Probabilistic Modeling and Model Checking

Erika Ábrahám

RWTH Aachen University, Germany

SFM'14 Tutorial
June 17, 2014

Where are probabilities in Computer Science?

Where are probabilities in Computer Science?

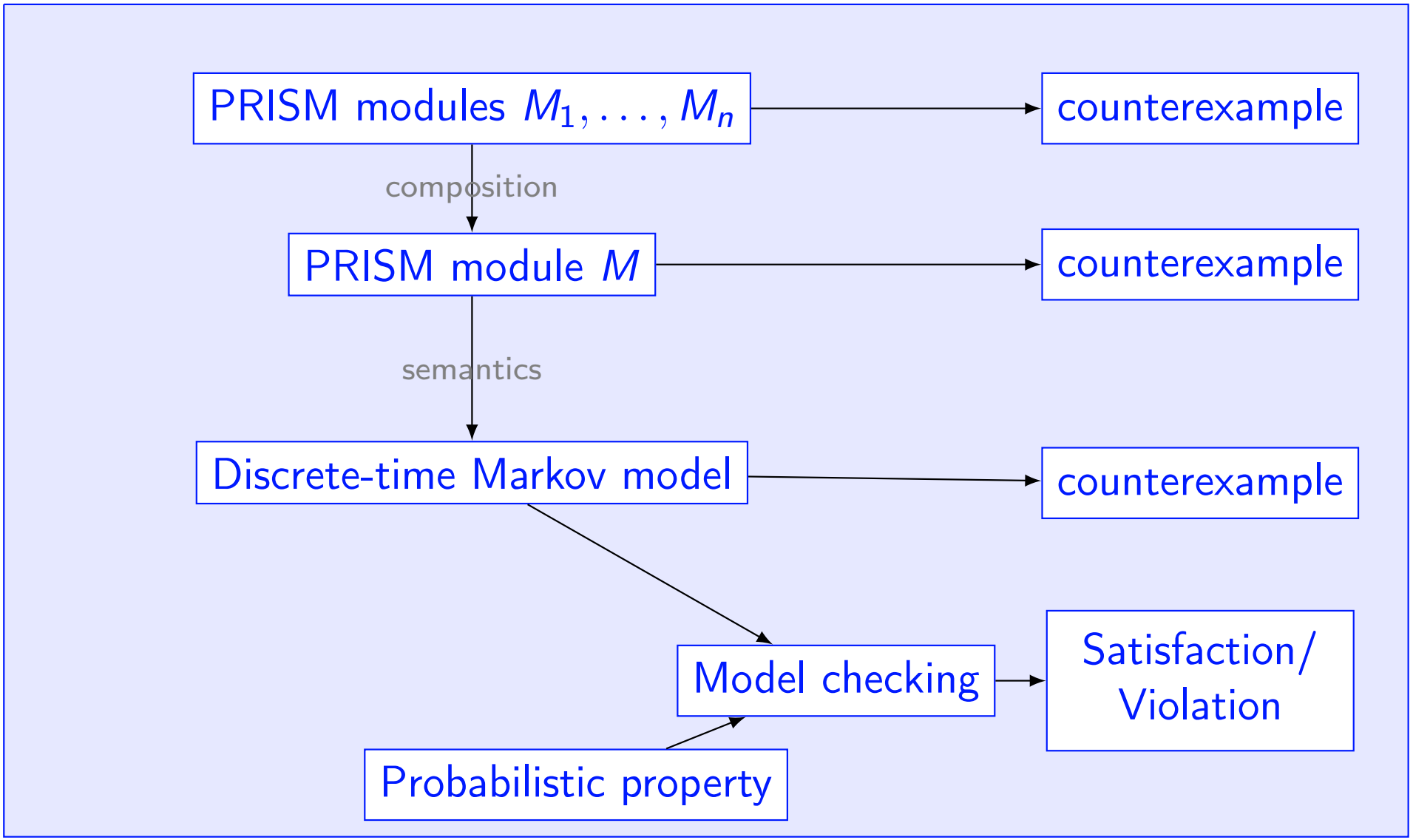
- Randomized algorithms
  - Randomized protocols for networked systems, Google search algorithm, cryptography, ...
- Unreliable or unpredictable system behaviors
  - Message losses, processor failures, unpredictable delays, soft deadlines, ...
- System performance and dependability
  - Quantify arrivals, waiting times, time between failures, ...
- Complex systems
  - Social networks, chemical plants, biological systems, ...

Where are probabilities in Computer Science?

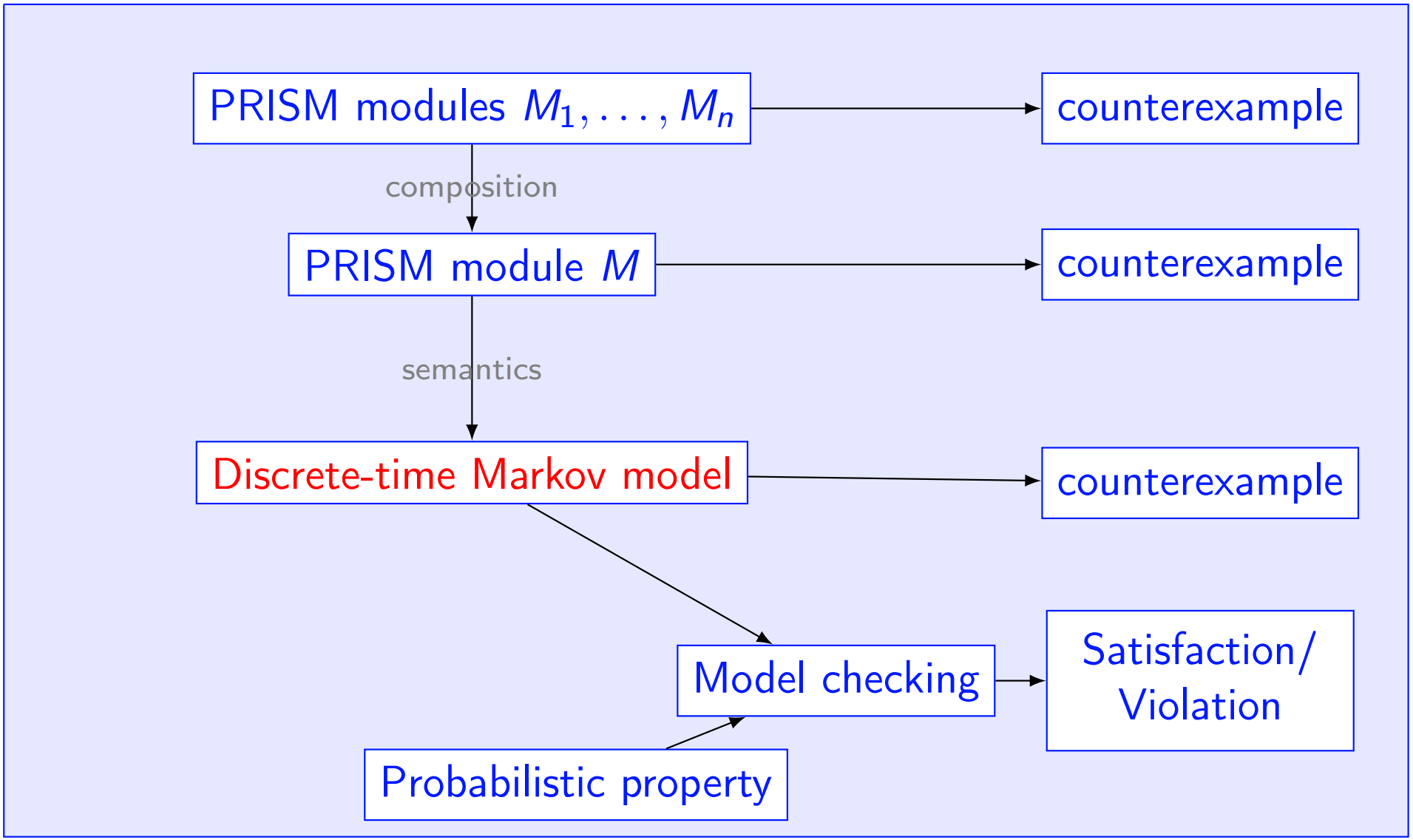
- Randomized algorithms
  - Randomized protocols for networked systems, Google search algorithm, cryptography, ...
- Unreliable or unpredictable system behaviors
  - Message losses, processor failures, unpredictable delays, soft deadlines, ...
- System performance and dependability
  - Quantify arrivals, waiting times, time between failures, ...
- Complex systems
  - Social networks, chemical plants, biological systems, ...

Topic of this tutorial: Modeling and analyzing such systems

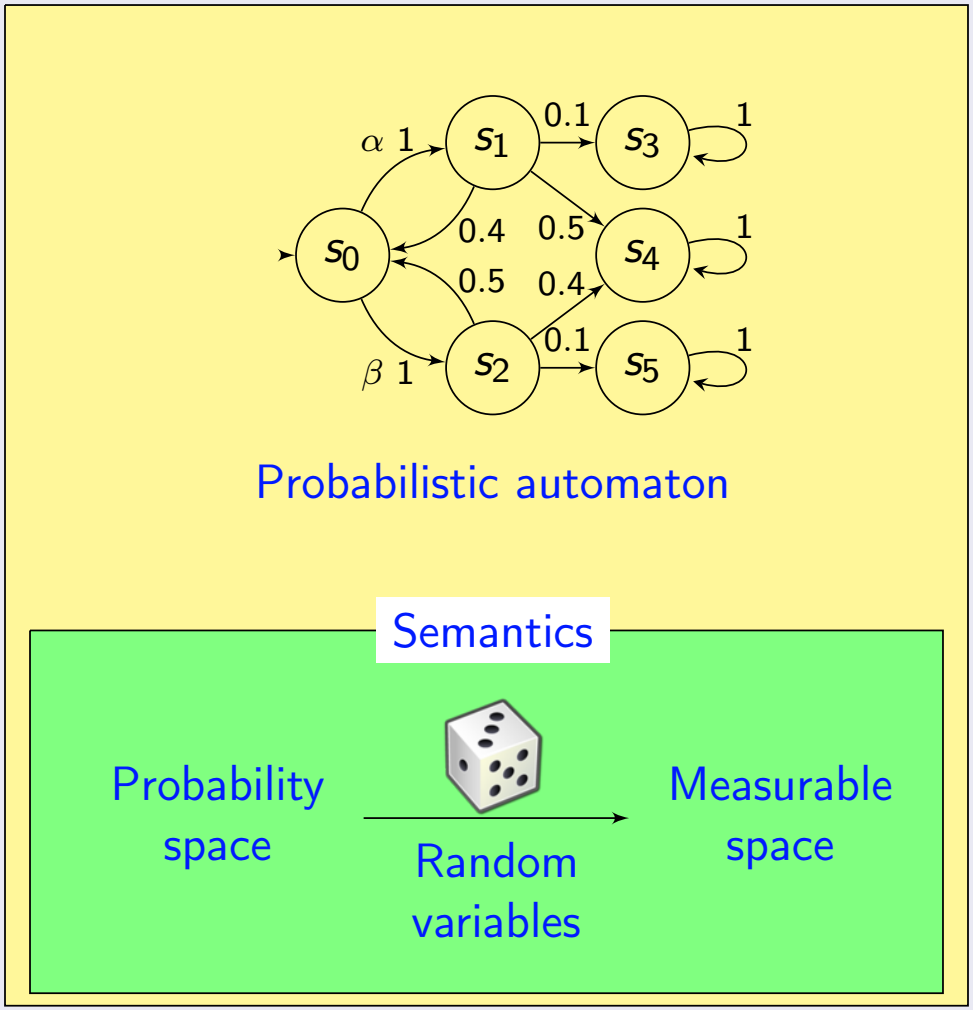
Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

The intuition behind the theory



Experiments

Definition (Experiments)

A random experiment is an act with uncertain outcome.

Examples

- Rolling a fair die
- Tossing a fair coin twice

Sample spaces

Definition (Sample spaces)

The sample space Ω of an experiment is the set of all possible outcomes (sample points) of the experiment.

NB: Sample spaces can be infinite or even uncountable. In this tutorial we deal with countable sample spaces only.

Examples

- Rolling a die:
- Tossing a coin twice:

Sample spaces

Definition (Sample spaces)

The sample space Ω of an experiment is the set of all possible outcomes (sample points) of the experiment.

NB: Sample spaces can be infinite or even uncountable. In this tutorial we deal with countable sample spaces only.

Examples

- Rolling a die: $\Omega = [1, 6] \subseteq \mathbb{N}$
- Tossing a coin twice: $\Omega = \{H, T\}^2$

Events

Definition (Events)

Given a sample space Ω, an event is a subset of Ω. A event containing a single sample point is a simple event.

Examples

Rolling a die: $\Omega = [1, 6]$

Tossing a coin twice: $\Omega = \{H, T\}^2$

Events

Definition (Events)

Given a sample space Ω, an event is a subset of Ω. A event containing a single sample point is a simple event.

Examples

Rolling a die: $\Omega = [1, 6]$

$\{6\}$	is the simple event of rolling a 6
$\{1, 3, 5\}$	is the event of rolling an odd number
$[1, 6]$	is the event of rolling any number $1, \ldots, 6$

Tossing a coin twice: $\Omega = \{H, T\}^2$

$\{(H, H)\}$	is the simple event of tossing heads twice
$\{(H, H), (H, T)\}$	is the event of tossing heads first
$\emptyset$	is the event of tossing nothing

Operations on events

Operations on events

$\bar{E} = \Omega \backslash E$	E does not occur
$E_1 \cup E_2$	either E_1 or E_2 occur
E_1, E_2	both E_1 and E_2 occur
$E_1 \backslash E_2 = E_1, \bar{E}_2$	E_1 occurs and E_2 not

σ-algebras

Definition (σ-algebra)

Given a sample space Ω, a σ-algebra $\mathcal{F}$ is a set of events containing the maximal event Ω and being closed under complement and countable union.

Examples

Rolling a die: $\Omega = [1, 6]$

$\{\{6\}, \Omega, \emptyset\}$	is
$\{\{6\}, \Omega, \emptyset, [1, 5]\}$	is

Tossing a coin twice: $\Omega = \{H, T\}^2$

$\{\Omega, \emptyset\}$	is
$\{\{(H, H)\}, \Omega, \emptyset\}$	is
2^{Ω}	is

σ-algebras

Definition (σ-algebra)

Given a sample space Ω, a σ-algebra $\mathcal{F}$ is a set of events containing the maximal event Ω and being closed under complement and countable union.

Examples

Rolling a die: $\Omega = [1, 6]$

$\{\{6\}, \Omega, \emptyset\}$	is	not a σ-algebra.
$\{\{6\}, \Omega, \emptyset, [1, 5]\}$	is	a σ-algebra.

Tossing a coin twice: $\Omega = \{H, T\}^2$

$\{\Omega, \emptyset\}$	is	the smallest σ-algebra.
$\{\{(H, H)\}, \Omega, \emptyset\}$	is	not a σ-algebra.
2^Ω	is	the largest σ-algebra.

Probability measures

Definition (Probability measures)

Given $(\Omega, \mathcal{F})$, a probability measure is a function $Pr : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ with

- $Pr(\Omega) = 1$,
- $Pr(E) = 1 - Pr(\bar{E})$ for all $E \in \mathcal{F}$
- $Pr(\cup_{i=0}^{\infty} E_i) = \sum_{i=0}^{\infty} Pr(E_i)$ where $E_i \in \mathcal{F}$ and $E_i \cap E_j = 0$ for all $i, j \in \mathbb{N}$, $i \neq j$.

