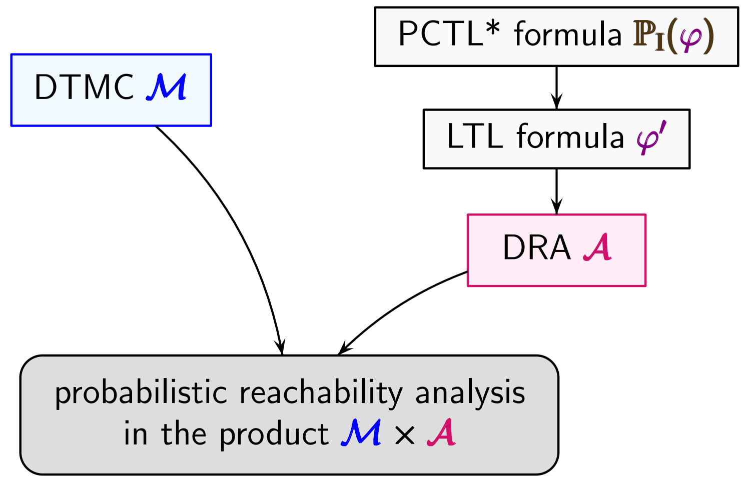
Summary: PCTL* model checking

given: Markov chain $\mathcal{M} = (S, P, AP, L, s_0)$
PCTL* state formula Φ
task: check whether $\mathcal{M} \models \Phi$

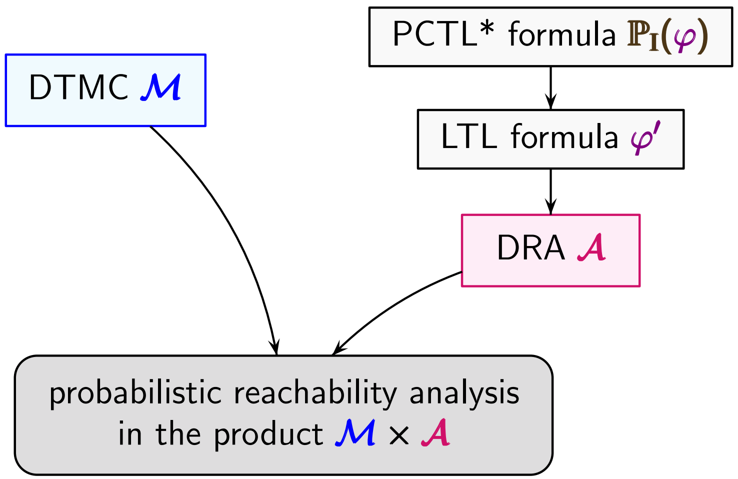
method: bottom-up treatment of sub-state formulas Ψ to compute

$$Sat(\Psi) = \{ s \in S : s \models \Psi \}$$

- propositional logic fragment: obvious
- probability operator $\mathbb{P}_I(\varphi)$ via
  - construction of a DRA $\mathcal{A}$ for φ
  - probabilistic reachability analysis in $\mathcal{M} \times \mathcal{A}$

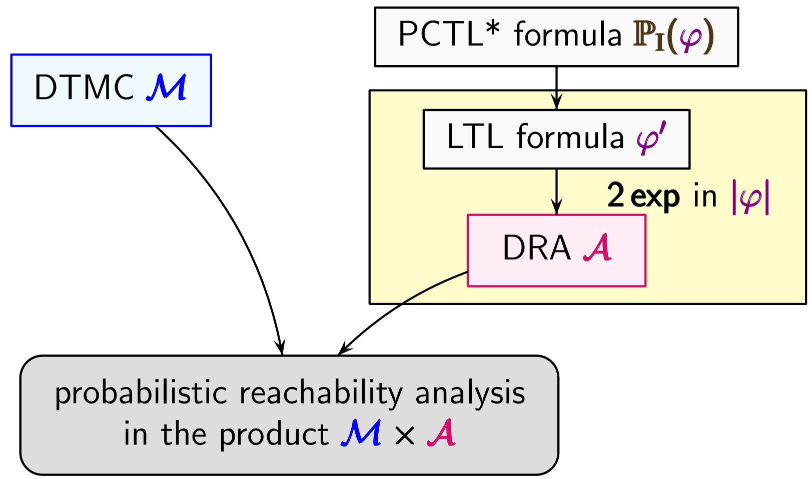


1. graph analysis to compute the accepting BSCCs of the product
2. linear equation system for the probabilities to reach an accepting BSCC



1. graph analysis to compute the accepting BSCCs of the product
2. linear equation system

time complexity: polynomial in the sizes of $\mathcal{M}$ and $\mathcal{A}$



1. graph analysis to compute the accepting BSCCs of the product
2. linear equation system

time complexity: polynomial in the sizes of $\mathcal{M}$ and $\mathcal{A}$

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards ⟵

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness

Markov reward model (MRM)

Markov reward model (MRM)

Markov chain $\mathcal{M} = (S, P, AP, L, rew)$ with a reward function for the states:

$$rew : S \to \mathbb{N}$$

analogously: rewards for edges given by $rew : S \times S \to \mathbb{N}$

Markov reward model (MRM)

Markov chain $\mathcal{M} = (S, P, AP, L, rew)$ with a reward function for the states:

$$rew : S \to \mathbb{N}$$

idea: reward $rew(s)$ will be earned whenever visiting s

analogously: rewards for edges given by $rew : S \times S \to \mathbb{N}$

Markov reward model (MRM)

Markov chain $\mathcal{M} = (S, P, AP, L, rew)$ with a reward function for the states:

$$rew : S \to \mathbb{N}$$

idea: reward $rew(s)$ will be earned whenever visiting s

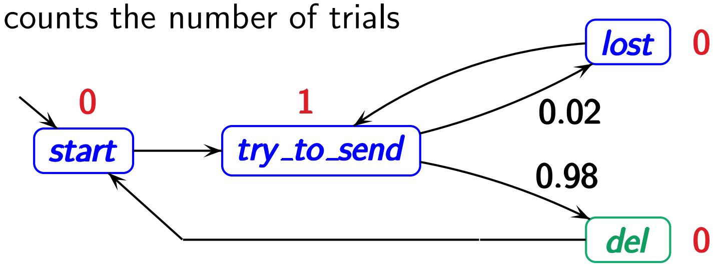
formalization by accumulated rewards of finite paths

$$rew(s_0\, s_1 \ldots s_n) \;=\; \sum_{0 \leqslant i < n} rew(s_i)$$

analogously: rewards for edges given by $rew : S \times S \to \mathbb{N}$

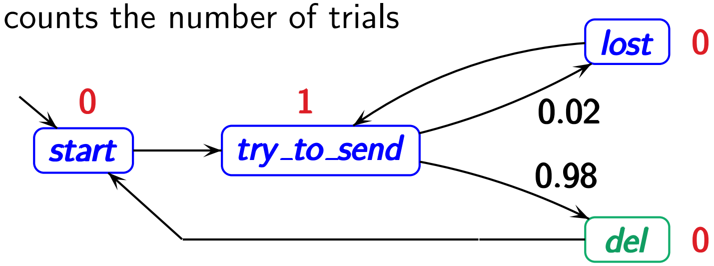
Example: Markov reward model

communication protocol with reward function that counts the number of trials



Example: Markov reward model

communication protocol with reward function that counts the number of trials

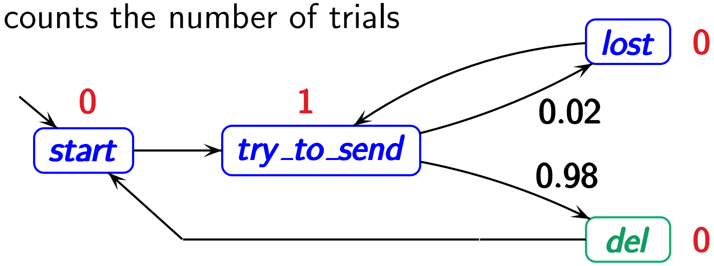


accumulated reward of finite paths, e.g.,

$$rew(start\ try\ lost\ try\ del) \;=\; 2$$

Example: Markov reward model

communication protocol with reward function that counts the number of trials



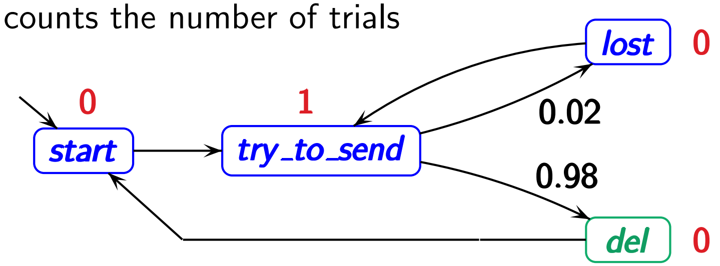
measures of interest, e.g.,

$\Pr^{\mathcal{M}}(\Diamond^{\leqslant 3} del)$ probability to deliver a message within at most **3** trials

↑ reachability with reward bound $\leqslant 3$

Example: Markov reward model

communication protocol with reward function that counts the number of trials



measures of interest, e.g.,

$Pr^{\mathcal{M}}(\Diamond^{\leqslant 3} del)$ probability to deliver a message within at most **3** trials

$\mathbb{E}^{\mathcal{M}}(\Diamond^{=?} del)$ expected number of trials

Reward-based extension of PCTL

Reward-based extension of PCTL

- probability operator for reward-bounded until path formulas $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$
- expected reward operator $\mathbb{C}_{\leqslant r}(\Phi)$

Reward-based extension of PCTL

- probability operator for reward-bounded until path formulas $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$
- expected reward operator $\mathbb{C}_{\leqslant r}(\Phi)$

$s \models \mathbb{C}_{\leqslant r}(\Phi)$ iff the expected accumulated reward on paths from s to a Φ-state is $\leqslant r$

Reward-based extension of PCTL

- probability operator for reward-bounded until path formulas $\mathbb{P}_I(\Phi_1 \mathsf{U}^{\leqslant r} \Phi_2)$
- expected reward operator $\mathbb{C}_{\leqslant r}(\Phi)$

$s \models \mathbb{C}_{\leqslant r}(\Phi)$ iff the expected accumulated reward on paths from s to a Φ-state is $\leqslant r$

e.g., for communication protocols:

$\mathbb{P}_{>0.9}(\Diamond^{\leqslant 3} del)$	probability for delivering the message within at most 3 trials is > 0.9
$\mathbb{C}_{\leqslant 5}(del)$	average number of trials is $\leqslant 5$

Model checking reward-based properties

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \mathsf{U}^{\leqslant r} \Phi_2)$ where $r \in \mathbb{N}$

compute $\Pr^{\mathcal{M}}(s, \Phi_1 \mathsf{U}^{\leqslant i} \Phi_2)$ iteratively for increasing reward bound $i = 0, 1, 2, \ldots, r$