Example

Rolling a die: $\Omega = [1,6]$, $\mathcal{F} = 2^{\Omega}$

$Pr(\{6\}) =$	The probability of rolling a 6 is
$Pr(\{1,3,5\}) =$	The probability of rolling an odd number is
$Pr([1,6]) =$	The probability of rolling any number is

Probability measures

Definition (Probability measures)

Given $(\Omega, \mathcal{F})$, a probability measure is a function $Pr : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ with

- $Pr(\Omega) = 1$,
- $Pr(E) = 1 - Pr(\bar{E})$ for all $E \in \mathcal{F}$
- $Pr(\cup_{i=0}^{\infty} E_i) = \sum_{i=0}^{\infty} Pr(E_i)$ where $E_i \in \mathcal{F}$ and $E_i \cap E_j = 0$ for all $i, j \in \mathbb{N}$, $i \neq j$.

Example

Rolling a die: $\Omega = [1,6]$, $\mathcal{F} = 2^{\Omega}$

$Pr(\{6\}) = 1/6$	The probability of rolling a 6 is $1/6$.
$Pr(\{1,3,5\}) = 1/2$	The probability of rolling an odd number is $1/2$.
$Pr([1,6]) = 1$	The probability of rolling any number is 1.

Probability measures

Definition (Probability measures)

Given $(\Omega, \mathcal{F})$, a probability measure is a function $Pr : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ with

- $Pr(\Omega) = 1$,
- $Pr(E) = 1 - Pr(\bar{E})$ for all $E \in \mathcal{F}$
- $Pr(\cup_{i=0}^{\infty} E_i) = \sum_{i=0}^{\infty} Pr(E_i)$ where $E_i \in \mathcal{F}$ and $E_i \cap E_j = 0$ for all $i, j \in \mathbb{N}$, $i \neq j$.

Example

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = 2^{\Omega}$

$Pr(\{(H,H),(H,T)\}) =$	The probability of tossing heads first
$Pr(\Omega) =$	The probability of tossing anything is
$Pr(\emptyset) =$	The probability of tossing nothing is

Probability measures

Definition (Probability measures)

Given $(\Omega, \mathcal{F})$, a probability measure is a function $Pr : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ with

- $Pr(\Omega) = 1$,
- $Pr(E) = 1 - Pr(\bar{E})$ for all $E \in \mathcal{F}$
- $Pr(\cup_{i=0}^{\infty} E_i) = \sum_{i=0}^{\infty} Pr(E_i)$ where $E_i \in \mathcal{F}$ and $E_i \cap E_j = 0$ for all $i, j \in \mathbb{N}$, $i \neq j$.

Example

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = 2^{\Omega}$

$Pr(\{(H,H),(H,T)\}) = 1/2$	The probability of tossing heads first $1/2$.
$Pr(\Omega) = 1$	The probability of tossing anything is 1.
$Pr(\emptyset) = 0$	The probability of tossing nothing is 0.

Measurability

Definition (Measurability)

An event is measurable if it is in the domain of Pr (i.e., in $\mathcal{F}$).

$(\Omega, \mathcal{F})$ is also called a measurable space.

Examples

Rolling a die: $\Omega = [1, 6]$, $\mathcal{F} = \{[1, 5], \{6\}, \Omega, \emptyset\}$
$Pr(\{1\}) =$ Rolling a 1 is

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = \{\Omega, \emptyset\}$
$Pr(\{(H, H)\}) =$ Tossing heads twice is

Measurability

Definition (Measurability)

An event is measurable if it is in the domain of Pr (i.e., in $\mathcal{F}$).

$(\Omega, \mathcal{F})$ is also called a measurable space.

Examples

Rolling a die: $\Omega = [1, 6]$, $\mathcal{F} = \{[1, 5], \{6\}, \Omega, \emptyset\}$
$Pr(\{1\}) = \bot$ Rolling a 1 is not measurable.

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = \{\Omega, \emptyset\}$
$Pr(\{(H, H)\}) = \bot$ Tossing heads twice is not measurable.

Probability spaces

Definition (Probability spaces)

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

- $(\Omega, \mathcal{F})$ a measurable space and
- Pr a probability measure for $(\Omega, \mathcal{F})$.

Random variables

Definition (Random variables)

Assume $(\Omega, \mathcal{F}, Pr)$ and a measurable space (S, Σ).

- For $X : \Omega \to S$, $s \in S$ and $\sigma \in \Sigma$ we define
  - $X = s$ to be $\{\omega \in \Omega \mid X(\omega) = s\}$ and
  - $X^{-1}(\sigma) = \cup_{s \in \sigma}(X = s)$.
- X is measurable if $X^{-1}(\sigma) \in \mathcal{F}$ for all $\sigma \in \Sigma$.
- A random variable is a measurable function $X : \Omega \to S$.

Random variables

Definition (Random variables)

Assume $(\Omega, \mathcal{F}, Pr)$ and a measurable space (S, Σ).

- For $X : \Omega \to S$, $s \in S$ and $\sigma \in \Sigma$ we define
  - $X = s$ to be $\{\omega \in \Omega \mid X(\omega) = s\}$ and
  - $X^{-1}(\sigma) = \cup_{s \in \sigma}(X = s)$.
- X is measurable if $X^{-1}(\sigma) \in \mathcal{F}$ for all $\sigma \in \Sigma$.
- A random variable is a measurable function $X : \Omega \to S$.

Examples

Rolling a die: $\Omega = [1, 6]$, $\mathcal{F} = 2^{\Omega}$, $S = \Omega$, X is identity

$Pr(X = 2) \quad =$

Random variables

Definition (Random variables)

Assume $(\Omega, \mathcal{F}, Pr)$ and a measurable space (S, Σ).

- For $X : \Omega \to S$, $s \in S$ and $\sigma \in \Sigma$ we define
  - $X = s$ to be $\{\omega \in \Omega \mid X(\omega) = s\}$ and
  - $X^{-1}(\sigma) = \cup_{s \in \sigma}(X = s)$.
- X is measurable if $X^{-1}(\sigma) \in \mathcal{F}$ for all $\sigma \in \Sigma$.
- A random variable is a measurable function $X : \Omega \to S$.

Examples

Rolling a die: $\Omega = [1, 6]$, $\mathcal{F} = 2^{\Omega}$, $S = \Omega$, X is identity

$Pr(X = 2) \quad = Pr(\{2\}) = 1/6$

Random variables

Definition (Random variables)

Assume $(\Omega, \mathcal{F}, Pr)$ and a measurable space (S, Σ).

- For $X : \Omega \to S$, $s \in S$ and $\sigma \in \Sigma$ we define
  - $X = s$ to be $\{\omega \in \Omega \mid X(\omega) = s\}$ and
  - $X^{-1}(\sigma) = \cup_{s \in \sigma}(X = s)$.
- X is measurable if $X^{-1}(\sigma) \in \mathcal{F}$ for all $\sigma \in \Sigma$.
- A random variable is a measurable function $X : \Omega \to S$.

Examples

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = 2^{\Omega}$, $S = [0, 2]$, X counts heads
$Pr(X = 1)$

Random variables

Definition (Random variables)

Assume $(\Omega, \mathcal{F}, Pr)$ and a measurable space (S, Σ).

- For $X : \Omega \to S$, $s \in S$ and $\sigma \in \Sigma$ we define
  - $X = s$ to be $\{\omega \in \Omega \mid X(\omega) = s\}$ and
  - $X^{-1}(\sigma) = \cup_{s \in \sigma}(X = s)$.
- X is measurable if $X^{-1}(\sigma) \in \mathcal{F}$ for all $\sigma \in \Sigma$.
- A random variable is a measurable function $X : \Omega \to S$.

Examples

Tossing a coin twice: $\Omega = \{H, T\}^2$, $\mathcal{F} = 2^{\Omega}$, $S = [0, 2]$, X counts heads

$Pr(X = 1) \quad = Pr(\{(H, T), (T, H)\}) = 2/4$

Some relevant properties

Definition

- Given two random variables X and Y and random values x and y with $Pr(X = x) > 0$, the conditional probability of $Y = y$ given $X = x$ is

$$Pr(Y = y \,|\, X = x) \quad = \quad \frac{Pr(Y = y, X = x)}{Pr(X = x)}\,.$$

Some relevant properties

Definition

- Given two random variables X and Y and random values x and y with $Pr(X = x) > 0$, the conditional probability of $Y = y$ given $X = x$ is

$$Pr(Y = y \,|\, X = x) \quad = \quad \frac{Pr(Y = y, X = x)}{Pr(X = x)}\,.$$

- Two random variables X and y are independent if

$$Pr(X = x, Y = y) \quad = \quad Pr(X = x) \cdot Pr(Y = y)$$

for all random values x and y.

Some relevant properties

Definition

- Given two random variables X and Y and random values x and y with $Pr(X = x) > 0$, the conditional probability of $Y = y$ given $X = x$ is

$$Pr(Y = y \mid X = x) \quad = \quad \frac{Pr(Y = y, X = x)}{Pr(X = x)}\,.$$

- Two random variables X and y are independent if

$$Pr(X = x, Y = y) \quad = \quad Pr(X = x) \cdot Pr(Y = y)$$

  for all random values x and y.

- The expected value of a random variable X is

$$\sum_{s \in S} s \cdot Pr(X = s)\,.$$

Stochastic processes: Some historical notes

Stochastic processes: Some historical notes

- Starting in the late 19th century
- 1880 Thorvald N. Thiele: least squares for Brownian motion
- 1900 Louis Bachelier: stochastic analysis of the financial markets
- 1905 Albert Einstein, 1906 Marian Smoluchowski, 1908 Jean Baptiste Perrin: indirect proof of existence of atoms and molecules

Stochastic processes: Some historical notes

- Starting in the late 19th century
- 1880 Thorvald N. Thiele: least squares for Brownian motion
- 1900 Louis Bachelier: stochastic analysis of the financial markets
- 1905 Albert Einstein, 1906 Marian Smoluchowski, 1908 Jean Baptiste Perrin: indirect proof of existence of atoms and molecules

"It must clearly be assumed that [...] the movements of one and the same particle in different time intervals are independent processes, as long as these time intervals are not chosen too small.

We introduce a time interval τ into consideration, which is very small [...], but nevertheless so large that in two successive time intervals τ, the motions executed by the particle can be thought of as events which are independent of each other". [Albert Einstein]

Stochastic processes

Stochastic processes

Definition (Stochastic processes)

Given $(\Omega, \mathcal{F}, Pr)$ and (S, Σ), a stochastic process is a collection of random variables, indexed by a totally ordered set T ("time"):

$$\{X_t \mid t \in T\}.$$

The space S is called the state space of the process.

Stochastic processes

Definition (Stochastic processes)

Given $(\Omega, \mathcal{F}, Pr)$ and (S, Σ), a stochastic process is a collection of random variables, indexed by a totally ordered set T ("time"):

$$\{X_t \mid t \in T\}.$$

The space S is called the state space of the process.

Classes of state-based Markov models

	Deterministic	Non-deterministic
Discrete time	Discrete-time Markov chains (DTMCs)	Markov decision processes (MDPs) Probabilistic automata (PA)
Continuous time	Continuous-time Markov chains (CTMCs)	Continuous-time Markov decision processes (CTMDPs)

Discrete-time Markov chains

Discrete-time Markov chains

- Stochastic process as basic model
  Ω: (infinite) experiment sequences, S: system states

Discrete-time Markov chains

- Stochastic process as basic model
  Ω: (infinite) experiment sequences, S: system states
- Discrete-time stochastic process: time model is discrete
  (one execution step = one time unit)

Discrete-time Markov chains

- Stochastic process as basic model
  Ω: (infinite) experiment sequences, S: system states
- Discrete-time stochastic process: time model is discrete
  (one execution step = one time unit)
- Discrete-time Markov process: discrete-time stochastic process with the "memoryless" Markov-property:

$$Pr(X_{t+1}{=}x_{t+1} \mid X_0 X_1 \ldots X_t{=}x_0 x_1 \ldots x_t) = Pr(X_{t+1}{=}x_{t+1} \mid X_t{=}x_t)$$

Discrete-time Markov chains

- Stochastic process as basic model
  Ω: (infinite) experiment sequences, S: system states
- Discrete-time stochastic process: time model is discrete
  (one execution step = one time unit)
- Discrete-time Markov process: discrete-time stochastic process with the "memoryless" Markov-property:

$$Pr(X_{t+1}{=}x_{t+1} \mid X_0 X_1 \ldots X_t{=}x_0 x_1 \ldots x_t) = Pr(X_{t+1}{=}x_{t+1} \mid X_t{=}x_t)$$

- Time-homogeneity:

$$Pr(X_{t+1}{=}x_{t+1} \mid X_t{=}x_t) = Pr(X_{t'+1}{=}x_{t'+1} \mid X_{t'}{=}x_{t'})$$

Discrete-time Markov chains

- Stochastic process as basic model
  Ω: (infinite) experiment sequences, S: system states
- Discrete-time stochastic process: time model is discrete
  (one execution step = one time unit)
- Discrete-time Markov process: discrete-time stochastic process with the "memoryless" Markov-property:

$$Pr(X_{t+1}{=}x_{t+1} \mid X_0 X_1 \ldots X_t{=}x_0 x_1 \ldots x_t) = Pr(X_{t+1}{=}x_{t+1} \mid X_t{=}x_t)$$

- Time-homogeneity:

$$Pr(X_{t+1}{=}x_{t+1} \mid X_t{=}x_t) = Pr(X_{t'+1}{=}x_{t'+1} \mid X_{t'}{=}x_{t'})$$

- Discrete-time Markov chain (DTMC):
  time-homogeneous discrete-time discrete-space Markov process

Probability distributions

Probability distributions

Can we give an automata-based definition for DTMCs?

Probability distributions

Can we give an automata-based definition for DTMCs?

Definition (Probability distributions)

A (discrete probability) distribution over a countable set S is a function $p : S \to [0, 1] \subseteq \mathbb{R}$ with $\sum_{s \in S} p(s) = 1$.

DTMCs: for each $x \in S$ we define the distribution $P(x, \cdot)$ by

$$P(x, y) = Pr(X_{t+1} = y \mid X_t = x)$$

for all $y \in S$.

Discrete-time Markov chains

Discrete-time Markov chains

A discrete-time Markov chain (DTMC) consists of

- a countable state space S,
- an initial state $s_{\text{init}} \in S$ in which the execution starts, and
- for each state $s \in S$ a distribution $P(s, \cdot)$ defining for each $s' \in S$ the probability $P(s, s')$ to move from s to s'.

Discrete-time Markov chains

A discrete-time Markov chain (DTMC) consists of

- a countable state space S,
- an initial state $s_{\text{init}} \in S$ in which the execution starts, and
- for each state $s \in S$ a distribution $P(s, \cdot)$ defining for each $s' \in S$ the probability $P(s, s')$ to move from s to s'.

Additionally,

- Labeling function $L : S \to 2^{AP}$ labels states with atomic propositions from AP.

Discrete-time Markov chains

A discrete-time Markov chain (DTMC) consists of

- a countable state space S,
- an initial state $s_{\text{init}} \in S$ in which the execution starts, and
- for each state $s \in S$ a distribution $P(s, \cdot)$ defining for each $s' \in S$ the probability $P(s, s')$ to move from s to s'.

Additionally,

- Labeling function $L : S \to 2^{AP}$ labels states with atomic propositions from AP.

Thus a DTMC is a tuple $(S, s_{\text{init}}, P, L)$.

Example DTMC [Kemeny, Snell and Thompson]

- In the Land of Oz there are never two nice days (N) in a row.
- A nice day is followed by a day with either rain (R) or snow (S) with equal probability.
- In half of the cases, a day with snow or rain is followed by the same kind; the remaining cases are equally probable.