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \mathsf{U}^{\leqslant r} \Phi_2)$ where $r \in \mathbb{N}$

compute $\Pr^{\mathcal{M}}(s, \Phi_1 \mathsf{U}^{\leqslant i} \Phi_2)$ iteratively for increasing reward bound $i = 0, 1, 2, \ldots, r$

Let $\rho(s, i) = \Pr^{\mathcal{M}}(s, \Phi_1 \mathsf{U}^{\leqslant i} \Phi_2)$. Then:

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

Let $\rho(s,i) = Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$. Then:

if $s \models \exists(\Phi_1 \, \mathsf{U} \, \Phi_2) \wedge \neg\Phi_2$ and $i \geqslant rew(s)$ then

$$\rho(s,i) = \sum_{s' \in S} P(s,s') \cdot \rho(s', i - rew(s))$$

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

Let $\rho(s,i) = Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$. Then:

if $s \models \exists(\Phi_1 \, \mathsf{U} \, \Phi_2) \wedge \neg\Phi_2$ and $i \geqslant rew(s)$ then

$$\rho(s,i) = \sum_{s' \in S} P(s,s') \cdot \rho(s', i - rew(s))$$

if $s \models \Phi_2$ then: $\rho(s,i) = 1$

in all other cases: $\rho(s,i) = 0$

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

treatment of the $\mathbb{C}_{\leqslant r}(\Phi)$:

compute the expected accumulated rewards
by solving the linear equation system:

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, \mathsf{U}^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, \mathsf{U}^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

treatment of the $\mathbb{C}_{\leqslant r}(\Phi)$: assuming $Pr^{\mathcal{M}}(\Diamond\Phi) = 1$

compute the expected accumulated rewards
by solving the linear equation system:

$$x_s = rew(s) + \sum_{s' \in S} P(s,s') \cdot x_{s'} \quad \text{if } s \not\models \Phi$$

$$x_s = 0 \quad \text{if } s \models \Phi$$

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, U^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, U^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

treatment of the $\mathbb{C}_{\leqslant r}(\Phi)$: assuming $Pr^{\mathcal{M}}(\Diamond\Phi) = 1$

compute the expected accumulated rewards
by solving the linear equation system

time complexity:

unit rewards: polynomial in $size(\mathcal{M})$ and $\log r$

expected reward: linear equation system

reward-bounded until: repeated squaring

Model checking reward-based properties

treatment of $\mathbb{P}_I(\Phi_1 \, U^{\leqslant r} \, \Phi_2)$ where $r \in \mathbb{N}$

compute $Pr^{\mathcal{M}}(s, \Phi_1 \, U^{\leqslant i} \, \Phi_2)$ iteratively
for increasing reward bound $i = 0, 1, 2, \ldots, r$

treatment of the $\mathbb{C}_{\leqslant r}(\Phi)$: assuming $Pr^{\mathcal{M}}(\Diamond\Phi) = 1$

compute the expected accumulated rewards
by solving the linear equation system

time complexity:

unit rewards: polynomial in $size(\mathcal{M})$ and $\log r$

general case: *NP*-hard [Larroussinie/Sproston]

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness

Markov decision processes (MDP)

Markov decision processes (MDP)

extend Markov chains by nondeterminism

Markov decision processes (MDP)

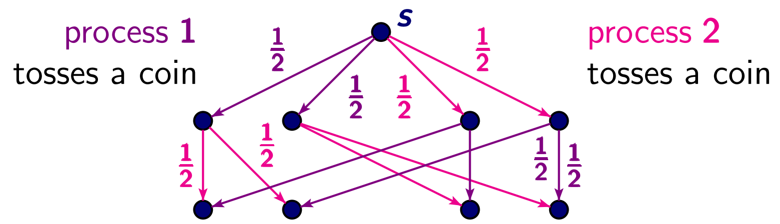
extend Markov chains by nondeterminism

- modeling asynchronous distributed systems by interleaving

Markov decision processes (MDP)

extend Markov chains by nondeterminism

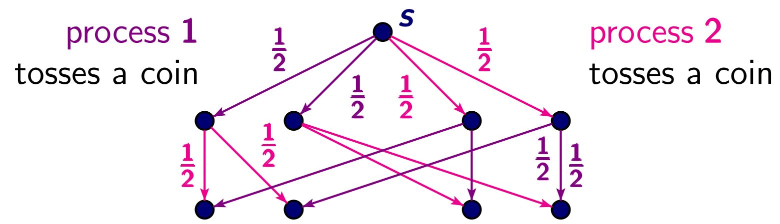
- modeling asynchronous distributed systems by interleaving



Markov decision processes (MDP)

extend Markov chains by nondeterminism

- modeling asynchronous distributed systems by interleaving
- useful for abstraction purposes
- representation of the interface with an unpredictable environment (e.g., human user)



Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

- finite state space S

Markov decision process (MDP)

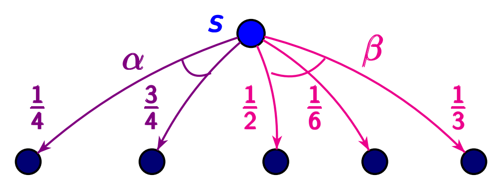
$\mathcal{M} = (S, Act, P, AP, L)$

- finite state space S
- Act finite set of actions

Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

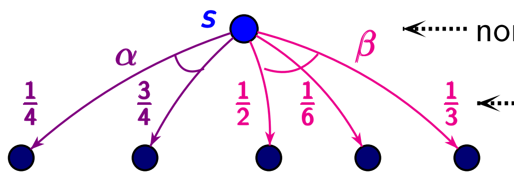
- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0, 1]$ s.t.
  $\forall s \in S \;\; \forall \alpha \in Act. \;\; \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$



Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

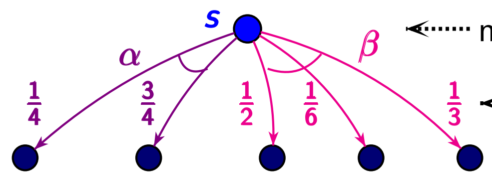
- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.
  $\forall s \in S\ \ \forall \alpha \in Act.\ \ \sum_{s' \in S} P(s,\alpha,s') \in \{0,1\}$



Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

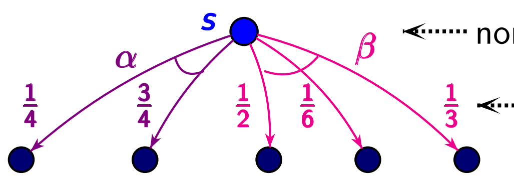
- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.
  $\forall s \in S\ \ \forall \alpha \in Act.\ \ \sum_{s' \in S} P(s,\alpha,s') \in \{0,1\}$
  $\nearrow$ $\alpha \notin Act(s)$ $\nwarrow$ $\alpha \in Act(s)$



Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.
  $\forall s \in S\ \ \forall \alpha \in Act.\ \ \sum_{s' \in S} P(s,\alpha,s') \in \{0,1\}$
  and $Act(s) \neq \varnothing$
  $\nearrow$ $\alpha \notin Act(s)$ $\nwarrow$ $\alpha \in Act(s)$



Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, AP, L)$ + initial distribution, rewards, ...

- finite state space S
- Act finite set of actions
- $P : S \times Act \times S \to [0,1]$ s.t.
  $\forall s \in S\ \ \forall \alpha \in Act.\ \ \sum_{s' \in S} P(s,\alpha,s') \in \{0,1\}$
  and $Act(s) \neq \varnothing$
  $\nearrow$ $\alpha \notin Act(s)$ $\nwarrow$ $\alpha \in Act(s)$
- AP set of atomic propositions
- labeling $L : S \to 2^{AP}$

  ⋮

Randomized mutual exclusion protocol

Randomized mutual exclusion protocol
PMC-72

- 2 concurrent processes P_1, P_2 with 3 phases:

n_i	noncritical actions of process P_i
w_i	waiting phase of process P_i
c_i	critical section of process P_i

Randomized mutual exclusion protocol
PMC-72

- 2 concurrent processes P_1, P_2 with 3 phases:

n_i	noncritical actions of process P_i
w_i	waiting phase of process P_i
c_i	critical section of process P_i

- competition of both processes are waiting

Randomized mutual exclusion protocol
PMC-72

- 2 concurrent processes P_1, P_2 with 3 phases:

n_i	noncritical actions of process P_i
w_i	waiting phase of process P_i
c_i	critical section of process P_i

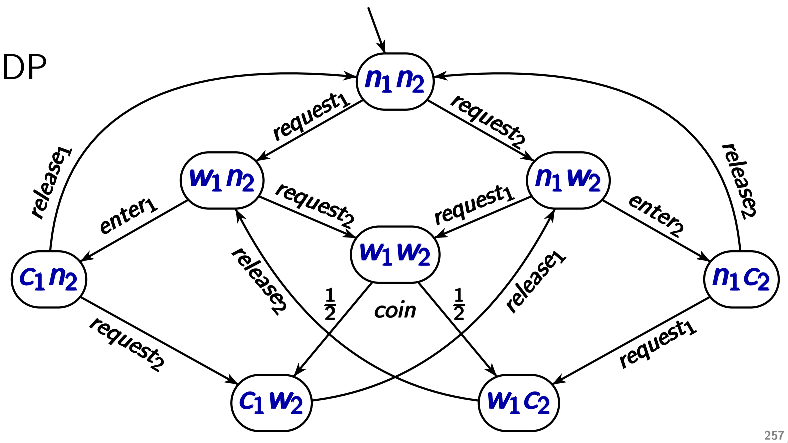
- competition of both processes are waiting