Example DTMC [Kemeny, Snell and Thompson]

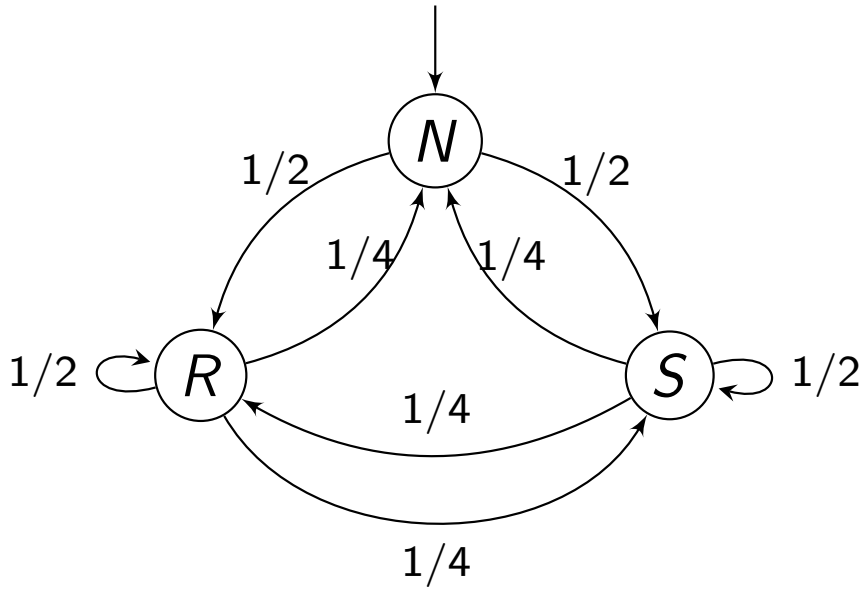
- In the Land of Oz there are never two nice days (N) in a row.
- A nice day is followed by a day with either rain (R) or snow (S) with equal probability.
- In half of the cases, a day with snow or rain is followed by the same kind; the remaining cases are equally probable.

$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$

Example DTMC [Kemeny, Snell and Thompson]

- In the Land of Oz there are never two nice days (N) in a row.
- A nice day is followed by a day with either rain (R) or snow (S) with equal probability.
- In half of the cases, a day with snow or rain is followed by the same kind; the remaining cases are equally probable.

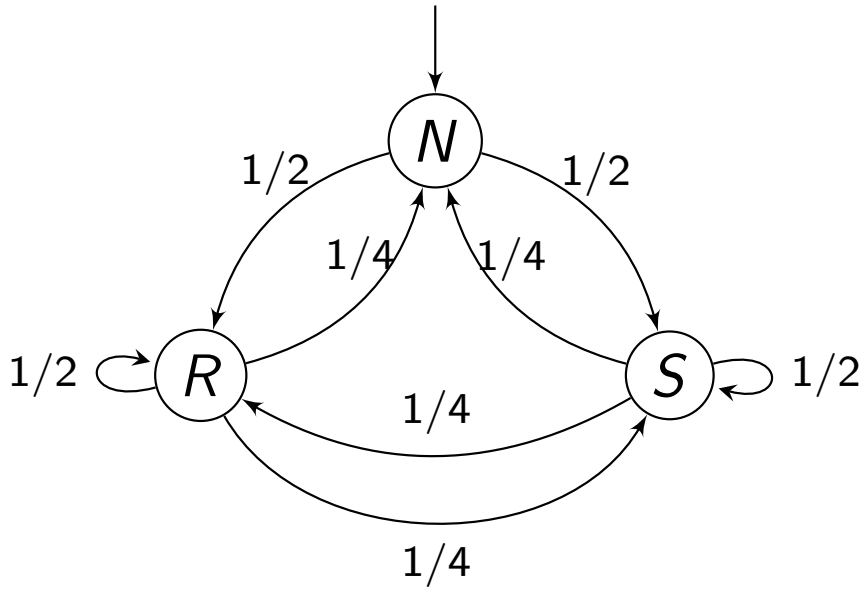
$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$



Example DTMC [Kemeny, Snell and Thompson]

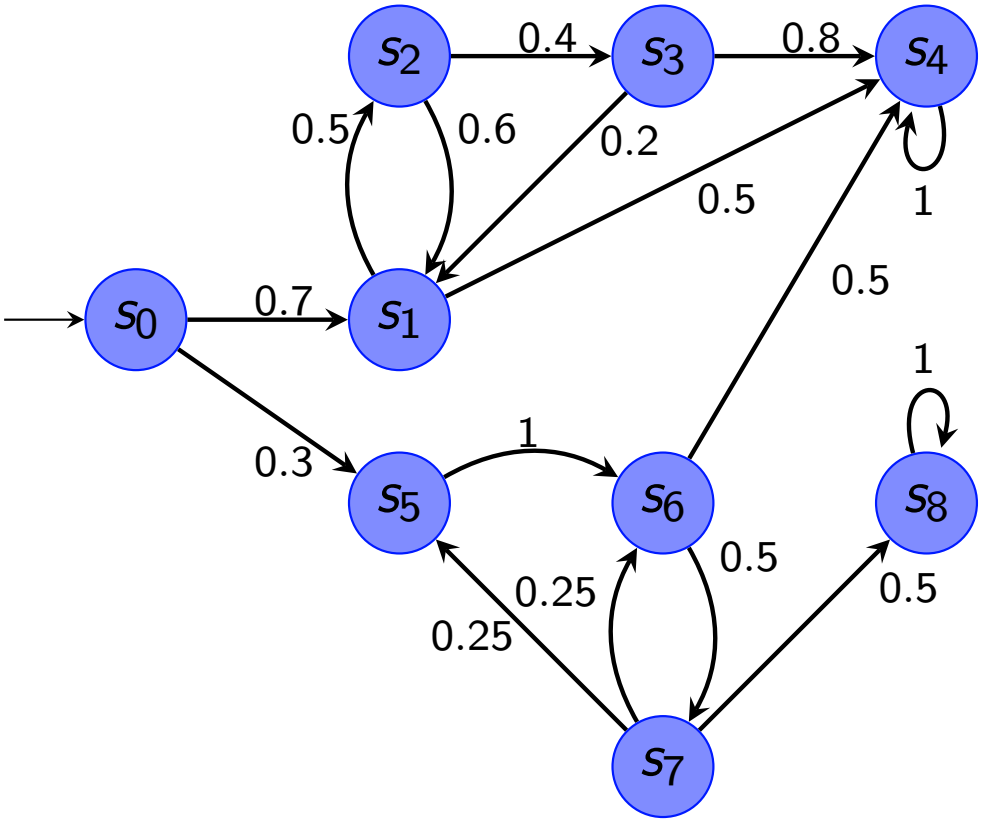
- In the Land of Oz there are never two nice days (N) in a row.
- A nice day is followed by a day with either rain (R) or snow (S) with equal probability.
- In half of the cases, a day with snow or rain is followed by the same kind; the remaining cases are equally probable.

$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$

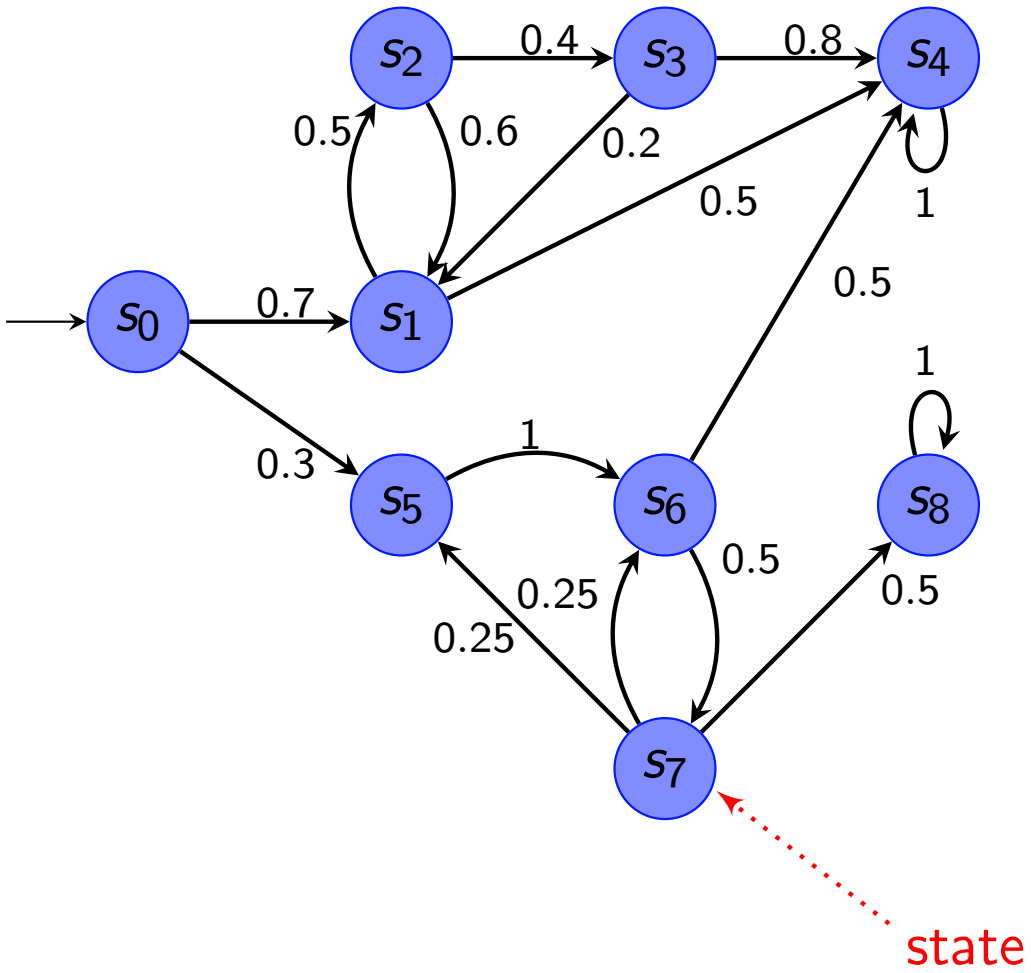


For all $n \in \mathbb{N}$, P^n is a stochastic matrix: its rows sum up to 1.

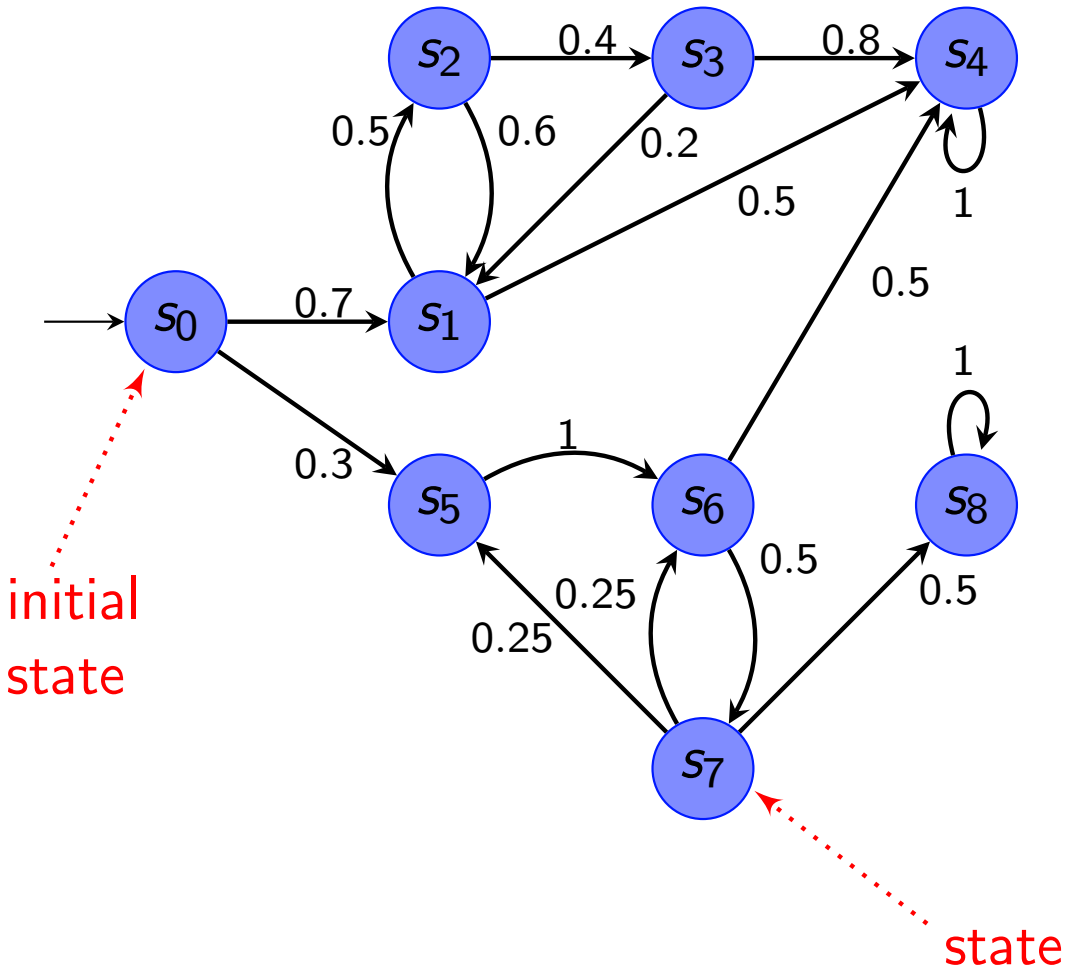
Another example DTMC



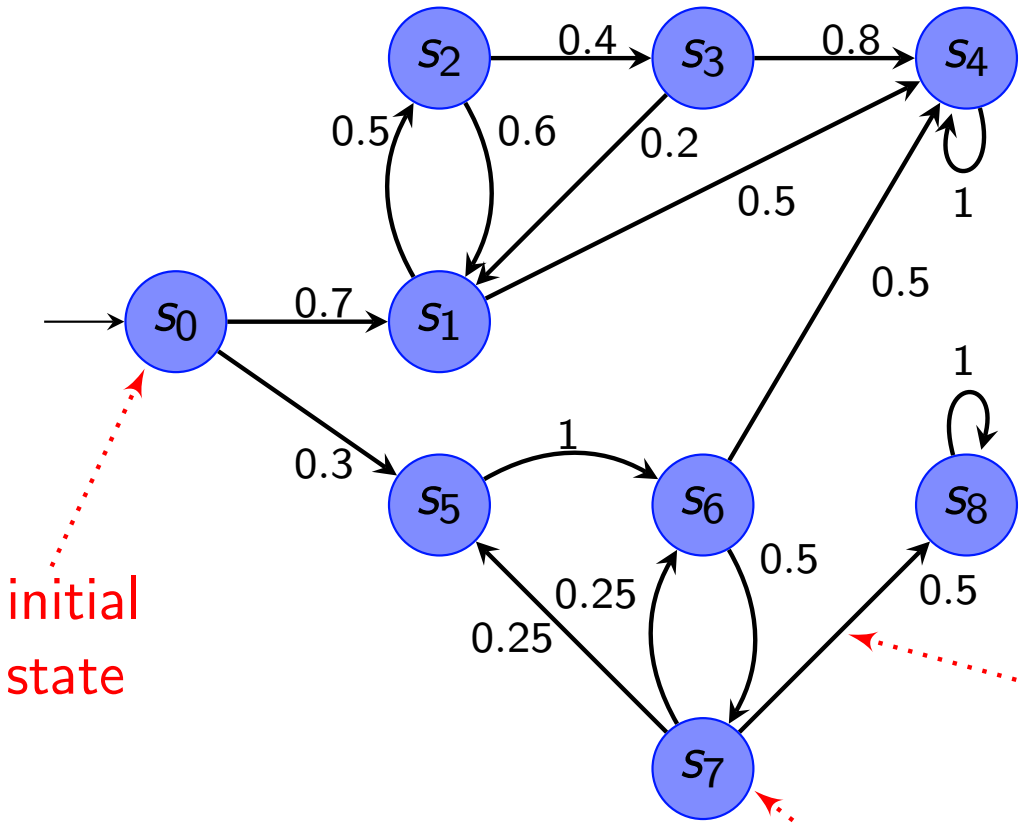
Another example DTMC



Another example DTMC



Another example DTMC

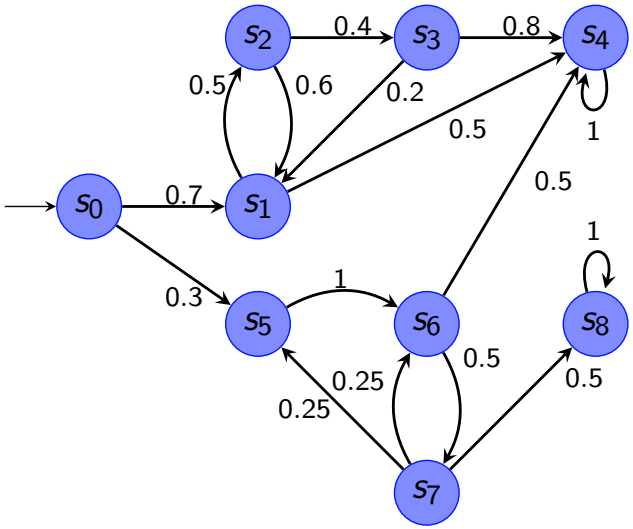


initial state

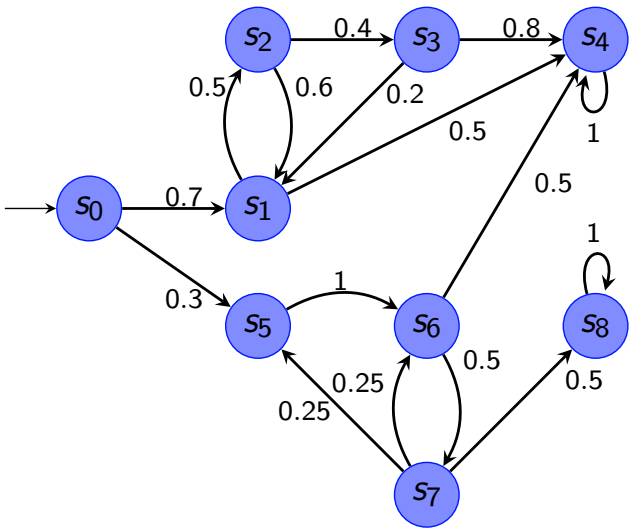
transition with probability $P(s_7, s_8) = 0.5$

state

Another example DTMC

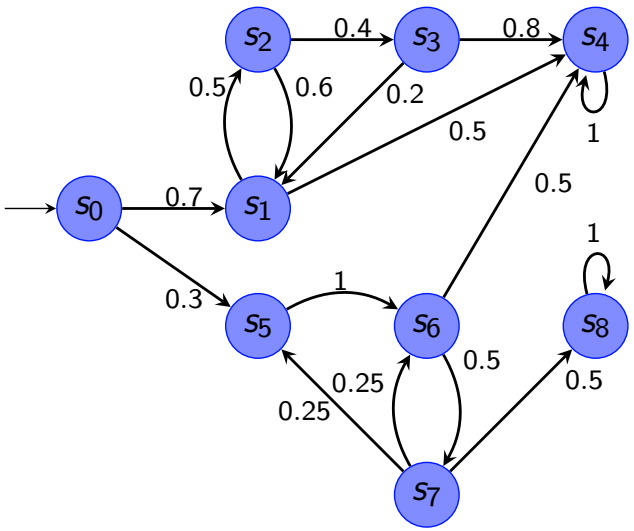


Another example DTMC



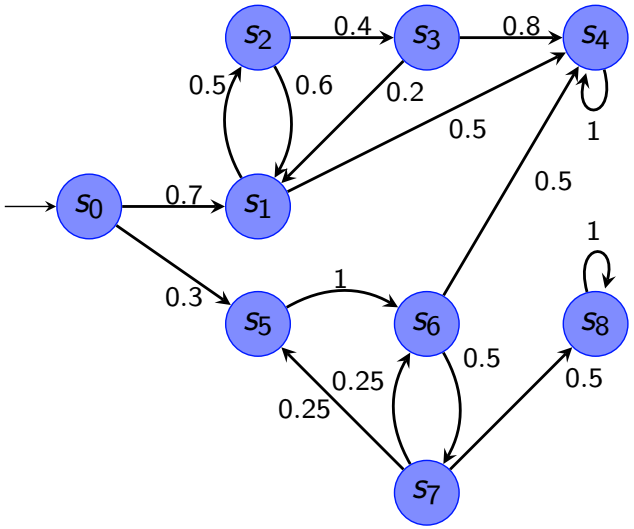
Paths are state sequences $s'_0 s'_1 \ldots$ with $P(s'_i, s'_{i+1}) > 0$

Another example DTMC



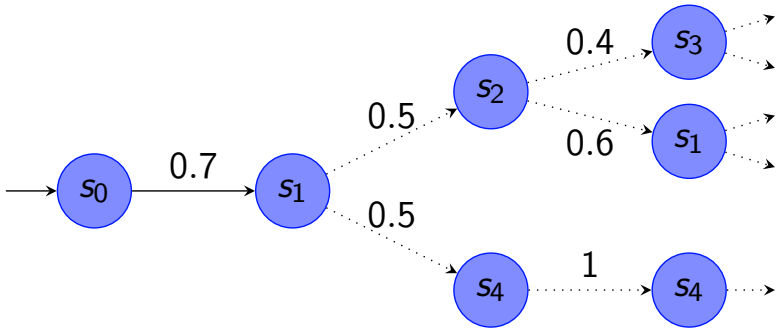
Paths are state sequences $s'_0 s'_1 \ldots$ with $P(s'_i, s'_{i+1}) > 0$ (probability?)

Another example DTMC

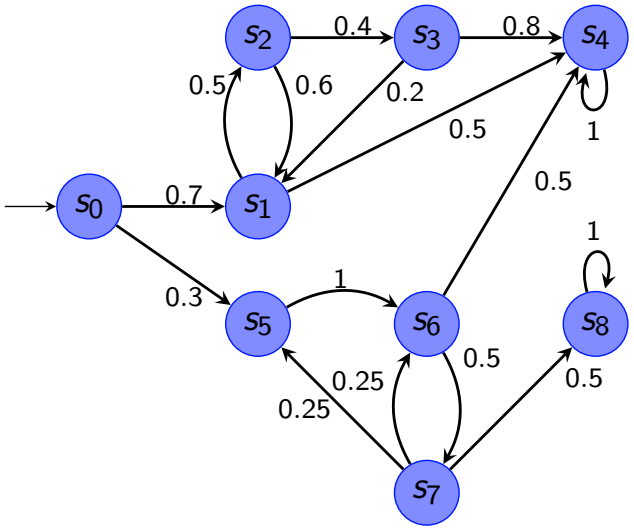


Paths are state sequences $s'_0 s'_1 \ldots$ with $P(s'_i, s'_{i+1}) > 0$ (probability?)

Cylinder set of finite path $s_0 s_1$:

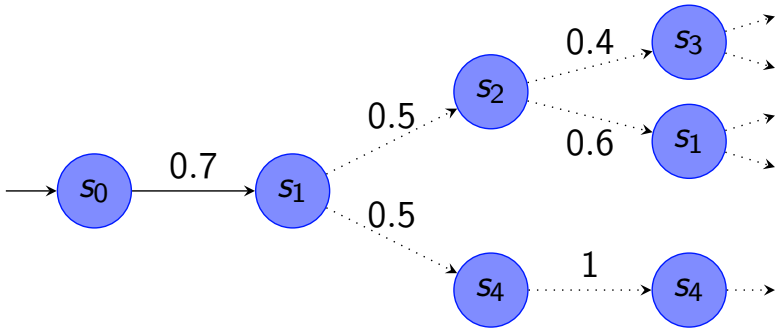


Another example DTMC



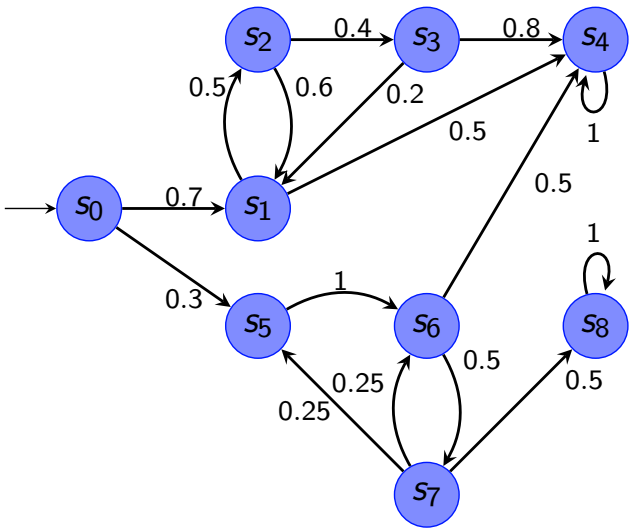
Paths are state sequences $s'_0 s'_1 \ldots$ with $P(s'_i, s'_{i+1}) > 0$ (probability?)

Cylinder set of finite path $s_0 s_1$:



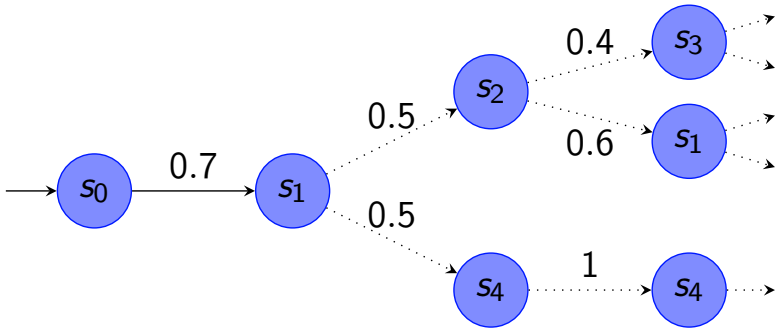
Semantical basis:

Another example DTMC



Paths are state sequences $s'_0 s'_1 \ldots$ with $P(s'_i, s'_{i+1}) > 0$ (probability?)

Cylinder set of finite path $s_0 s_1$:



Semantical basis: smallest σ-algebra over all cylinder sets with

$$Pr_{s_0}\big(\mathrm{Cyl}(s_0 \ldots s_n)\big) = P(s_0, s_1) \cdot Pr_{s_1}\big(\mathrm{Cyl}(s_1 \ldots s_n)\big) = \prod_{i=0}^{n-1} P(s_i, s_{i+1})\,.$$

We write $Pr_{s_0}(s_0 \ldots s_n)$ for $Pr_{s_0}\,(\mathrm{Cyl}(s_0 \ldots s_n))$.

n-step transition probabilities

Probability $P^{(i)}(s,t)$ to reach t from s in i steps:

$$P^{(0)}(s,t) = I(s,t) = \begin{cases} 1 & s = t \\ 0 & else \end{cases}$$

$$P^{(i+1)}(s,t) = \sum_{s' \in S} P(s,s') \cdot P^{(i)}(s',t) \quad \textit{(Chapman-Kolmogorov)}$$

n-step transition probabilities

Probability $P^{(i)}(s,t)$ to reach t from s in i steps:

$$P^{(0)}(s,t) = I(s,t) = \begin{cases} 1 & s = t \\ 0 & else \end{cases}$$

$$P^{(i+1)}(s,t) = \sum_{s' \in S} P(s,s') \cdot P^{(i)}(s',t) \quad \textit{(Chapman-Kolmogorov)}$$

i-step transition probabilities: $P^{(0)} = I$ (id matrix) and $P^{(i)} = P^i$ for $i > 0$.

n-step transition probabilities

Probability $P^{(i)}(s,t)$ to reach t from s in i steps:

$$P^{(0)}(s,t) = I(s,t) = \begin{cases} 1 & s = t \\ 0 & else \end{cases}$$

$$P^{(i+1)}(s,t) = \sum_{s' \in S} P(s,s') \cdot P^{(i)}(s',t) \quad \textit{(Chapman-Kolmogorov)}$$

i-step transition probabilities: $P^{(0)} = I$ (id matrix) and $P^{(i)} = P^i$ for $i > 0$.

Transient probability of reaching t from s_{init} in i steps:

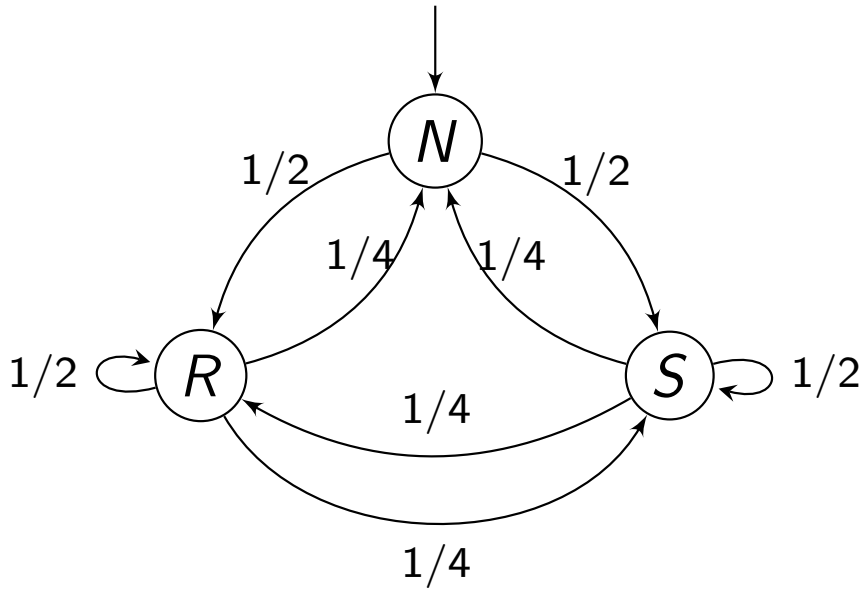
$$P^{(i)}(s_{\text{init}}, t)$$

Transient probability distribution:

$$\underbrace{s_{\text{init}}}_{\textit{characteristic vector for } s_{\text{init}}} \cdot P^i$$

Weather forecast in the Land of Oz

$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$



The world's largest matrix computation

The world's largest matrix computation

Google matrix for n pages:

$$G = dA + (1 - d)E$$

with

- d: damping factor
- A: $n \times n$ stochastic matrix obtained from the page link adjacency matrix of the web by scaling the rows
- E: $n \times n$ stochastic matrix with entries $1/n$

The world's largest matrix computation

Google matrix for n pages:

$$G = dA + (1 - d)E$$

with

- d: damping factor
- A: $n \times n$ stochastic matrix obtained from the page link adjacency matrix of the web by scaling the rows
- E: $n \times n$ stochastic matrix with entries $1/n$

G is an $n \times n$ stochastic matrix with eigenvalue 1.
The corresponding eigenvector scaled so that the largest value is 10 is called the page rank of d.