- resolved by a randomized arbiter who tosses a coin

Randomized mutual exclusion protocol

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

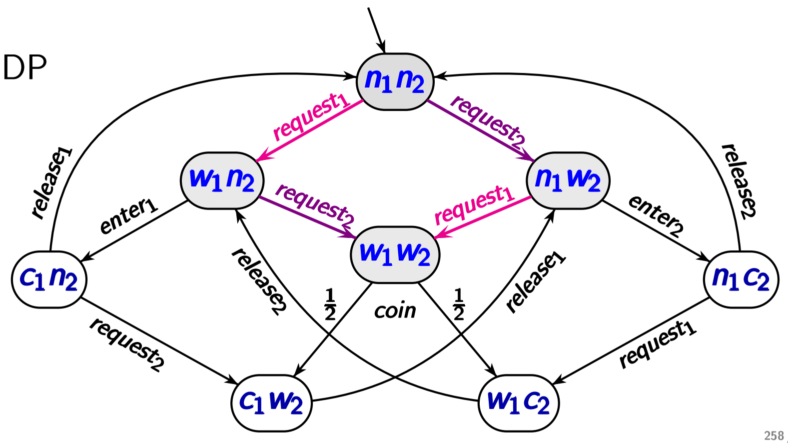
MDP



Randomized mutual exclusion protocol

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

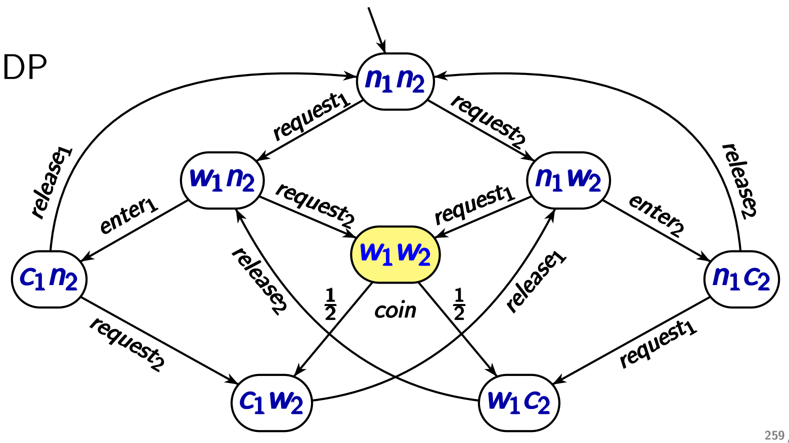
MDP



Randomized mutual exclusion protocol

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

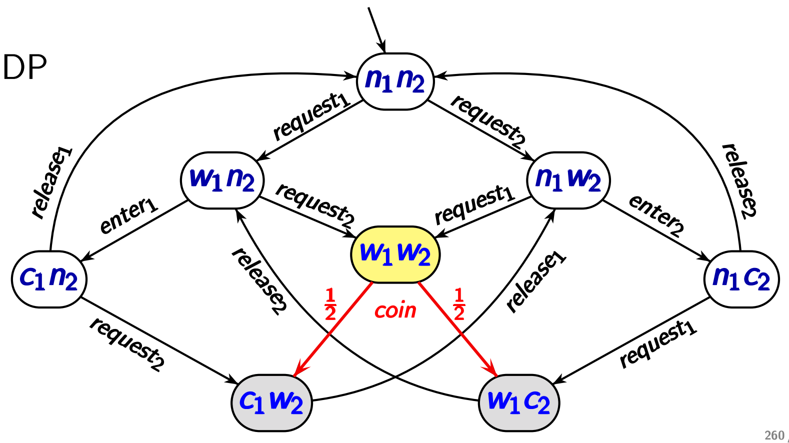
MDP



Randomized mutual exclusion protocol

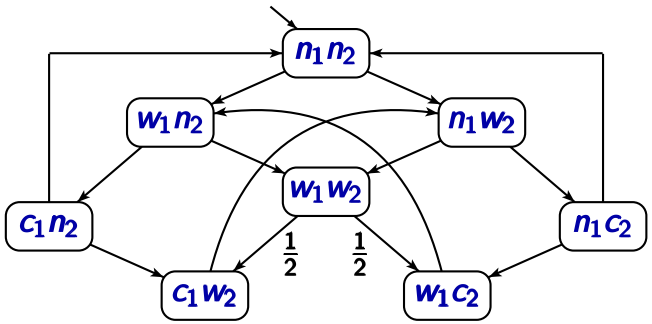
- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting

MDP



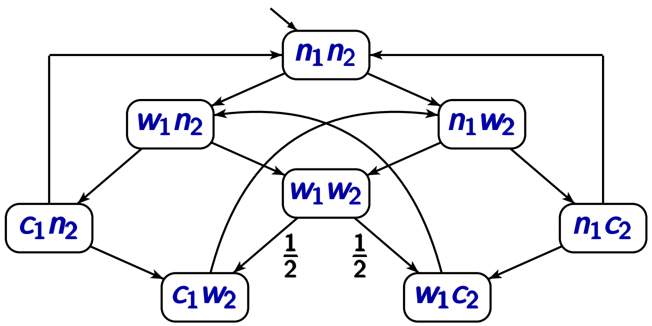
Properties of the randomized MUTEX

Properties of the randomized MUTEX



safety: the processes are never simultaneously in their critical section

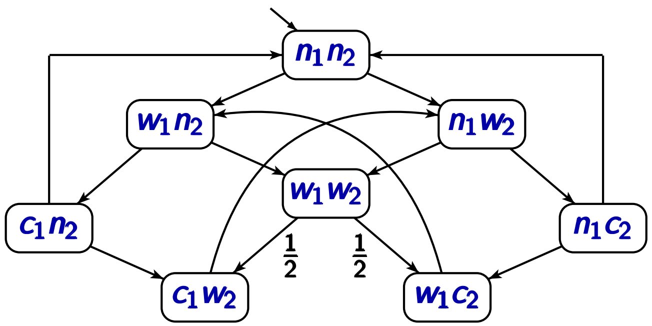
Properties of the randomized MUTEX



safety: the processes are never simultaneously in their critical section

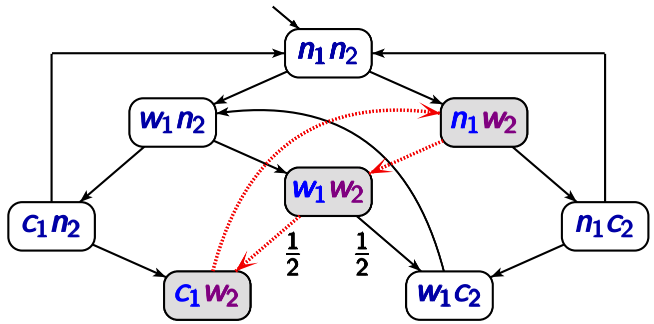
holds on all paths as state $\langle c_1, c_2 \rangle$ is unreachable

Properties of the randomized MUTEX



liveness: each waiting process will eventually enter its critical section

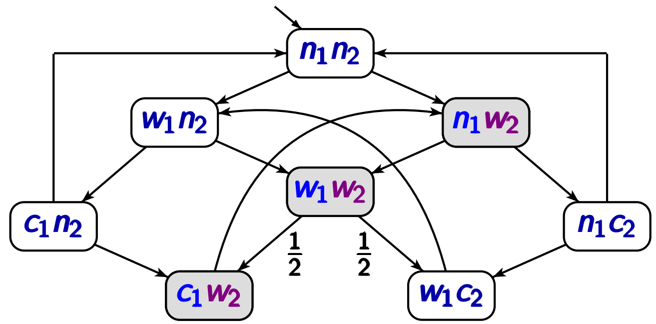
Properties of the randomized MUTEX



liveness: each waiting process will eventually enter its critical section

does not hold on all paths, but almost surely

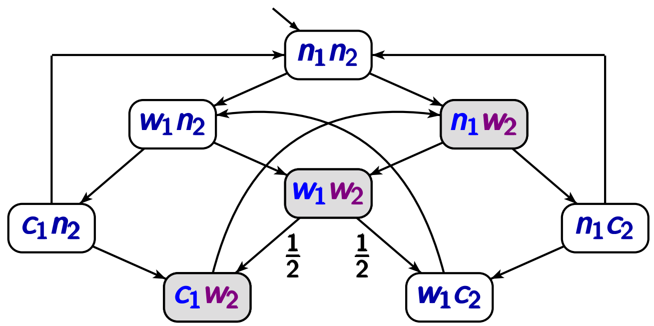
Properties of the randomized MUTEX



Suppose process **2** is waiting.

What is the probability that process **2** enters its critical section within the next **3** steps **?**

Properties of the randomized MUTEX

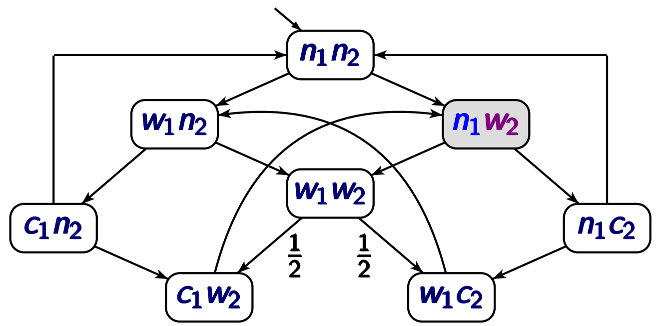


Suppose process **2** is waiting.

What is the probability that process **2** enters its critical section within the next **3** steps **?**

... depends ...

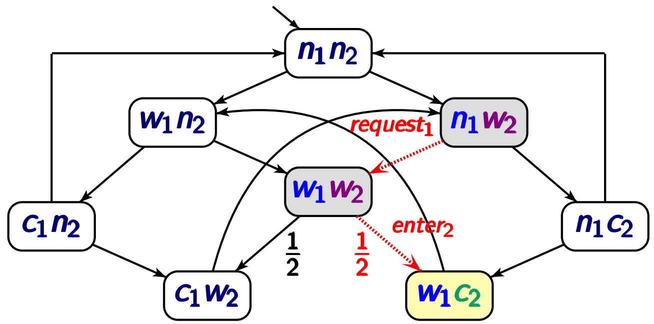
Randomized mutual exclusion protocol



Suppose the current state is $\langle n_1, w_2 \rangle$.

Randomized mutual exclusion protocol

PMC-74

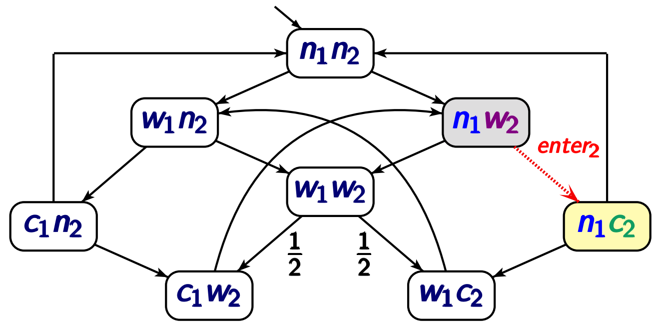


The probability that process **2** enters its critical section within the next **3** steps is:

$\frac{1}{2}$ if process **1** is scheduled in state $\langle n_1, w_2 \rangle$

Randomized mutual exclusion protocol

PMC-74



The probability that process **2** enters its critical section within the next **3** steps is:

$\frac{1}{2}$ if process **1** is scheduled in state $\langle n_1, w_2 \rangle$

1 if process **2** is scheduled in state $\langle n_1, w_2 \rangle$

Reasoning about probabilities in MDP

PMC-77

- requires resolving the nondeterminism by schedulers

Reasoning about probabilities in MDP

PMC-77

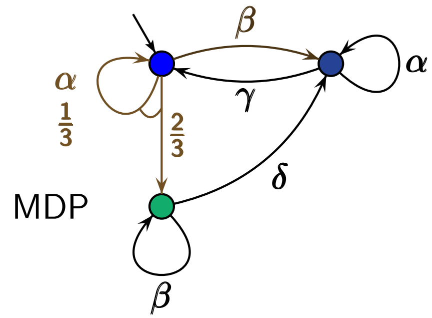
- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \ldots s_n)$ is enabled in state s_n

Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain

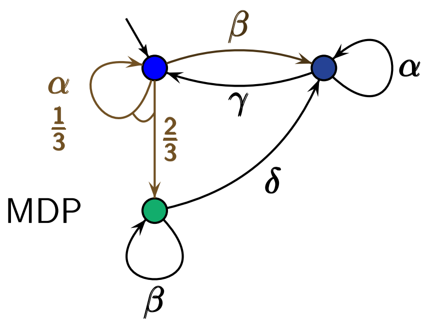
Reasoning about probabilities in MDP

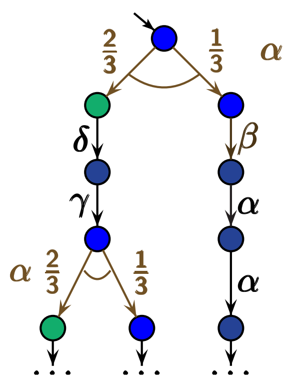
- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain



Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain





Reasoning about probabilities in MDP

- requires resolving the nondeterminism by schedulers
- a scheduler is a function $D : S^* \longrightarrow Act$ s.t. action $D(s_0 \dots s_n)$ is enabled in state s_n
- each scheduler induces an infinite Markov chain

↑

yields a notion of probability measure $\mathbf{Pr}^D$ on measurable sets of infinite paths

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking ⟵
3. fairness

PCTL* over MDPs [Bianco/de Alfaro'95]

- syntax of state and path formulas as for PCTL* over Markov chains
- probability operator $\mathbb{P}_I(\ldots)$ ranges over all schedulers

PCTL* over MDPs [Bianco/de Alfaro'95]

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \, \mathsf{U} \, \varphi_2$$

PCTL* over MDPs [Bianco/de Alfaro'95]

state formulas:

$$\Phi \ ::= \ true \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathbb{P}_I(\varphi)$$

path formulas:

$$\varphi \ ::= \ \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \, \mathsf{U} \, \varphi_2$$

given an MDP $\mathcal{M}$, define by structural induction:

- a satisfaction relation $\models$ for states s in $\mathcal{M}$ and **PCTL*** state formulas
- a satisfaction relation $\models$ for infinite paths π in $\mathcal{M}$ and **PCTL*** path formulas

Satisfaction relation for PCTL* state formulas

$s \models \mathit{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \Phi_1 \wedge \Phi_2$	iff	$s \models \Phi_1$ and $s \models \Phi_2$
$s \models \neg\Phi$	iff	$s \not\models \Phi$
$s \models \mathbb{P}_I(\varphi)$	iff	for all schedulers D:
		$Pr^D\{\pi \in Paths(s) : \pi \models \varphi\} \in I$

Satisfaction relation for PCTL* state formulas

$s \models \mathit{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \Phi_1 \wedge \Phi_2$	iff	$s \models \Phi_1$ and $s \models \Phi_2$
$s \models \neg\Phi$	iff	$s \not\models \Phi$
$s \models \mathbb{P}_I(\varphi)$	iff	for all schedulers D:
		$Pr^D\{\pi \in Paths(s) : \pi \models \varphi\} \in I$

probability measure in the Markov chain induced by D

Satisfaction relation for PCTL* state formulas

$s \models \mathit{true}$		
$s \models a$	iff	$a \in L(s)$
$s \models \Phi_1 \wedge \Phi_2$	iff	$s \models \Phi_1$ and $s \models \Phi_2$
$s \models \neg\Phi$	iff	$s \not\models \Phi$
$s \models \mathbb{P}_I(\varphi)$	iff	for all schedulers D:
		$Pr^D\{\pi \in Paths(s) : \pi \models \varphi\} \in I$

probability measure in the Markov chain induced by D

semantics of path formulas as for Markov chains

PCTL* model checking for MDP

PCTL* model checking for MDP

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
PCTL* state formula Φ

task: check whether $s_0 \models \Phi$

PCTL* model checking for MDP

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
PCTL* state formula Φ

task: check whether $s_0 \models \Phi$

main procedure as for PCTL* over Markov chains:

recursively compute the satisfaction sets

$$Sat(\Psi) \;=\; \{\, s \in S : s \models \Psi \,\}$$

for all sub-state formulas Ψ of Φ

PCTL* model checking for MDP

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
PCTL* state formula Φ

task: check whether $s_0 \models \Phi$

main procedure as for PCTL* over Markov chains:

recursively compute the satisfaction sets

$$Sat(\Psi) \;=\; \{\, s \in S : s \models \Psi \,\}$$

for all sub-state formulas Ψ of Φ

treatment of the propositional logic fragment: $\surd$

Treatment of probability operator

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

- compute the maximal probabilities for φ

  $$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \sup_{D} \Pr^{D}\{\, \pi \in Paths(s) : \pi \models \varphi \}$$

  for all states s

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

- compute the maximal probabilities for φ

  $$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \max_{D} \Pr^{D}\{\, \pi \in Paths(s) : \pi \models \varphi \}$$

  for all states s

there exists optimal
finite-memory schedulers

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

- compute the maximal probabilities for φ

  $$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \;=\; \max_{D} \Pr^{D}\{\, \pi \in Paths(s) : \pi \models \varphi \}$$

  for all states s
- return $\{\, s \in S : \Pr^{\mathcal{M}}_{\max}(s, \varphi) \leqslant p \,\}$

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

- compute the maximal probabilities for φ

$$\Pr^{\mathcal{M}}_{\max}(s,\varphi) \;=\; \max_{D} \Pr^{D}\{\,\pi \in Paths(s) \,:\, \pi \models \varphi\}$$

  for all states s

- return $\{\, s \in S \,:\, \Pr^{\mathcal{M}}_{\max}(s,\varphi) \leqslant p \,\}$

lower probability bounds $\mathbb{P}_{\geqslant p}(\varphi)$ or $\mathbb{P}_{>p}(\varphi)$

analogous, but minimal probabilities for φ

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

compute the maximal probabilities for φ

$$\Pr^{\mathcal{M}}_{\max}(s,\varphi) \;=\; \max_{D} \Pr^{D}\{\,\pi \in Paths(s) \,:\, \pi \models \varphi\}$$

special case: $\varphi = \Diamond\Psi$

↑ reachability condition

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

compute the maximal probabilities for φ

$$\Pr^{\mathcal{M}}_{\max}(s,\varphi) \;=\; \max_{D} \Pr^{D}\{\,\pi \in Paths(s) \,:\, \pi \models \varphi\}$$

special case: $\varphi = \Diamond\Psi$

compute $\Pr^{\mathcal{M}}_{\max}(s,\Diamond\Psi)$ by solving a linear program

↑ maximal reachability probabilities

Treatment of probability operator

upper probability bounds $\mathbb{P}_{\leqslant p}(\varphi)$ or $\mathbb{P}_{<p}(\varphi)$

compute the maximal probabilities for φ

$$\Pr^{\mathcal{M}}_{\max}(s,\varphi) \;=\; \max_{D} \Pr^{D}\{\,\pi \in Paths(s) \,:\, \pi \models \varphi\}$$

special case: $\varphi = \Diamond\Psi$

compute $\Pr^{\mathcal{M}}_{\max}(s,\Diamond\Psi)$ by solving a linear program

general case:

via determininistic automaton $\mathcal{A}$ for φ and maximal reachability probabilities in $\mathcal{M} \times \mathcal{A}$

Maximal reachability probabilities

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S
set $T \subseteq S$ of goal states

task: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \Diamond T) = \max_D \Pr^D(s, \Diamond T)$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S
set $T \subseteq S$ of goal states

task: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \Diamond T) = \max_D \Pr^D(s, \Diamond T)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]$ of the equation system:

$$\begin{aligned} x_s &= 1 && \text{if } s \in T \\ x_s &= \max_{\alpha} \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} && \text{if } s \notin T \end{aligned}$$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S
set $T \subseteq S$ of goal states

task: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \Diamond T) = \max_D \Pr^D(s, \Diamond T)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]$ of the equation system:

$$\begin{aligned} x_s &= 1 && \text{if } s \in T \\ x_s &= \max_{\alpha} \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} && \text{if } s \notin T \end{aligned}$$

α ranges over all actions in $Act(s)$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S
set $T \subseteq S$ of goal states

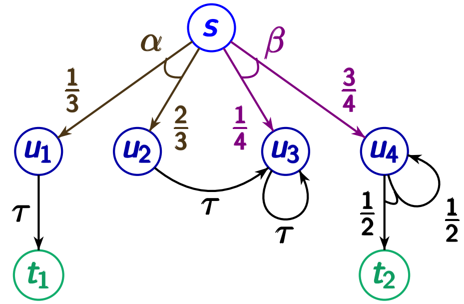
task: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \Diamond T) = \max_D \Pr^D(s, \Diamond T)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]$ of the equation system:

$$x_s = 1 \quad \text{if } s \in T$$

$$x_s = \max_{\alpha} \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \quad \text{if } s \notin T$$

α ranges over all actions in $Act(s)$

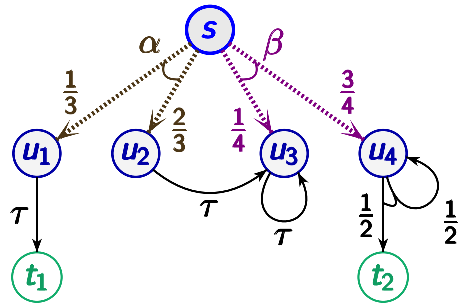


$T = \{t_1, t_2\}$

The vector $(x_s)_{s \in S}$ where $x_s = \Pr^{\mathcal{M}}_{\max}(s, \Diamond T)$ is the least solution in $[0,1]$ of

$$x_s = 1 \quad \text{if } s \in T$$

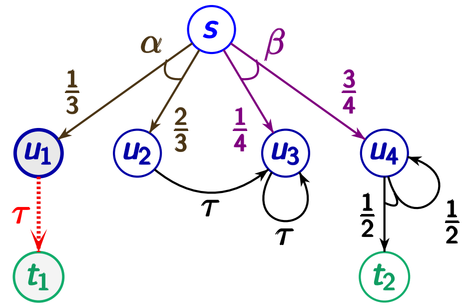
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \quad \text{if } s \notin T$$