The world's largest matrix computation

Google matrix for n pages:

$$G = dA + (1 - d)E$$

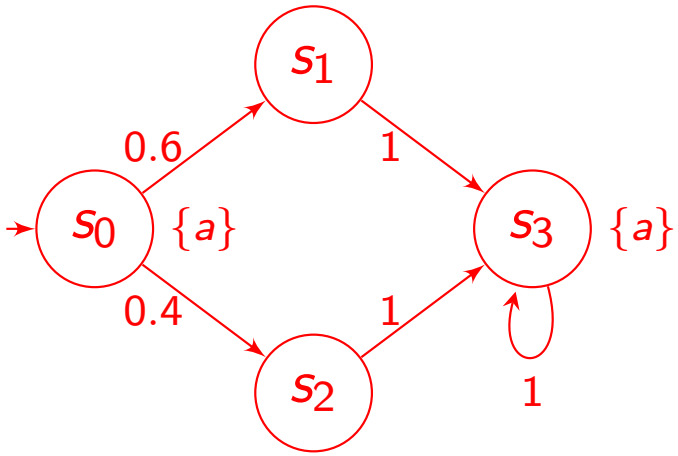
with

- d: damping factor
- A: $n \times n$ stochastic matrix obtained from the page link adjacency matrix of the web by scaling the rows
- E: $n \times n$ stochastic matrix with entries $1/n$

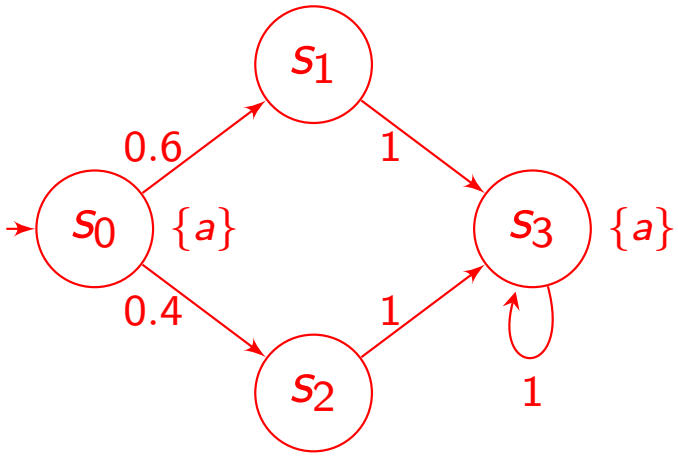
G is an $n \times n$ stochastic matrix with eigenvalue 1.
The corresponding eigenvector scaled so that the largest value is 10 is called the page rank of d.

In 2006: $n = 8.1$ billion

Symbolic representation of DTMCs

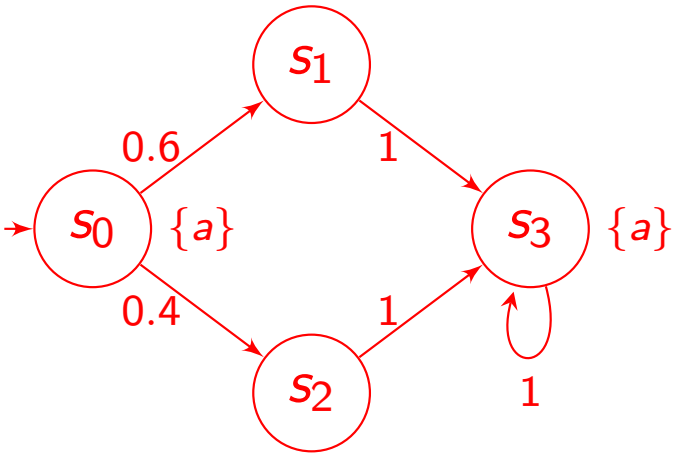


Symbolic representation of DTMCs

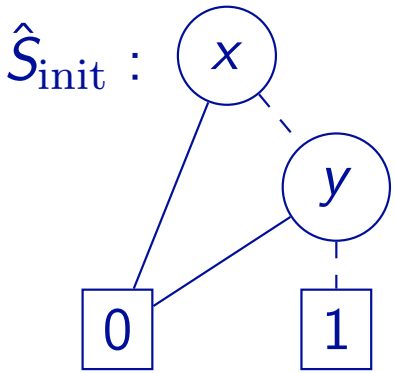


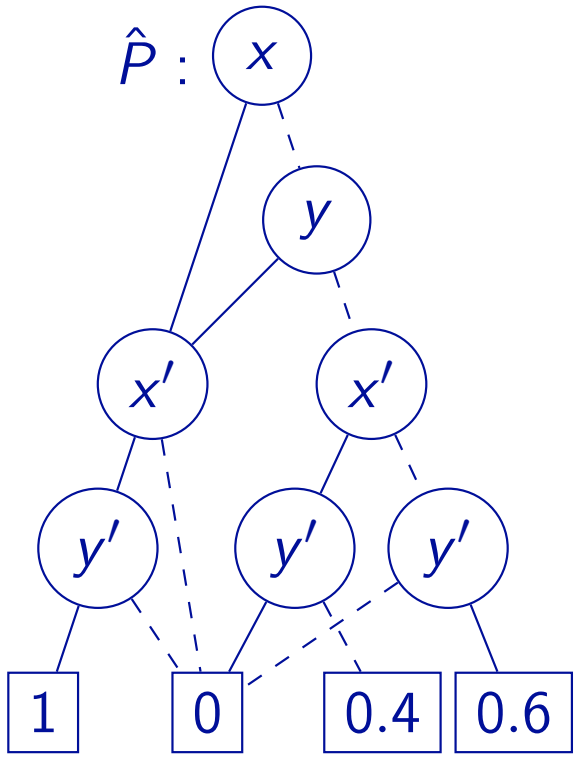
	s_0	s_1	s_2	s_3
x	0	0	1	1
y	0	1	0	1

Symbolic representation of DTMCs



	s_0	s_1	s_2	s_3
x	0	0	1	1
y	0	1	0	1



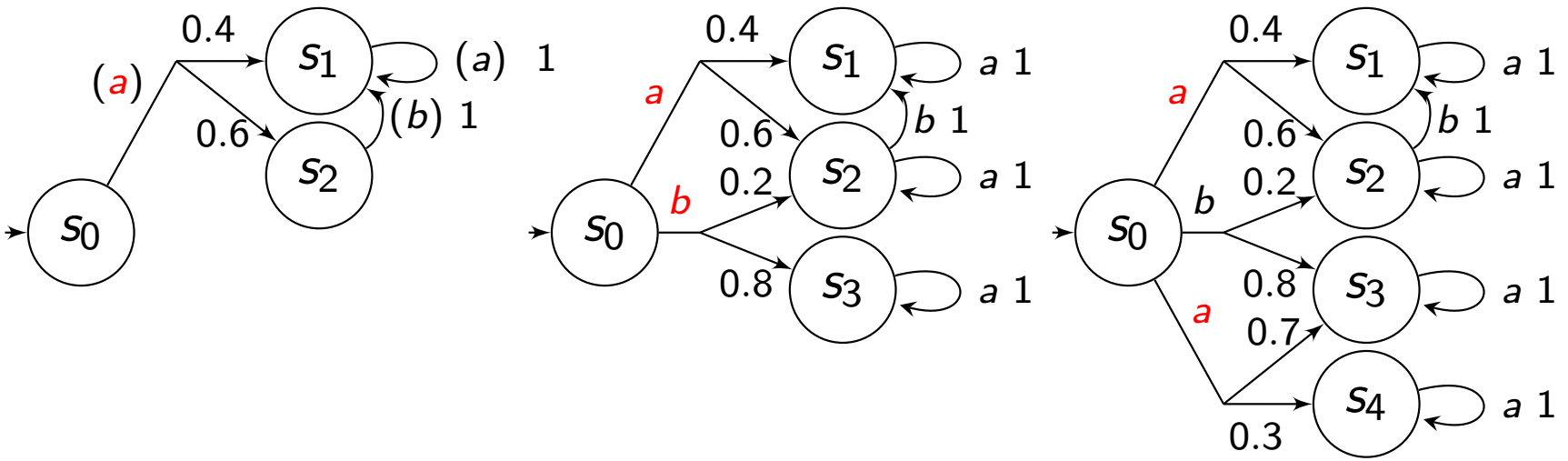


Markov decision processes (MDPs) and probabilistic automata (PA)

When non-determinism comes into play...

Markov decision processes (MDPs) and probabilistic automata (PA)

When non-determinism comes into play...



Discrete-time Markov chain

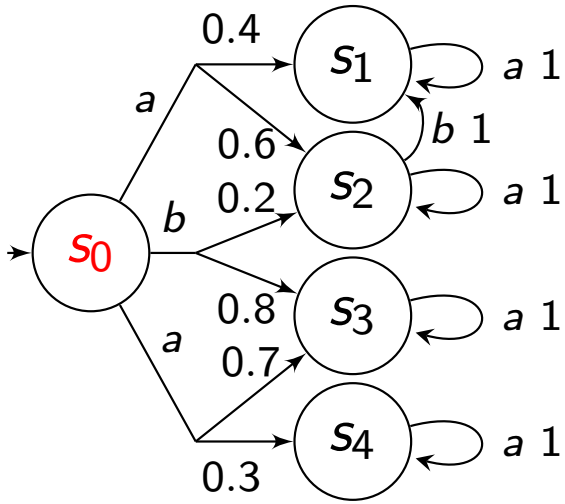
Probabilistic automaton

Markov decision process

Semantics of MDPs and PA

Atomic execution step:

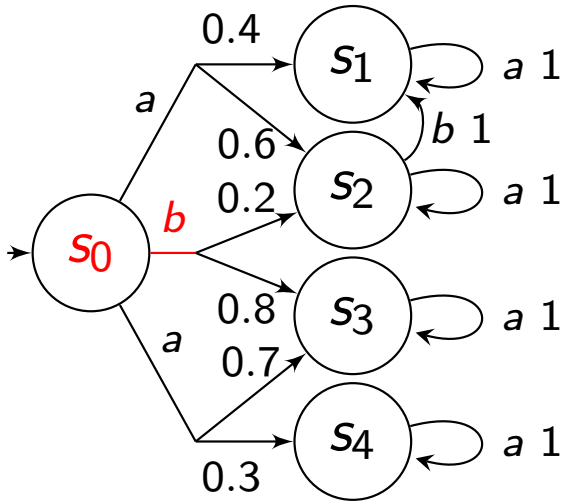
1. choice of an action-distribution pair
2. choice of a transition



Semantics of MDPs and PA

Atomic execution step:

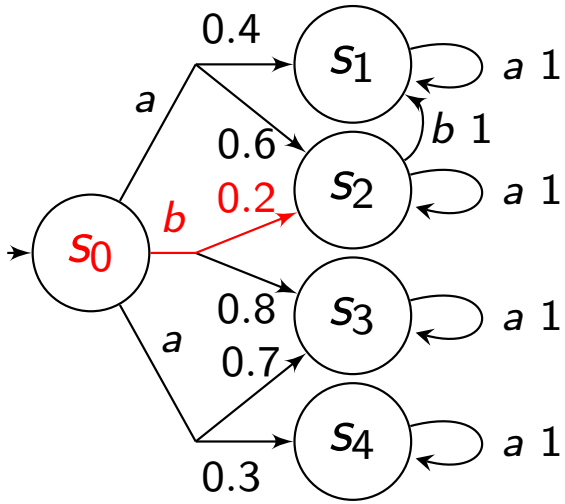
1. choice of an action-distribution pair
2. choice of a transition



Semantics of MDPs and PA

Atomic execution step:

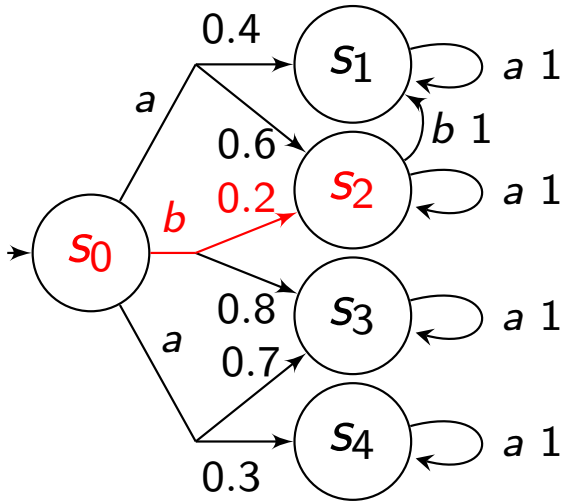
1. choice of an action-distribution pair
2. choice of a transition



Semantics of MDPs and PA

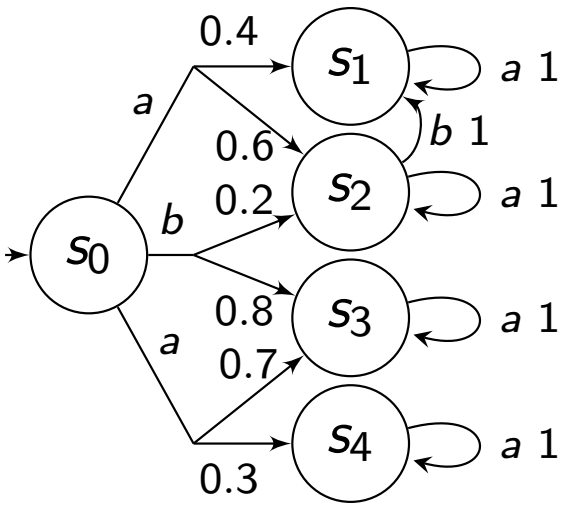
Atomic execution step:

1. choice of an action-distribution pair
2. choice of a transition



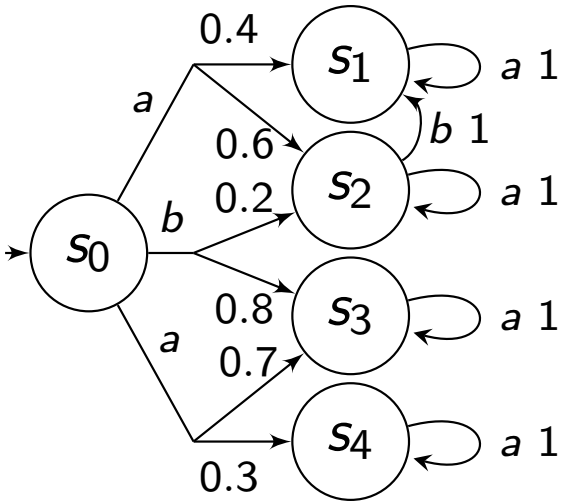
Scheduler

- The non-determinism is resolved by a scheduler.
- It assigns to each finite path a distribution over the action-distribution pairs possible in the last state.



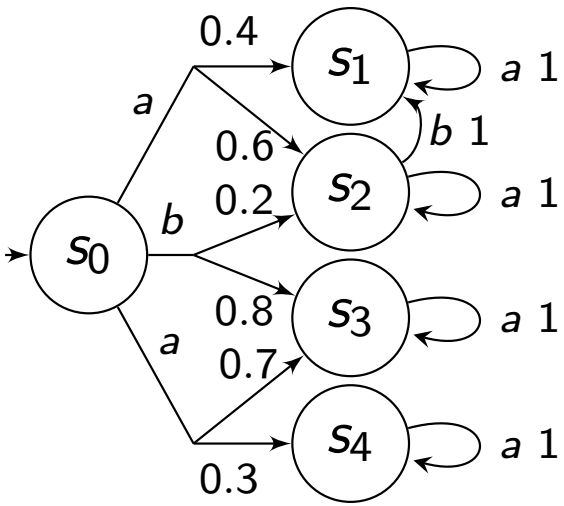
Scheduler

- The non-determinism is resolved by a scheduler.
- It assigns to each finite path a distribution over the action-distribution pairs possible in the last state.
- A deterministic scheduler puts the whole probability into a unique action-distribution pair.
- The decisions made by a memoryless scheduler depend only on the last state of the path.