$T = \{t_1, t_2\}$

$$x_s = \max\{ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2}, \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4} \}$$

action α action β

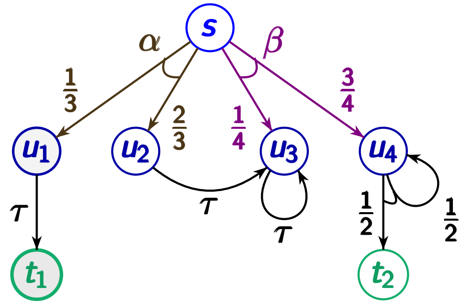


$T = \{t_1, t_2\}$

$$x_s = \max\{ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2}, \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4} \}$$

$$x_{u_1} = x_{t_1}$$

unique successor of u_1

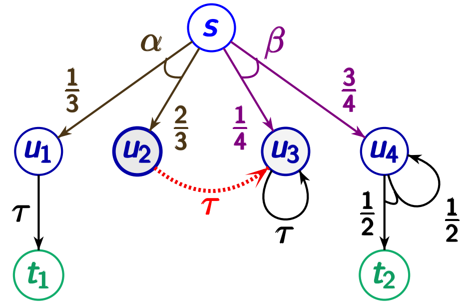


$T = \{t_1, t_2\}$

$$x_s = \max\{\ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4}\ \}$$

$$x_{u_1} = x_{t_1} = 1$$

↑ goal state $t_1 \in T$



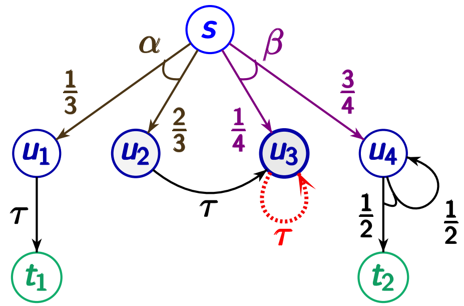
$T = \{t_1, t_2\}$

$$x_s = \max\{\ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4}\ \}$$

$$x_{u_1} = x_{t_1} = 1$$

$$x_{u_2} = x_{u_3}$$

↑ unique successor of state u_2



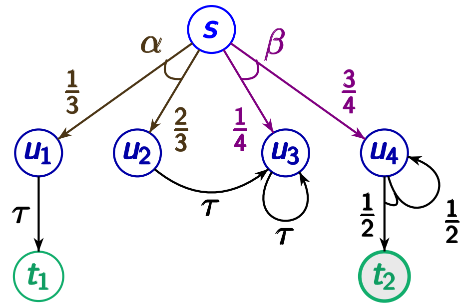
$T = \{t_1, t_2\}$

$$x_s = \max\{\ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4}\ \}$$

$$x_{u_1} = x_{t_1} = 1$$

$$x_{u_2} = x_{u_3} = 0$$ least solution of $x_{u_3} = x_{u_3}$

↑ unique successor of state u_2



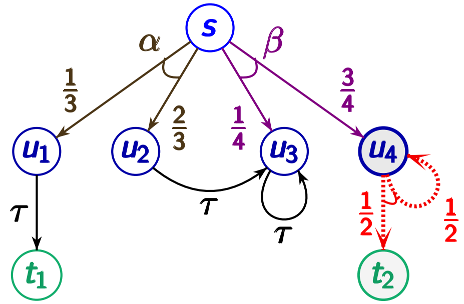
$T = \{t_1, t_2\}$

$$x_s = \max\{\ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4}\ \}$$

$$x_{u_1} = x_{t_1} = 1$$

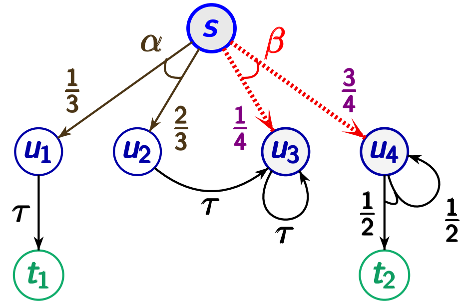
$$x_{u_2} = x_{u_3} = 0$$ least solution of $x_{u_3} = x_{u_3}$

$$x_{t_2} = 1$$



$T = \{t_1, t_2\}$

$$x_s = \max\{ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4} \}$$
$$x_{u_1} = x_{t_1} = 1$$
$$x_{u_2} = x_{u_3} = 0 \qquad \text{least solution of } x_{u_3} = x_{u_3}$$
$$x_{t_2} = 1$$
$$x_{u_4} = \tfrac{1}{2}x_{u_4} + \tfrac{1}{2}x_{t_2} = \tfrac{1}{2}x_{u_4} + \tfrac{1}{2} = 1$$



$T = \{t_1, t_2\}$

$$x_s = \max\{ \tfrac{1}{3}x_{u_1} + \tfrac{2}{3}x_{u_2},\ \tfrac{1}{4}x_{u_3} + \tfrac{3}{4}x_{u_4} \} = \tfrac{3}{4}$$
$$x_{u_1} = x_{t_1} = 1$$
$$x_{u_2} = x_{u_3} = 0 \qquad \text{least solution of } x_{u_3} = x_{u_3}$$
$$x_{t_2} = 1$$
$$x_{u_4} = \tfrac{1}{2}x_{u_4} + \tfrac{1}{2}x_{t_2} = \tfrac{1}{2}x_{u_4} + \tfrac{1}{2} = 1$$

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \lozenge T)$ is the least solution in $[0,1]$ of

$$x_s = 1 \quad \text{if } s \in T$$
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} \qquad \text{if } s \notin T$$

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \lozenge T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \quad \text{if } s \in T$$
$$x_s = 0 \quad \text{if } T \text{ is not reachable from } s$$
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'}$$

if $s \notin T$ and T is reachable from s

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \text{ if } s \in S^1$$
$$x_s = 0 \text{ if } s \in S^0$$
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \qquad \text{if } s \in S^?$$

$$S^1 = \{ s \in S : x_s = 1 \}$$
$$S^0 = \{ s \in S : x_s = 0 \}$$
$$S^? = \{ s \in S : 0 < x_s < 1 \} = S \setminus (S^1 \cup S^0)$$

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \text{ if } s \in S^1$$
$$x_s = 0 \text{ if } s \in S^0$$
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \qquad \text{if } s \in S^?$$

$$S^1 = \{ s \in S : x_s = 1 \}$$
$$S^0 = \{ s \in S : s \not\models \exists \Diamond T \}$$
$$S^? = \{ s \in S : 0 < x_s < 1 \} = S \setminus (S^1 \cup S^0)$$

graph algorithms

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \text{ if } s \in S^1$$
$$x_s = 0 \text{ if } s \in S^0$$
$$x_s = \max_{\alpha} \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \qquad \text{if } s \in S^?$$

↑ can be rephrased as a linear program

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \text{ if } s \in S^1$$
$$x_s = 0 \text{ if } s \in S^0$$
$$x_s \geqslant \sum_{s' \in S^?} P(s,\alpha,s') \cdot x_{s'} + P(s,\alpha,S^1)$$

if $s \in S^?$ and $\alpha \in Act(s)$

Maximal reachability probabilities

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \quad \text{if} \quad s \in S^1$$

$$x_s = 0 \quad \text{if} \quad s \in S^0$$

$$x_s \geqslant \sum_{s' \in S^?} P(s, \alpha, s') \cdot x_{s'} + P(s, \alpha, S^1)$$

$$\text{if } s \in S^? \text{ and } \alpha \in Act(s)$$

where $\sum_{s \in S^?} x_s$ is minimal

Maximal reachability probabilities

PMC-87

The vector $(x_s)_{s \in S}$ where $x_s = \Pr_{\max}(s, \Diamond T)$ is the unique solution in $[0,1]$ of

$$x_s = 1 \quad \text{if} \quad s \in S^1$$

$$x_s = 0 \quad \text{if} \quad s \in S^0$$

$$x_s \geqslant \sum_{s' \in S^?} P(s, \alpha, s') \cdot x_{s'} + P(s, \alpha, S^1)$$

$$\text{if } s \in S^? \text{ and } \alpha \in Act(s)$$

where $\sum_{s \in S^?} x_s$ is minimal

linear program

Maximal probabilities for limit properties

Maximal probabilities for limit properties

PMC-89

given: MDP $\mathcal{M} = (S, P, \ldots)$
prefix-independent limit property E for paths

task: compute $\Pr_{\max}(s, E)$

↑

$$\max_D \Pr^D \{ \pi \in Paths(s) : \pi \in E \}$$

Maximal probabilities for limit properties

given: MDP $\mathcal{M} = (S, P, \ldots)$
prefix-independent limit property E for paths

task: compute $\mathbf{Pr}_{\mathbf{max}}(s, E)$

i.e., there exists subsets $T_1, \ldots, T_k$ of S s.t.
for all paths π in $\mathcal{M}$:

$$\pi \models E \quad \text{iff} \quad \exists i \in \{1, \ldots, k\}.\ \inf(\pi) = T_i$$

where $\inf(s_0\, s_1\, s_2 \ldots) = \{t \in S : \overset{\infty}{\exists}\, i \geqslant 0.\ s_i = t\}$

Maximal probabilities for limit properties

given: MDP $\mathcal{M} = (S, P, \ldots)$
prefix-independent limit property E for paths

task: compute $\mathbf{Pr}_{\mathbf{max}}(s, E)$

i.e., there exists subsets $T_1, \ldots, T_k$ of S s.t.
for all paths π in $\mathcal{M}$:

$$\pi \models E \quad \text{iff} \quad \exists i \in \{1, \ldots, k\}.\ \inf(\pi) = T_i$$

... relies on an analysis of end components

End component [de Alfaro'96]

End component [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, AP, L)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP

End component [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, AP, L)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair (T, A) where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) ...

(2) ...

(3) ...

End component [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, AP, L)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair (T, A) where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

$\varnothing \neq A(t) \subseteq Act(t)$ for all $t \in T$

(2) ...

(3) ...

End component [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, AP, L)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair (T, A) where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

$\varnothing \neq A(t) \subseteq Act(t)$ for all $t \in T$

(2) closed under probabilistic branching:

$\forall t \in T \, \forall \alpha \in A(t). \, (P(t, \alpha, u) > 0 \Longrightarrow u \in T)$

(3) ...

End component [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, AP, L)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair (T, A) where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

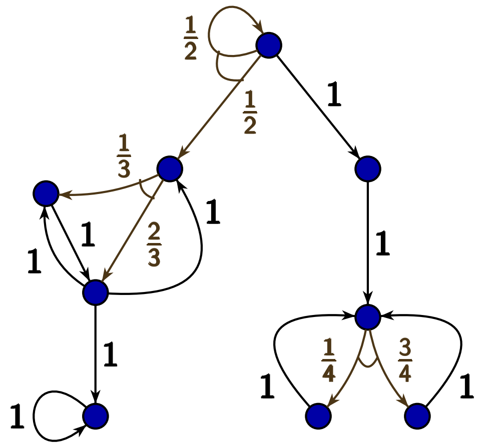
$\varnothing \neq A(t) \subseteq Act(t)$ for all $t \in T$

(2) closed under probabilistic branching:

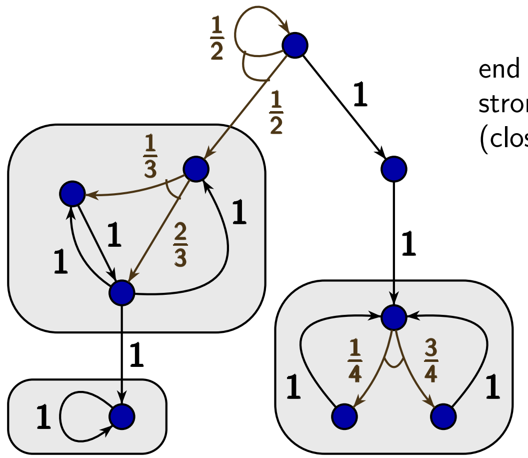
$\forall t \in T \, \forall \alpha \in A(t). \, (P(t, \alpha, u) > 0 \Longrightarrow u \in T)$

(3) the underlying graph is strongly connected

Example: end components

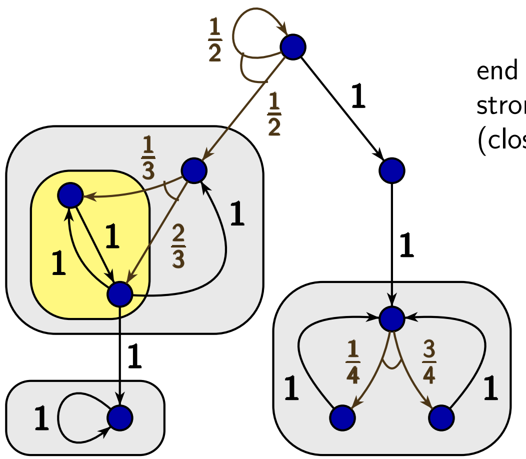


Example: end components



end component:
strongly connected sub-MDP
(closed under prob. branching)

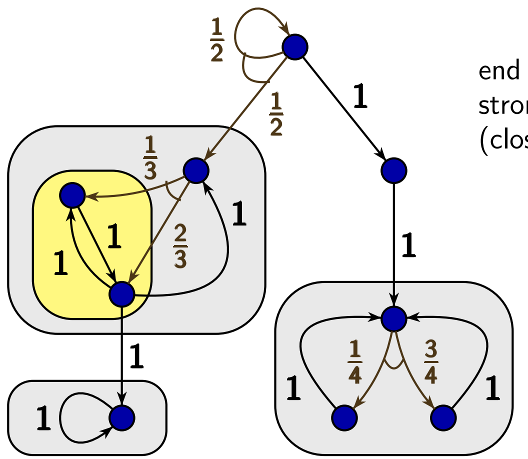
Example: end components



end component:
strongly connected sub-MDP
(closed under prob. branching)

Long-run behavior of MDPs

For all schedulers ***D***, almost surely an end component will be reached and all its states visited infinitely often.



end component:
strongly connected sub-MDP
(closed under prob. branching)

Long-run behavior of MDPs

For all schedulers D, almost surely an end component will be reached and all its states visited infinitely often.

i.e., for all schedulers D and states s:

$$Pr^D\left\{ \pi \in Paths(s) : \begin{array}{l} inf(\pi) \text{ constitutes} \\ \text{an end component} \end{array} \right\} = 1$$

Long-run behavior of MDPs

For all schedulers D, almost surely an end component will be reached and all its states visited infinitely often.

i.e., for all schedulers D and states s:

$$Pr^D\left\{ \pi \in Paths(s) : \begin{array}{l} inf(\pi) \text{ constitutes} \\ \text{an end component} \end{array} \right\} = 1$$

Let E be a limit property and $T_1, \ldots, T_k \subseteq S$ s.t.

$$\pi \models E \quad \text{iff} \quad \exists i \geqslant 0.\ inf(\pi) = T_i$$

Long-run behavior of MDPs

For all schedulers D, almost surely an end component will be reached and all its states visited infinitely often.

i.e., for all schedulers D and states s:

$$Pr^D\left\{ \pi \in Paths(s) : \begin{array}{l} inf(\pi) \text{ constitutes} \\ \text{an end component} \end{array} \right\} = 1$$

Let E be a limit property and $T_1, \ldots, T_k \subseteq S$ s.t.

$$\pi \models E \quad \text{iff} \quad \exists i \geqslant 0.\ inf(\pi) = T_i$$

Then: $Pr_{max}(s, E) = Pr_{max}(s, \Diamond A)$ where

$$A = \bigcup \{ T_i : T_i \text{ constitutes an end component} \}$$

PCTL* model checking for MDP

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$
PCTL* state formula $\mathbb{P}_{\leqslant p}(\varphi)$

task: compute $Sat(\mathbb{P}_{\leqslant p}(\varphi))$

PCTL* model checking for MDP

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$
PCTL* state formula $\mathbb{P}_{\leqslant p}(\varphi)$

task: compute $Sat(\mathbb{P}_{\leqslant p}(\varphi))$

method: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \varphi)$ via a reduction to the probabilistic reachability problem

PCTL* model checking for MDP

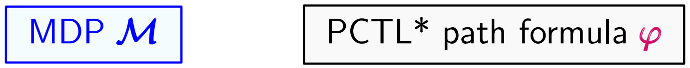
given: MDP $\mathcal{M} = (S, Act, P, \ldots)$
PCTL* state formula $\mathbb{P}_{\leqslant p}(\varphi)$

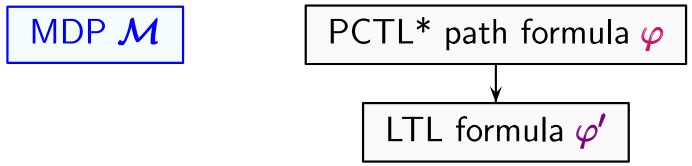
task: compute $Sat(\mathbb{P}_{\leqslant p}(\varphi))$

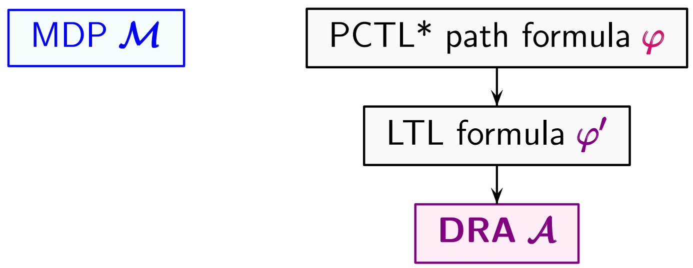
method: compute $x_s = \Pr^{\mathcal{M}}_{\max}(s, \varphi)$ via a reduction to the probabilistic reachability problem

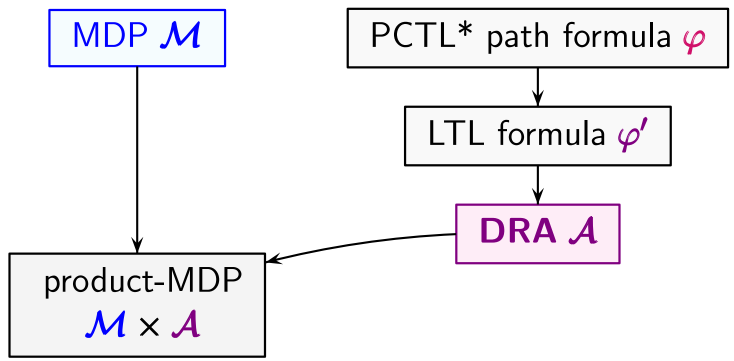
using DRA $\mathcal{A}$ for φ and linear program for $\mathcal{M} \times \mathcal{A}$

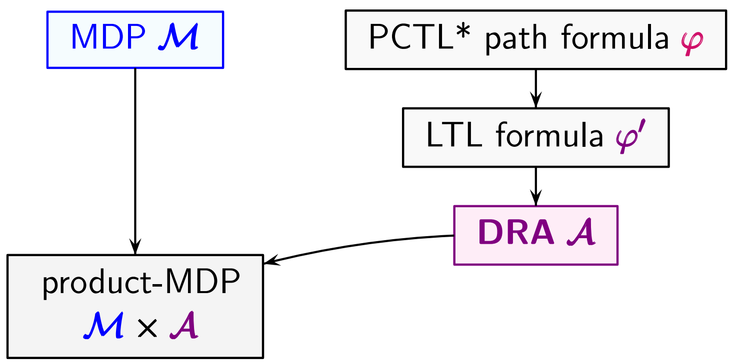
DRA: deterministic Rabin automaton



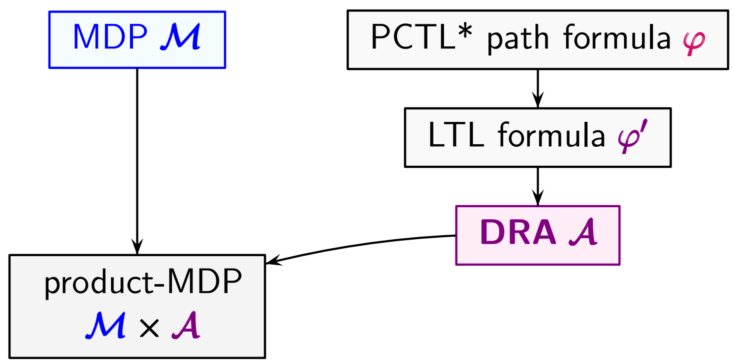






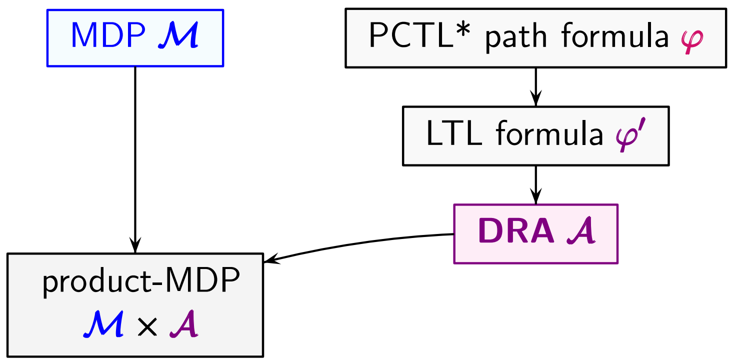


$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \quad = \quad \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\Big(\langle s, init_s\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\Big)$$



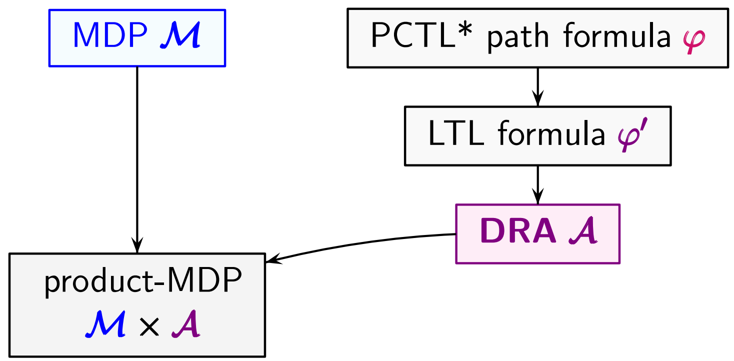
$$\Pr^{\mathcal{M}}_{\max}(s, \varphi) \quad = \quad \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\Big(\langle s, init_s\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\Big)$$

initial state in the product, if $\mathcal{M}$ starts in s,
i.e., $init_s = \delta(q_0, L(s))$