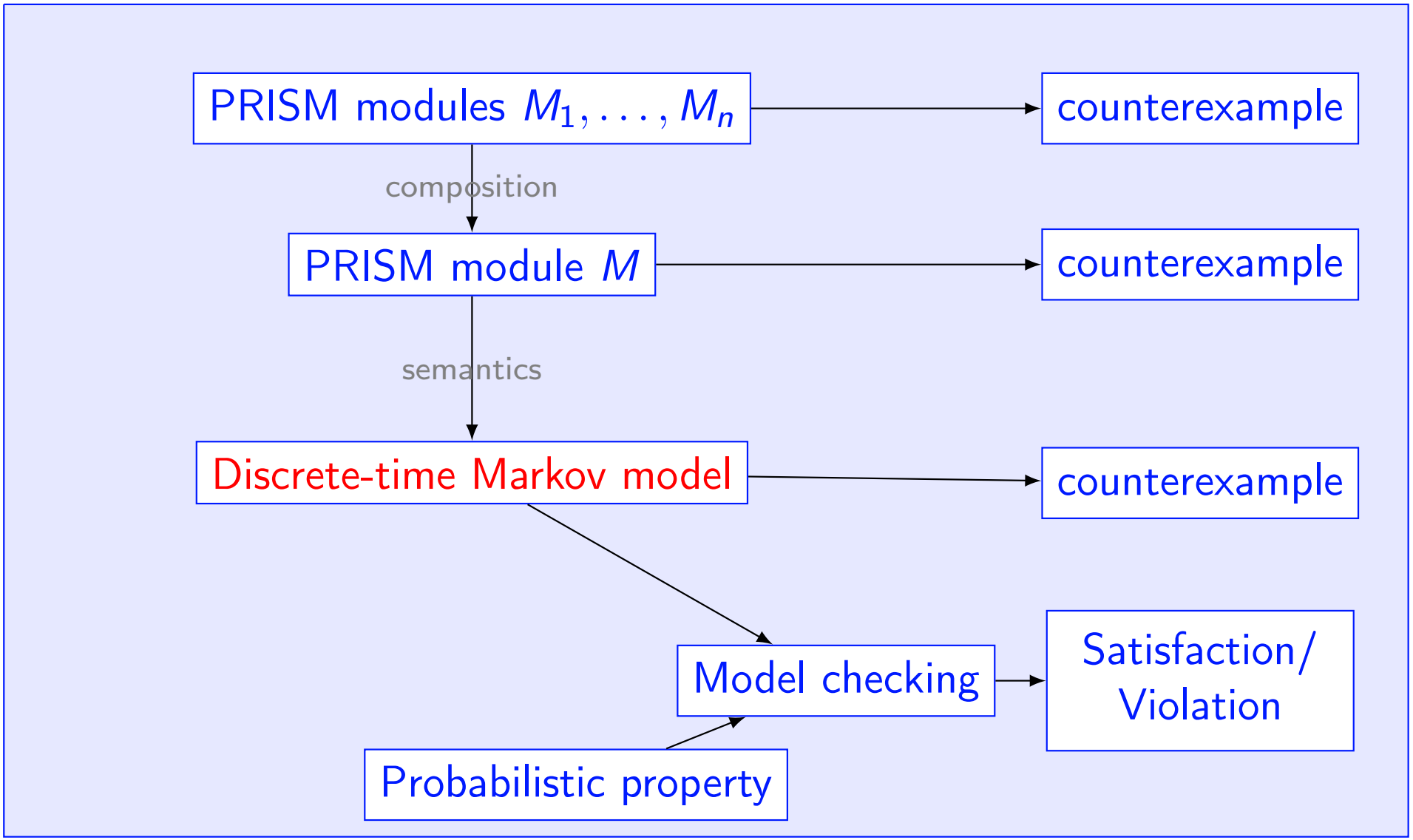
Scheduler

- The non-determinism is resolved by a scheduler.
- It assigns to each finite path a distribution over the action-distribution pairs possible in the last state.
- A deterministic scheduler puts the whole probability into a unique action-distribution pair.
- The decisions made by a memoryless scheduler depend only on the last state of the path.

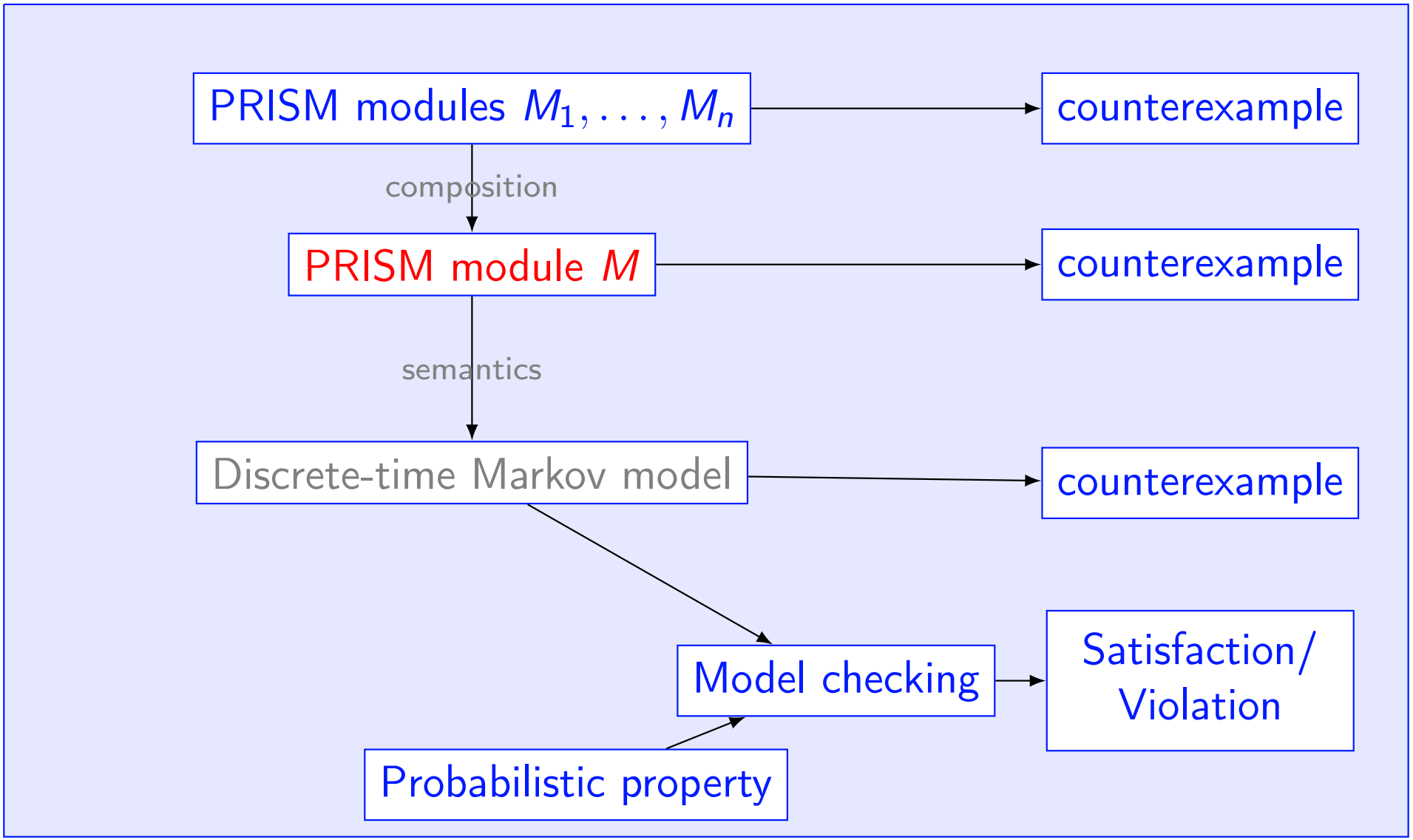


Each scheduler for an MDP/PA induces a DTMC.

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Example PRISM module

---

module `coin` $\|$ `processor`

f: **bool** **init** 0; c: **bool** **init** 0; p: **bool** **init** 0;

$[\tau]\ \neg f \rightarrow 0.5 : (f' = 1) \,\&\, (c' = 1) + 0.5 : (f' = 1) \,\&\, (c' = 0);$

$[\texttt{reset}]\ f \wedge \neg c \rightarrow 1 : (f' = 0) \,\&\, (p' = 0);$

$[\texttt{proc}]\ f \wedge \neg p \rightarrow 0.99 : (f' = 1) \,\&\, (p' = 1) + 0.01 : (c' = 1) \,\&\, (p' = 1);$

$[\tau]\ p \rightarrow 1 : (p' = 1);$

endmodule

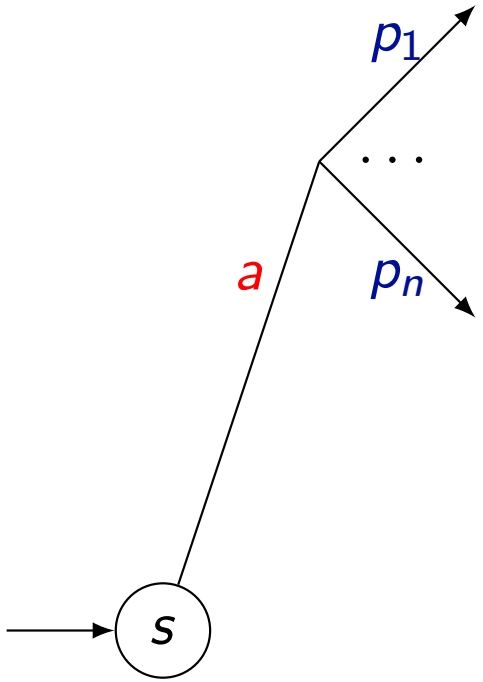
The semantics of PRISM modules

The semantics of PRISM modules

$$c_1 = [a]\ b_1 \rightarrow p_1 : f_1 + \ldots + p_n : f_n$$

The semantics of PRISM modules

$$c_1 = [a]\ b_1 \rightarrow p_1 : f_1 + \ldots + p_n : f_n$$

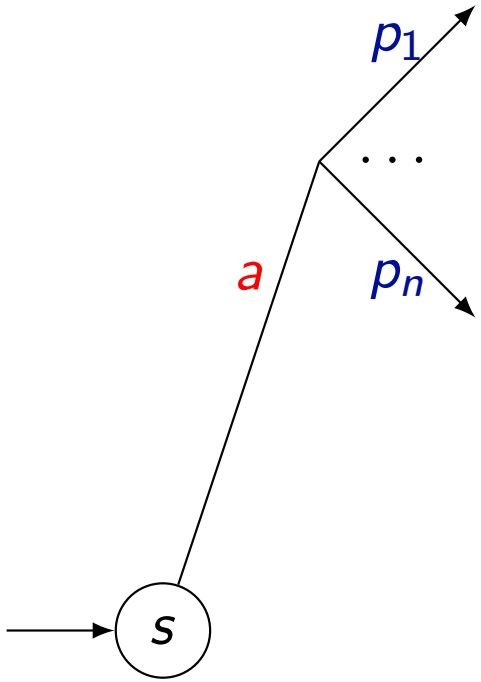


DTMC

The semantics of PRISM modules

$$c_1 = [a]\ b_1 \to p_1 : f_1 + \ldots + p_n : f_n$$

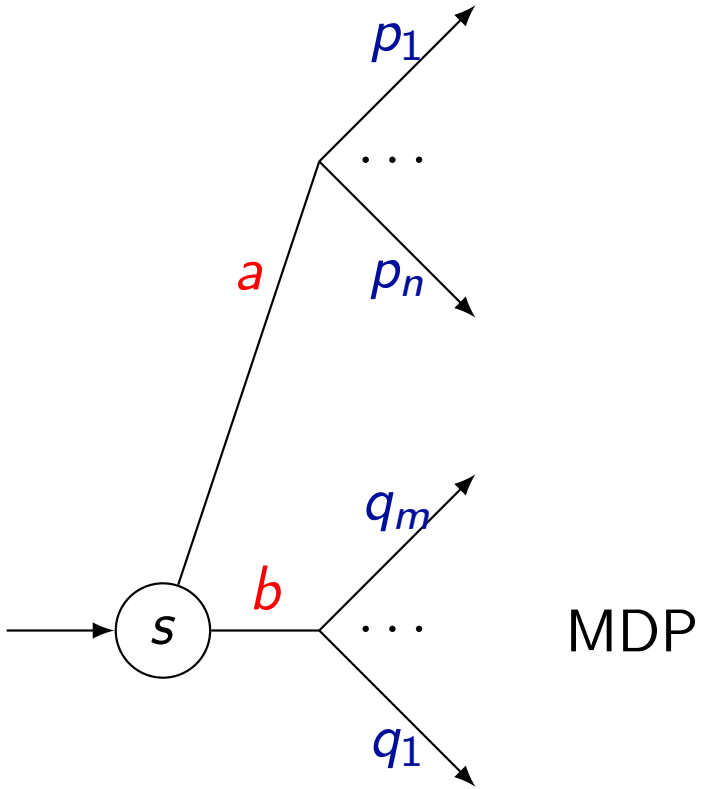
$$c_2 = [b]\ b_2 \to q_1 : g_1 + \ldots + q_m : g_m$$



The semantics of PRISM modules

$$c_1 = [a]\ b_1 \to p_1 : f_1 + \ldots + p_n : f_n$$

$$c_2 = [b]\ b_2 \to q_1 : g_1 + \ldots + q_m : g_m$$



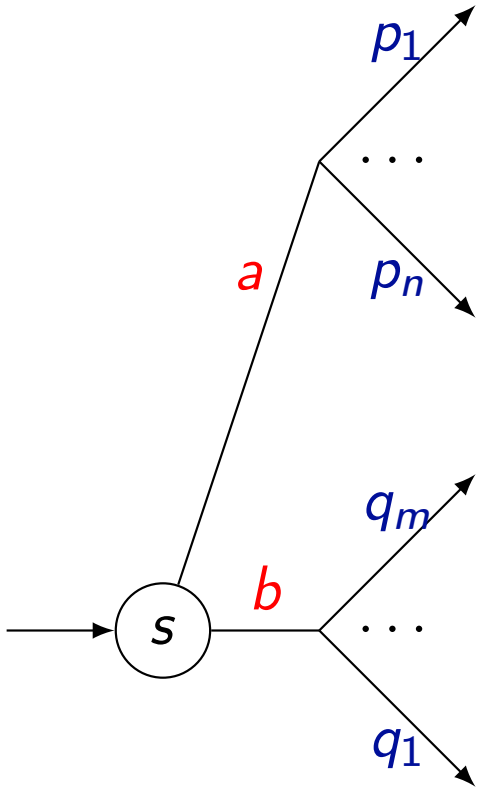
MDP

The semantics of PRISM modules

$c_1 = [a]\ b_1 \to p_1 : f_1 + \ldots + p_n : f_n$

$c_2 = [b]\ b_2 \to q_1 : g_1 + \ldots + q_m : g_m$

$c_3 = [a]\ b_3 \to r_1 : h_1 + \ldots + r_k : h_k$

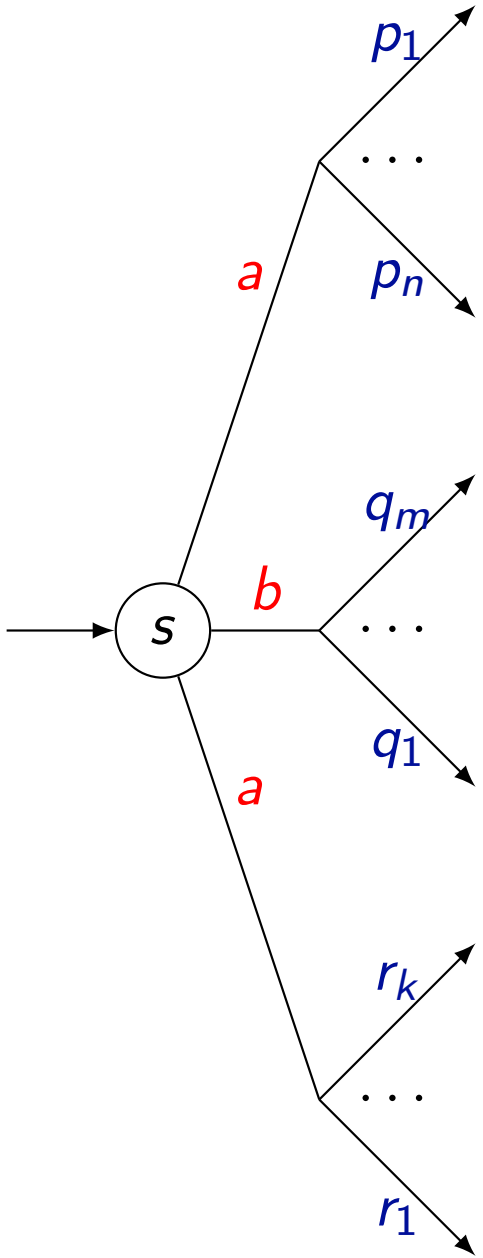


The semantics of PRISM modules

$c_1 = [a]\ b_1 \to p_1 : f_1 + \ldots + p_n : f_n$

$c_2 = [b]\ b_2 \to q_1 : g_1 + \ldots + q_m : g_m$

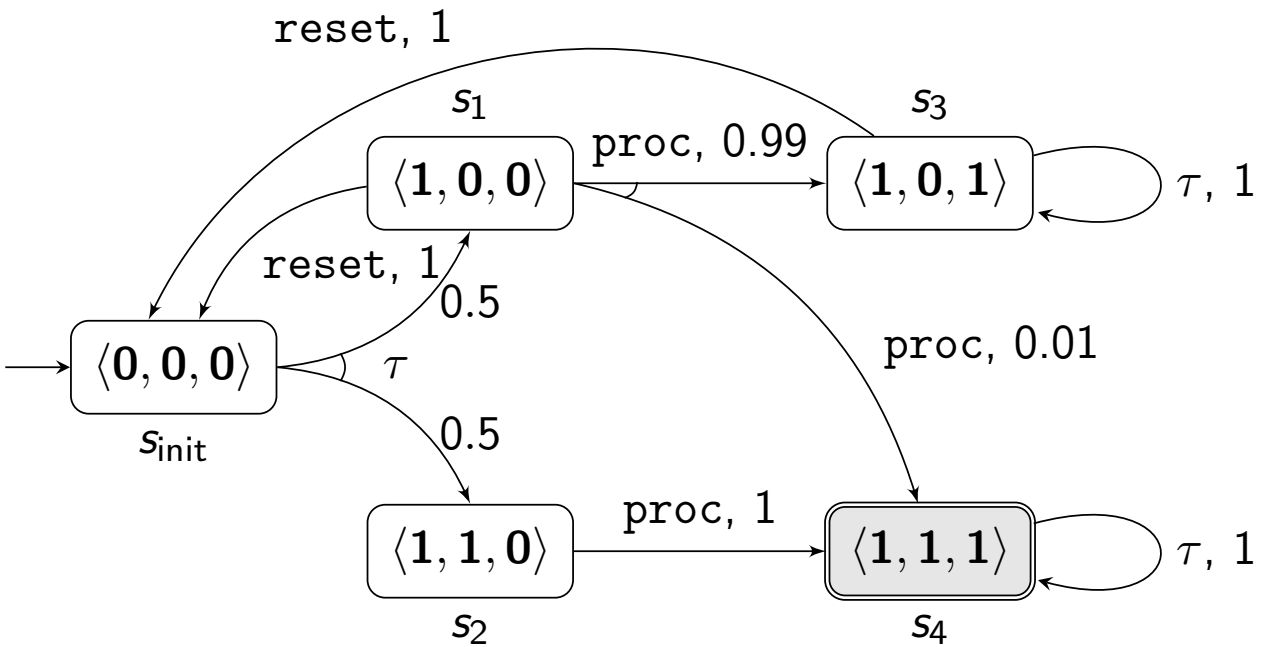
$c_3 = [a]\ b_3 \to r_1 : h_1 + \ldots + r_k : h_k$



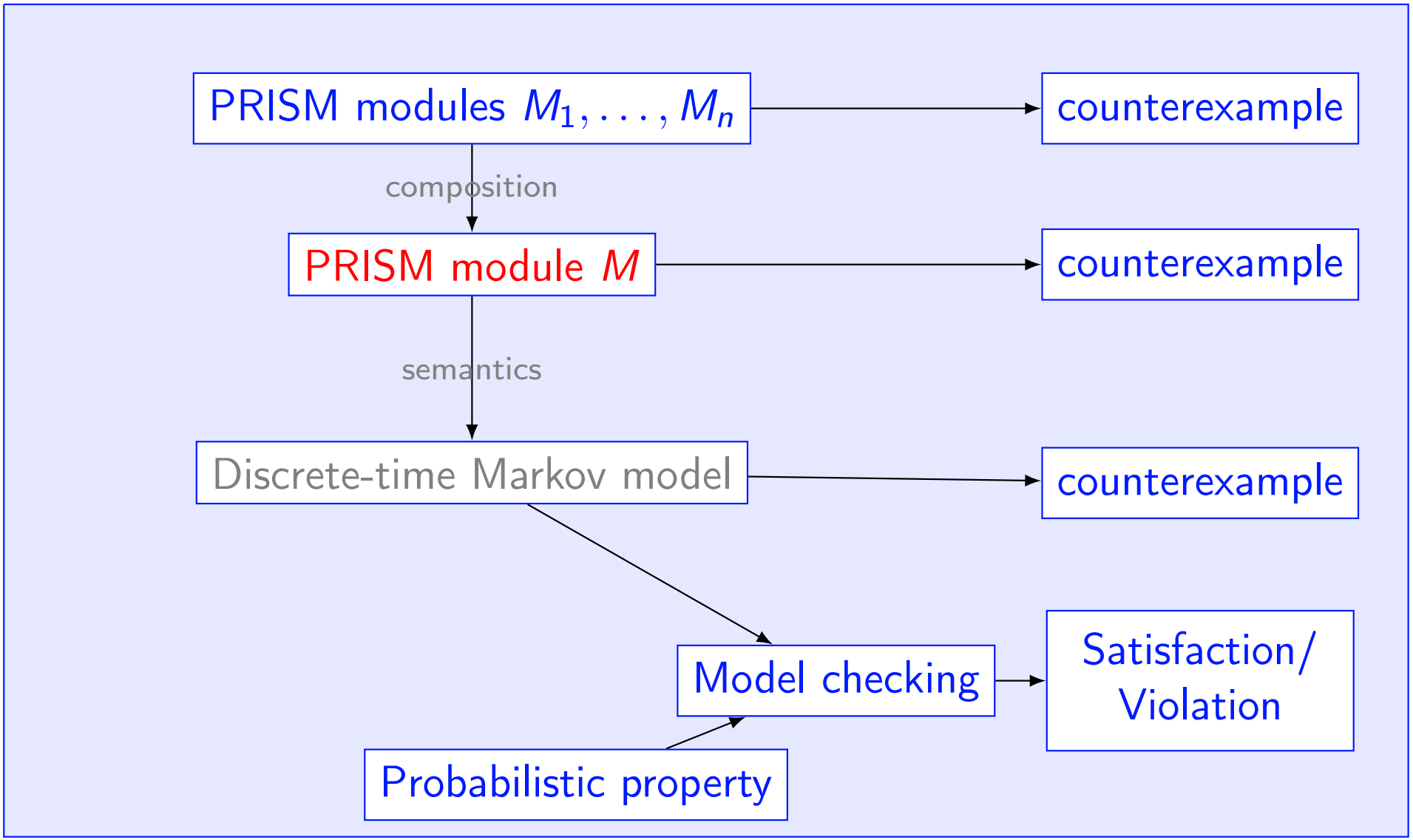
PA

Example PRISM module

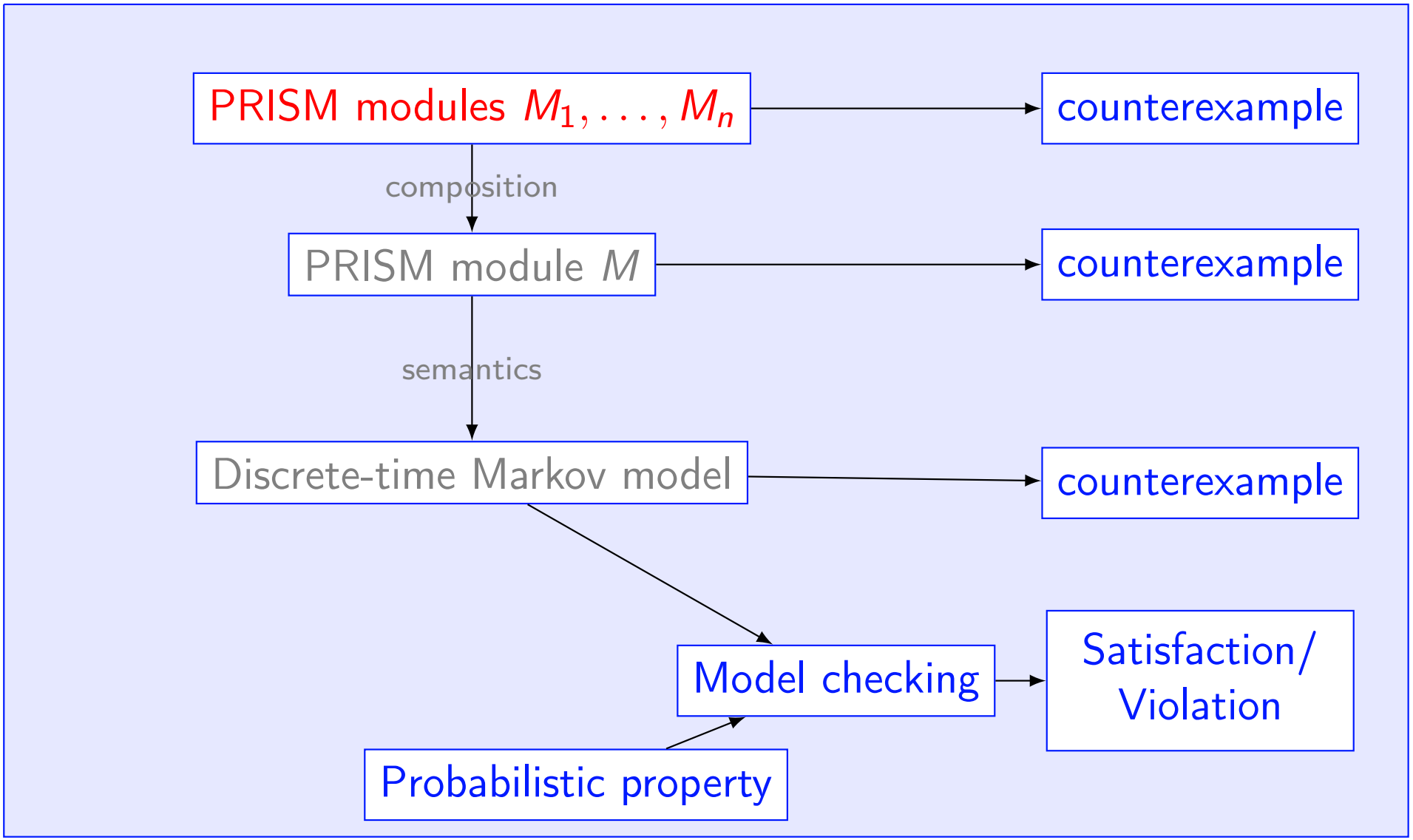
module coin $\|$ processor
f: **bool** **init** 0; c: **bool** **init** 0; p: **bool** **init** 0;
$[\tau]\ \neg f \to 0.5 : (f' = 1) \,\&\, (c' = 1) + 0.5 : (f' = 1) \,\&\, (c' = 0)$;
[reset] $f \wedge \neg c \to 1 : (f' = 0) \,\&\, (p' = 0)$;
[proc] $f \wedge \neg p \to 0.99 : (f' = 1) \,\&\, (p' = 1) + 0.01 : (c' = 1) \,\&\, (p' = 1)$;
$[\tau]\ p \to 1 : (p' = 1)$;
endmodule



Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Example PRISM module composition

---

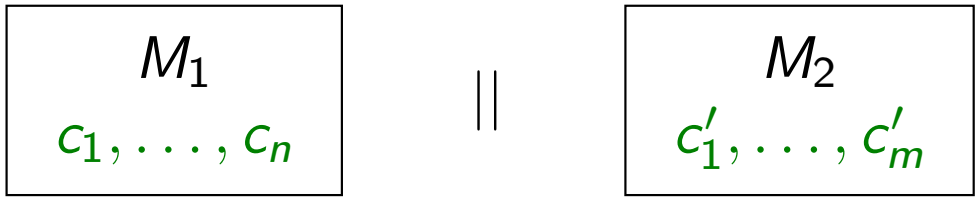
module coin
f: **bool** **init** 0; c: **bool** **init** 0;
$[\tau]\ \neg f \rightarrow 0.5 : (f' = 1)\,\&\,(c' = 1) + 0.5 : (f' = 1)\,\&\,(c' = 0)$;
[reset] $f \wedge \neg c \rightarrow 1 : (f' = 0)$;
[proc] $f \rightarrow 0.99 : (f' = 1) + 0.01 : (c' = 1)$;
endmodule
module processor
p: **bool** **init** 0;
[proc] $\neg p \rightarrow 1 : (p' = 1)$;
$[\tau]\ p \rightarrow 1 : (p' = 1)$;
[reset] $true \rightarrow 1 : (p' = 0)$
endmodule

---

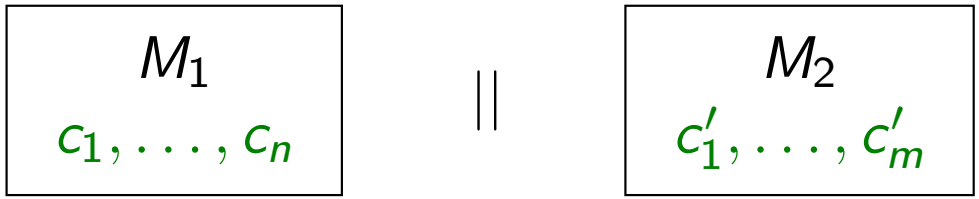
module coin $\|$ processor
f: **bool** **init** 0; c: **bool** **init** 0; p: **bool** **init** 0;
$[\tau]\ \neg f \rightarrow 0.5 : (f' = 1)\,\&\,(c' = 1) + 0.5 : (f' = 1)\,\&\,(c' = 0)$;
[reset] $f \wedge \neg c \rightarrow 1 : (f' = 0)\,\&\,(p' = 0)$;
[proc] $f \wedge \neg p \rightarrow 0.99 : (f' = 1)\,\&\,(p' = 1) + 0.01 : (c' = 1)\,\&\,(p' = 1)$;
$[\tau]\ p \rightarrow 1 : (p' = 1)$;
endmodule