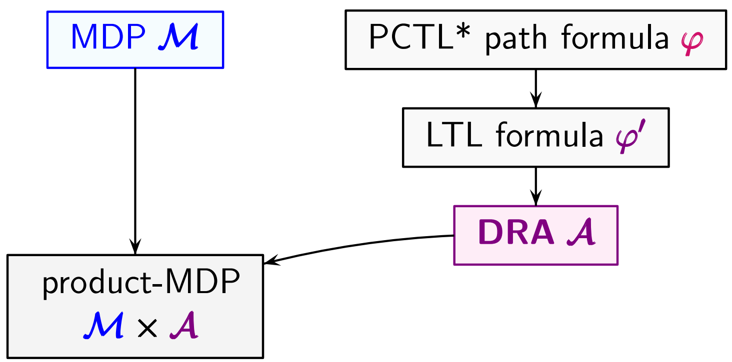
$$\Pr^{\mathcal{M}}_{\max}(s,\varphi) = \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\big)$$

acceptance condition of $\mathcal{A}$



$$\begin{aligned}\Pr^{\mathcal{M}}_{\max}(s,\varphi) &= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\big)\\ &= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \Diamond accEC\big)\end{aligned}$$

union of all accepting end components



$$\begin{aligned}\Pr^{\mathcal{M}}_{\max}(s,\varphi) &= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\big)\\ &= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \Diamond accEC\big)\\ &= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s, init_s\rangle, \Diamond accMEC\big)\end{aligned}$$

Lower probability bounds

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

PCTL* star formula $\mathbb{P}_{\geqslant p}(\varphi)$

task: compute $Sat\big(\mathbb{P}_{\geqslant p}(\varphi)\big)$

Lower probability bounds

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

PCTL* star formula $\mathbb{P}_{\geqslant p}(\varphi)$

task: compute $Sat(\mathbb{P}_{\geqslant p}(\varphi))$

use the duality of lower and upper probability bounds

for each scheduler D and state s:

$$\Pr^D(s, \varphi) \;=\; 1 - \Pr^D(s, \neg\varphi)$$

Lower probability bounds

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

PCTL* star formula $\mathbb{P}_{\geqslant p}(\varphi)$

task: compute $Sat(\mathbb{P}_{\geqslant p}(\varphi))$

use the duality of lower and upper probability bounds

for each scheduler D and state s:

$$\Pr^D(s, \varphi) \;=\; 1 - \Pr^D(s, \neg\varphi)$$

For each state s and PCTL* path formula φ:

$$\Pr^{\mathcal{M}}_{\min}(s, \varphi) \;=\; 1 - \Pr^{\mathcal{M}}_{\max}(s, \neg\varphi)$$

Complexity of PCTL/PCTL* model checking

Complexity of PCTL/PCTL* model checking

	PCTL	PCTL*
Markov chain		
Markov decision process		

Complexity of PCTL/PCTL* model checking

	PCTL	PCTL*
Markov chain	graph algorithms + linear equation systems	
Markov decision process	graph algorithms + linear program	

Complexity of PCTL/PCTL* model checking

	PCTL	PCTL*
Markov chain	graph algorithms + linear equation systems	
	PTIME	*PSPACE*-complete [Vardi/Wolper'86]
Markov decision process	graph algorithms + linear program	

Complexity of PCTL/PCTL* model checking

	PCTL	PCTL*
Markov chain	graph algorithms + linear equation systems	
	PTIME	*PSPACE*-complete [Vardi/Wolper'86]
Markov decision process	graph algorithms + linear program	
	PTIME	2*EXP*-complete [Courcoubetis/Yannakakis'88]

Complexity of PCTL/PCTL* model checking

	PCTL	PCTL*
Markov chain	graph algorithms + linear equation systems	
	PTIME	*PSPACE*-complete [Vardi/Wolper'86]
Markov decision process	graph algorithms + linear program	
	PTIME	2*EXP*-complete [Courcoubetis/Yannakakis'88]

tools: PRISM (Oxford), MRMC (Aachen), ...

Discrete-time Markovian models

Part 1: discrete-time Markov chains (DTMC)

1. basic definitions
2. probabilistic computation tree logic (PCTL/PCTL*)
3. expected rewards

Part 2: Markov decision processes (MDP)

1. basic definitions
2. PCTL/PCTL* model checking
3. fairness ⟵

Why MDP with fairness ?

Markov decision processes

- extend Markov chains by nondeterminism
- modelling asynchronous distributed systems by interleaving

Markov decision processes

- extend Markov chains by nondeterminism
- modelling asynchronous distributed systems by interleaving

↑

verification of liveness properties
(qualitative or quantitative)
often requires fairness assumptions

Markov decision processes

- extend Markov chains by nondeterminism
- modelling asynchronous distributed systems by interleaving

↑

> verification of liveness properties
> (qualitative or quantitative)
> often requires fairness assumptions

e.g., strong process fairness:

$\Box\Diamond$ process P is enabled $\longrightarrow$ $\Box\Diamond$ actions of process P are taken

Markov decision processes

PMC-110

- extend Markov chains by nondeterminism
- modelling asynchronous distributed systems by interleaving

↑

> verification of liveness properties
> (qualitative or quantitative)
> often requires fairness assumptions

general case: fairness assumptions impose restrictions on the resolution of nondeterminism to rule out unrealistic behaviors

MDP with fairness

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

A fairness assumption $\mathcal{F}$ for $\mathcal{M}$ is a conjunction of limit properties of the form:

$\Box\Diamond Y$	unconditional fairness
$\Box\Diamond X \rightarrow \Box\Diamond Y$	strong fairness
$\Diamond\Box X \rightarrow \Box\Diamond Y$	weak fairness

where $X, Y \subseteq S$

MDP with fairness

PMC-111

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

A fairness assumption $\mathcal{F}$ for $\mathcal{M}$ is a conjunction of limit properties of the form:

$\Box\Diamond Y$	unconditional fairness
$\Box\Diamond X \rightarrow \Box\Diamond Y$	strong fairness
$\Diamond\Box X \rightarrow \Box\Diamond Y$	weak fairness

where $X, Y \subseteq S$

here: just state-based fairness conditions

action-based fairness conditions can be encoded, e.g.,

$\Box\Diamond enabled(A) \rightarrow \Box\Diamond taken(A)$ where $A \subseteq Act$

MDP with fairness

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$

A fairness assumption $\mathcal{F}$ for $\mathcal{M}$ is a conjunction of limit properties of the form:

$\Box\Diamond Y$	unconditional fairness
$\Box\Diamond X \to \Box\Diamond Y$	strong fairness
$\Diamond\Box X \to \Box\Diamond Y$	weak fairness

where $X, Y \subseteq S$

Scheduler D for $\mathcal{M}$ is called $\mathcal{F}$-fair iff

$\Pr^D(s, \mathcal{F}) = 1$ for all (reachable) states s

Realizability of fairness assumptions

Realizability of fairness assumptions

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ and
a fairness assumption $\mathcal{F}$ for $\mathcal{M}$

Scheduler D is called $\mathcal{F}$-fair iff $\Pr^D(s, \mathcal{F}) = 1$
for all states $s \in S$.

Realizability of fairness assumptions

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ and
a fairness assumption $\mathcal{F}$ for $\mathcal{M}$

$\mathcal{F}$ is realizable iff there exists a $\mathcal{F}$-fair scheduler

Scheduler D is called $\mathcal{F}$-fair iff $\Pr^D(s, \mathcal{F}) = 1$
for all states $s \in S$.

Realizability of fairness assumptions

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ and
a fairness assumption $\mathcal{F}$ for $\mathcal{M}$

$\mathcal{F}$ is realizable iff there exists a $\mathcal{F}$-fair scheduler
iff $s \models \exists \Diamond FairMEC$ for all $s \in S$

Scheduler D is called $\mathcal{F}$-fair iff $\Pr^D(s, \mathcal{F}) = 1$ for all states $s \in S$.

Realizability of fairness assumptions

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ and
a fairness assumption $\mathcal{F}$ for $\mathcal{M}$
e.g., $\mathcal{F} = \Box\Diamond X \to \Box\Diamond Y$

$\mathcal{F}$ is realizable iff there exists a $\mathcal{F}$-fair scheduler
iff $s \models \exists \Diamond FairMEC$ for all $s \in S$

↑ union of all maximal end components that contain a sub-component T where $\mathcal{F}$ holds,
i.e., $X \cap T = \varnothing$ or $Y \cap T \neq \varnothing$

Realizability of fairness assumptions

given: MDP $\mathcal{M} = (S, Act, P, \ldots)$ and
a fairness assumption $\mathcal{F}$ for $\mathcal{M}$
e.g., $\mathcal{F} = \Box\Diamond X \to \Box\Diamond Y$

$\mathcal{F}$ is realizable iff there exists a $\mathcal{F}$-fair scheduler
iff $s \models \exists \Diamond FairMEC$ for all $s \in S$

poly-time algorithm for computing *FairMEC*:

... recursive computation of maximal end components in sub-MDPs

PCTL* with fairness

PCTL* with fairness

- syntax of state and path formulas as before

PCTL* with fairness

- syntax of state and path formulas as before
- semantics as for standard PCTL* over MDP, but:

$s \models_{\mathcal{F}} \mathbb{P}_{I}(\varphi)$ iff for all $\mathcal{F}$-fair schedulers D: $\Pr^{D}(s, \varphi) \in I$

PCTL* with fairness

- syntax of state and path formulas as before
- semantics as for standard PCTL* over MDP, but:

$s \models_{\mathcal{F}} \mathbb{P}_{I}(\varphi)$ iff for all $\mathcal{F}$-fair schedulers D: $\Pr^{D}(s, \varphi) \in I$

There are some simple cases: e.g., if $\mathcal{F}$ is realizable

$$s \models_{\mathcal{F}} \mathbb{P}_{\leqslant p}(\Diamond b) \quad \text{iff} \quad s \models \mathbb{P}_{\leqslant p}(\Diamond b)$$

PCTL* with fairness

- syntax of state and path formulas as before
- semantics as for standard PCTL* over MDP, but:

$s \models_{\mathcal{F}} \mathbb{P}_{I}(\varphi)$ iff for all $\mathcal{F}$-fair schedulers D: $\Pr^{D}(s, \varphi) \in I$

There are some simple cases: e.g., if $\mathcal{F}$ is realizable

$$s \models_{\mathcal{F}} \mathbb{P}_{\leqslant p}(\Diamond b) \quad \text{iff} \quad s \models \mathbb{P}_{\leqslant p}(\Diamond b)$$

but $s \models_{\mathcal{F}} \mathbb{P}_{\geqslant p}(\Diamond b)$ and $s \not\models \mathbb{P}_{\geqslant p}(\Diamond b)$ is possible

PCTL* with fairness

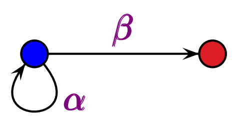
- syntax of state and path formulas as before
- semantics as for standard PCTL* over MDP, but:

$s \models_{\mathcal{F}} \mathbb{P}_{I}(\varphi)$ iff for all $\mathcal{F}$-fair schedulers D: $Pr^{D}(s, \varphi) \in I$

There are some simple cases: e.g., if $\mathcal{F}$ is realizable

$s \models_{\mathcal{F}} \mathbb{P}_{\leqslant p}(\Diamond b)$ iff $s \models \mathbb{P}_{\leqslant p}(\Diamond b)$

but $s \models_{\mathcal{F}} \mathbb{P}_{\geqslant p}(\Diamond b)$ and $s \not\models \mathbb{P}_{\geqslant p}(\Diamond b)$ is possible



$\mathcal{F} = \Box\Diamond blue \rightarrow \Box\Diamond red$

PCTL* with fairness

- syntax of state and path formulas as before
- semantics as for standard PCTL* over MDP, but:

$s \models_{\mathcal{F}} \mathbb{P}_{I}(\varphi)$ iff for all $\mathcal{F}$-fair schedulers D: $Pr^{D}(s, \varphi) \in I$

There are some simple cases: e.g., if $\mathcal{F}$ is realizable

$s \models_{\mathcal{F}} \mathbb{P}_{\leqslant p}(\Diamond b)$ iff $s \models \mathbb{P}_{\leqslant p}(\Diamond b)$

$s \models_{\mathcal{F}} \mathbb{P}_{\geqslant p}(\Diamond b)$ iff $s \models \mathbb{P}_{\leqslant 1-p}(\neg b \, \mathsf{U} \, ECFair(\neg b))$

↑ union of all fair end components without any b-state

PCTL* model checking for MDP with fairness

PCTL* model checking for MDP with fairness

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
realizable fairness assumption $\mathcal{F}$
PCTL* state formula Φ

task: check whether $s_0 \models_{\mathcal{F}} \Phi$

PCTL* model checking for MDP with fairness

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
realizable fairness assumption $\mathcal{F}$
PCTL* state formula Φ

task: check whether $s_0 \models_{\mathcal{F}} \Phi$

main procedure as for standard PCTL*:

recursively compute the satisfaction sets

$$Sat_{\mathcal{F}}(\Psi) = \{ s \in S : s \models_{\mathcal{F}} \Psi \}$$

for all sub-state formulas Ψ of Φ

PCTL* model checking for MDP with fairness

given: MDP $\mathcal{M} = (S, Act, P, AP, L, s_0)$
realizable fairness assumption $\mathcal{F}$
PCTL* state formula Φ

task: check whether $s_0 \models_{\mathcal{F}} \Phi$

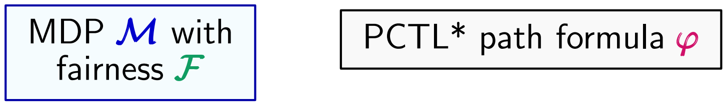
main procedure as for standard PCTL*:

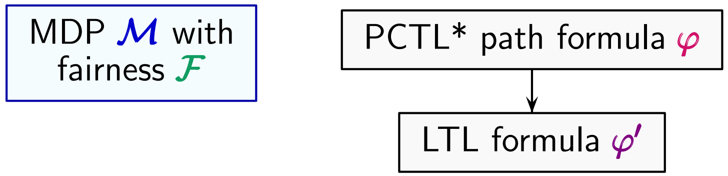
recursively compute the satisfaction sets

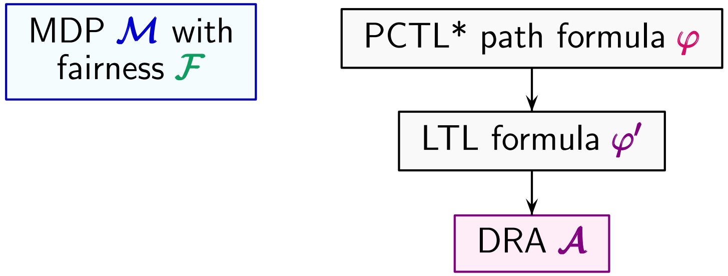
$$Sat_{\mathcal{F}}(\Psi) = \{ s \in S : s \models_{\mathcal{F}} \Psi \}$$

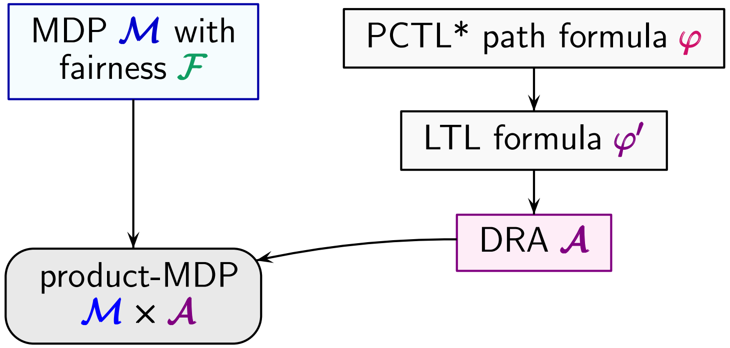
for all sub-state formulas Ψ of Φ

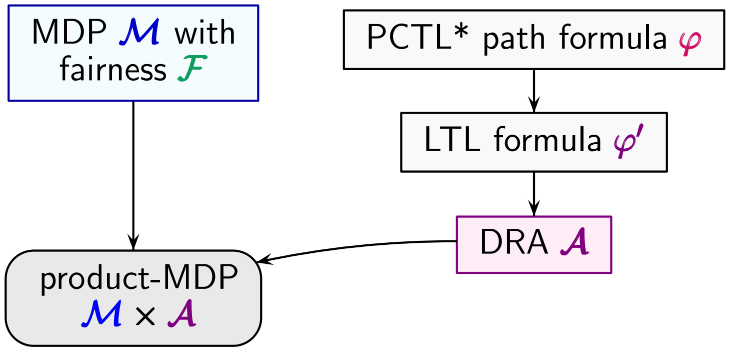
treatment of the propositional logic fragment: √





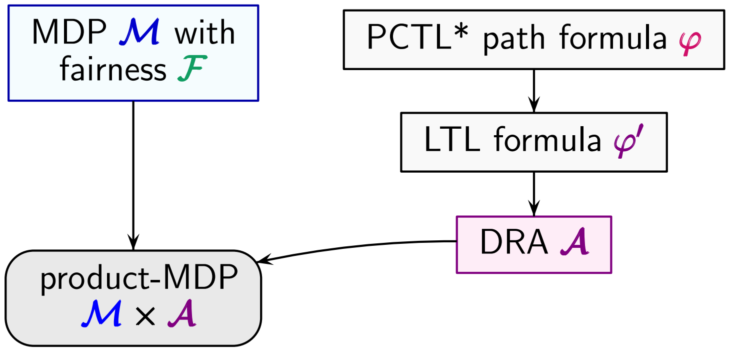






$$\max_{D \text{ fair}} \Pr^D(s, \varphi) = \max_{E \text{ fair}} \Pr^E\Big(\langle s, \dots\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\Big)$$

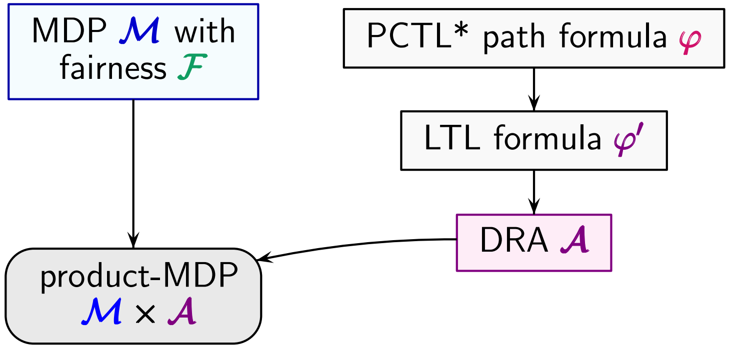
acceptance condition of $\mathcal{A}$



$$\max_{D \text{ fair}} \Pr^D(s, \varphi) = \max_{E \text{ fair}} \Pr^E\Big(\langle s, \dots\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\Big)$$

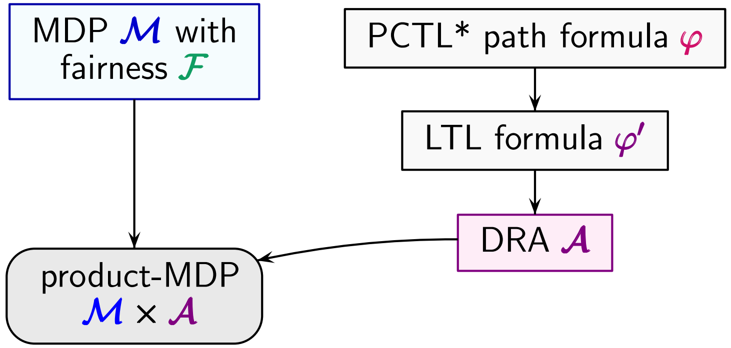
D ranges over all fair schedulers for $\mathcal{M}$

E ranges over all fair schedulers for $\mathcal{M} \times \mathcal{A}$



$$\max_{D \text{ fair}} \Pr^D(s,\varphi) = \max_{E \text{ fair}} \Pr^E\big(\langle s,\ldots\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\big)$$
$$= \max_{E \text{ fair}} \Pr^E\big(\langle s,\ldots\rangle, \Diamond \mathit{FairAccEC}\big)$$

union of all maximal end components that contain a fair sub-component C s.t. $\exists i.\ C \cap L_i = \varnothing$ and $C \cap U_i \neq \varnothing$



$$\max_{D \text{ fair}} \Pr^D(s,\varphi) = \max_{E \text{ fair}} \Pr^E\big(\langle s,\ldots\rangle, \bigvee_i (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)\big)$$
$$= \max_{E \text{ fair}} \Pr^E\big(\langle s,\ldots\rangle, \Diamond \mathit{FairAccEC}\big)$$
$$= \Pr^{\mathcal{M}\times\mathcal{A}}_{\max}\big(\langle s,\ldots\rangle, \Diamond \mathit{FairAccEC}\big)$$

Conclusion

Conclusion

- model checking for systems with discrete probabilities
  - ∗ techniques for verifying non-probabilistic systems
    graph algorithms, automata, ...
  - ∗ numerical methods for solving
    linear equation systems (Markov chains)
    linear programs (MDP)

Conclusion

- model checking for systems with discrete probabilities
  - techniques for verifying non-probabilistic systems
    graph algorithms, automata, ...
  - numerical methods for solving
    linear equation systems (Markov chains)
    linear programs (MDP)
- tool support
  - symbolic MTBDD-based PRISM [Kwiatk. et al]
  - abstraction, bisimulation MRMC [Katoen et al]
  - abstraction-refinement PASS [Hermanns et al]
    RAPTURE [d'Argenio et al]
  - partial order reduction LiQuor [Ciesinski et al]

    ⋮

Conclusion

PMC-CONC

- model checking for systems with discrete probabilities
  - techniques for verifying non-probabilistic systems
    graph algorithms, automata, ...
  - numerical methods for solving
    linear equation systems (Markov chains)
    linear programs (MDP)
- very active research field
  - continuous-time and -space
  - probabilistic real-time/hybrid systems
  - stochastic games
  - various techniques for state explosion problem
  - applications in system biology, security, ...

    ⋮