The parallel composition of PRISM modules

The parallel composition of PRISM modules



The parallel composition of PRISM modules



$$c = [a]\ b \rightarrow \ldots + p_i : f_i + \ldots \qquad c' = [a]\ b' \rightarrow \ldots + p'_j : f'_j + \ldots$$

$$c \otimes c' = [a]\ b \wedge b' \rightarrow \ldots + p_i \cdot p'_j : f_i \otimes f'_j + \ldots$$

The parallel composition of PRISM modules

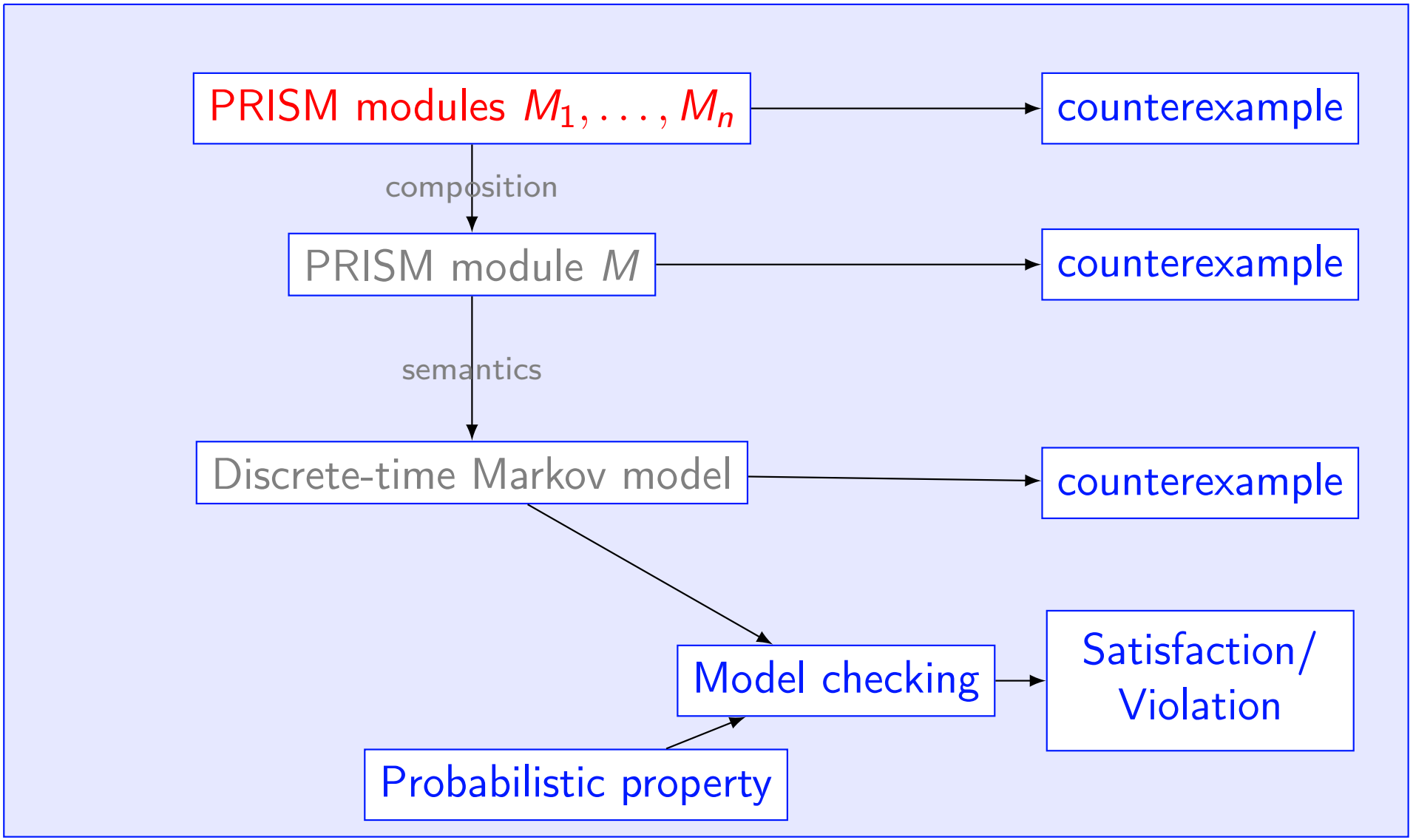
$$\boxed{\begin{array}{c} M_1 \\ c_1, \ldots, c_n \end{array}} \quad || \quad \boxed{\begin{array}{c} M_2 \\ c'_1, \ldots, c'_m \end{array}}$$

$$\boxed{\begin{array}{c} M_1 || M_2 \\ c_i \otimes c'_j \quad \textit{for synchronizing commands} \\ c_k, \; c'_l \quad \textit{for non-synchronizing commands} \end{array}}$$

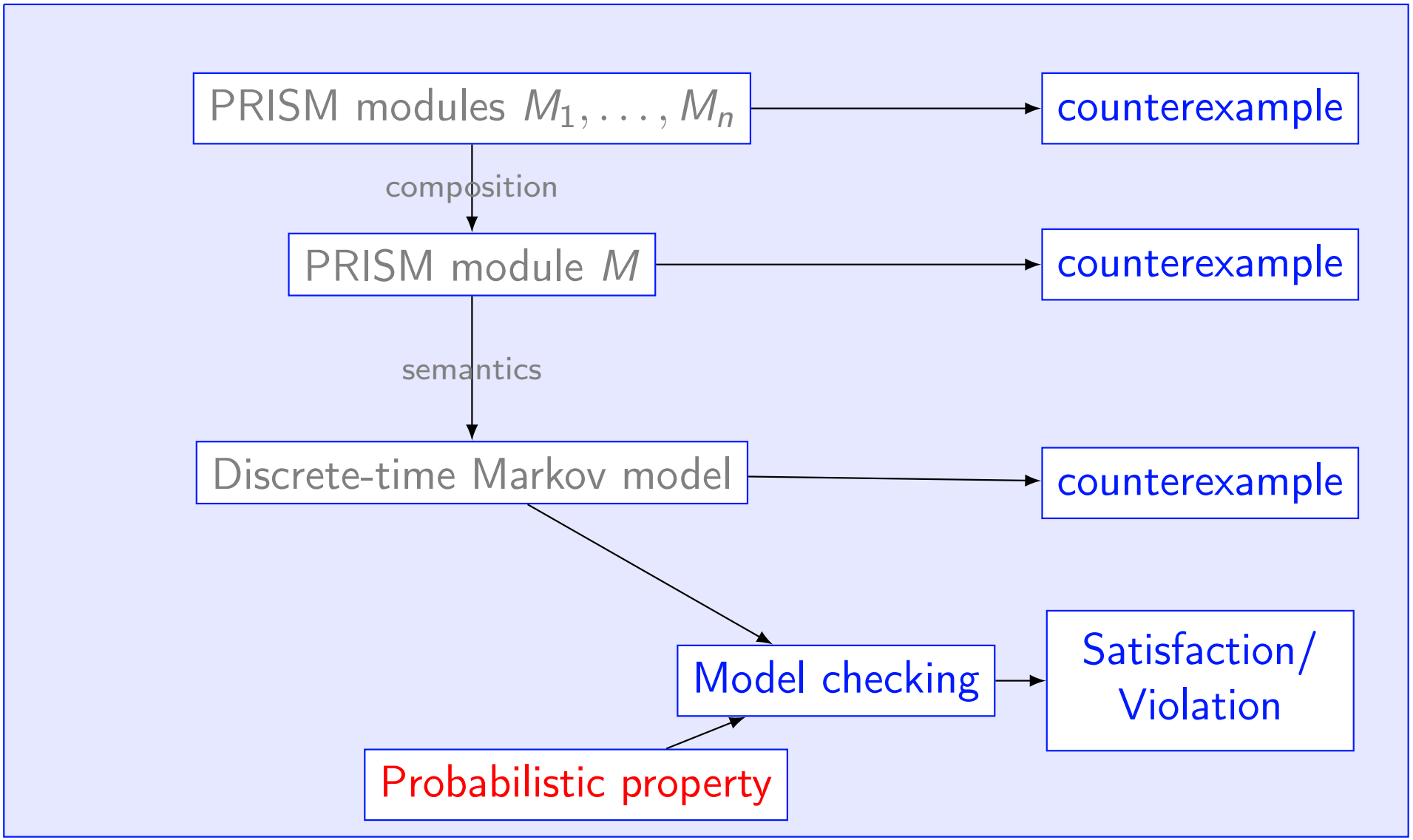
$$c = [a]\ b \to \ldots + p_i : f_i + \ldots \qquad c' = [a]\ b' \to \ldots + p'_j : f'_j + \ldots$$

$$c \otimes c' = [a]\ b \wedge b' \to \ldots + p_i \cdot p'_j : f_i \otimes f'_j + \ldots$$

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Probabilistic computation tree logic

Probabilistic computation tree logic

Probabilistic computation tree logic (PCTL) extends CTL with probabilities.

Probabilistic computation tree logic

Probabilistic computation tree logic (PCTL) extends CTL with probabilities.

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

with $a \in AP$ an atomic proposition, ψ a path formula and $J \subseteq [0,1]$ a non-empty real-valued interval.

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

with φ a state formula and $k \in \mathbb{N}$.

Probabilistic computation tree logic

Probabilistic computation tree logic (PCTL) extends CTL with probabilities.

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

with $a \in AP$ an atomic proposition, ψ a path formula and $J \subseteq [0,1]$ a non-empty real-valued interval.

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \mathcal{U} \varphi \mid \varphi \mathcal{U}^{\leq k} \varphi$$

with φ a state formula and $k \in \mathbb{N}$.

Syntactic sugar:

$$\begin{aligned} \Diamond\varphi &= \mathit{true}\mathcal{U}\,\varphi \\ \mathbb{P}_{\leq p}(\Box\varphi) &= \mathbb{P}_{>1-p}(\Diamond\neg\varphi) \end{aligned}$$

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_{J}(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J
- $\mathcal{D}, \pi \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ if

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J
- $\mathcal{D}, \pi \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ if φ_1 holds until φ_2 holds within k steps

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J
- $\mathcal{D}, \pi \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ if φ_1 holds until φ_2 holds within k steps
- $\mathcal{D} \models \varphi$ if

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J
- $\mathcal{D}, \pi \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ if φ_1 holds until φ_2 holds within k steps
- $\mathcal{D} \models \varphi$ if $\mathcal{D}, s_{\text{init}} \models \varphi$

Probabilistic computation tree logic

Definition (PCTL syntax)

- State formulas:

$$\varphi \ ::= \ \mathit{true} \mid a \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbb{P}_J(\psi)$$

- Path formulas:

$$\psi \ ::= \ \bigcirc\varphi \mid \varphi \,\mathcal{U}\, \varphi \mid \varphi \,\mathcal{U}^{\leq k}\, \varphi$$

Definition (PCTL semantics)

- $\mathcal{D}, s \models \mathbb{P}_J(\psi)$ if the probability that paths starting in s fulfill ψ lies in J
- $\mathcal{D}, \pi \models \varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2$ if φ_1 holds until φ_2 holds within k steps
- $\mathcal{D} \models \varphi$ if $\mathcal{D}, s_{\text{init}} \models \varphi$

NB: The satisfaction sets of PCTL path formulas are measurable.

Probabilistic reachability properties

- Model checking PCTL formulas for DTMCs can be reduced to check probabilistic reachability properties.
- E.g., $\mathcal{D}, s_{\text{init}} \models \mathbb{P}_{<0.01}(\Diamond t)$

  "The probability to reach t from s_{init} in $\mathcal{D}$ is less than 0.01."

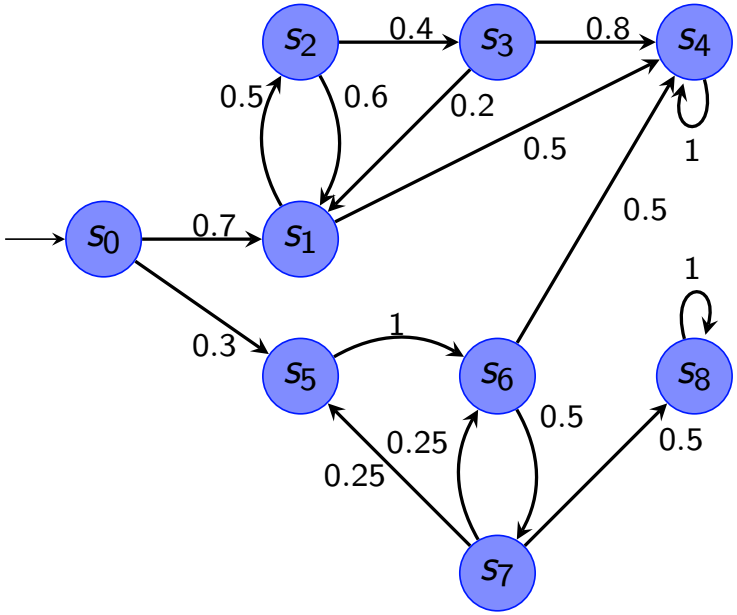
Probabilistic reachability properties

- Model checking PCTL formulas for DTMCs can be reduced to check probabilistic reachability properties.
- E.g., $\mathcal{D}, s_{\text{init}} \models \mathbb{P}_{<0.01}(\Diamond t)$

  "The probability to reach t from s_{init} in $\mathcal{D}$ is less than 0.01."

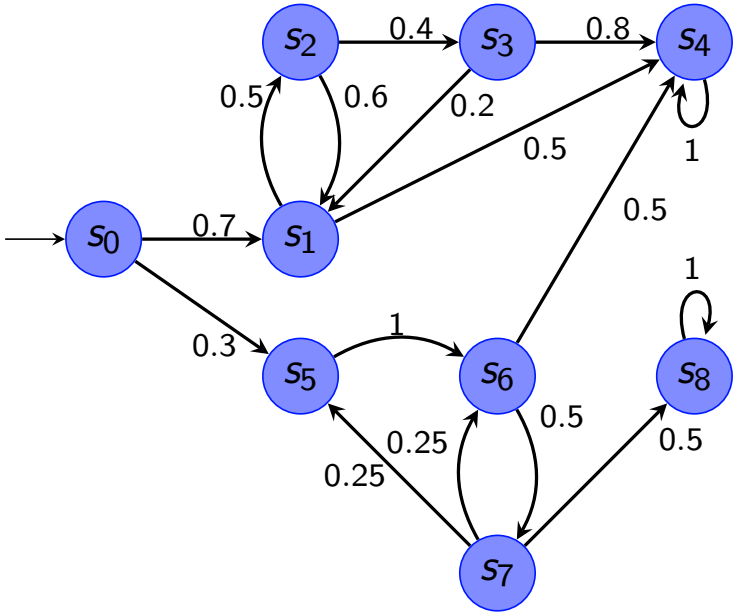
$$
\begin{aligned}
& Pr_{s_{\text{init}}}(\Diamond t) \\
= \;& Pr_{s_{\text{init}}}\big(\{\textit{all infinite paths starting in } s_{\text{init}} \textit{ and visiting } t\}\big) \\
= \;& Pr_{s_{\text{init}}}\big(\cup_{\textit{finite paths } s_{\text{init}} \ldots t} \text{Cyl}(s_{\text{init}} \ldots t)\big) \\
= \;& \sum_{\textit{finite paths } s_0 \ldots s_n \textit{ with } s_0 = s_{\text{init}},\, s_n = t,\, t \notin \{s_0, \ldots, s_{n-1}\}} \prod_{i=0}^{n-1} P(s_i, s_{i+1}) \,.
\end{aligned}
$$

Reachability probabilities



$$Pr_{s_0}(\Diamond s_4) \quad =$$

Reachability probabilities

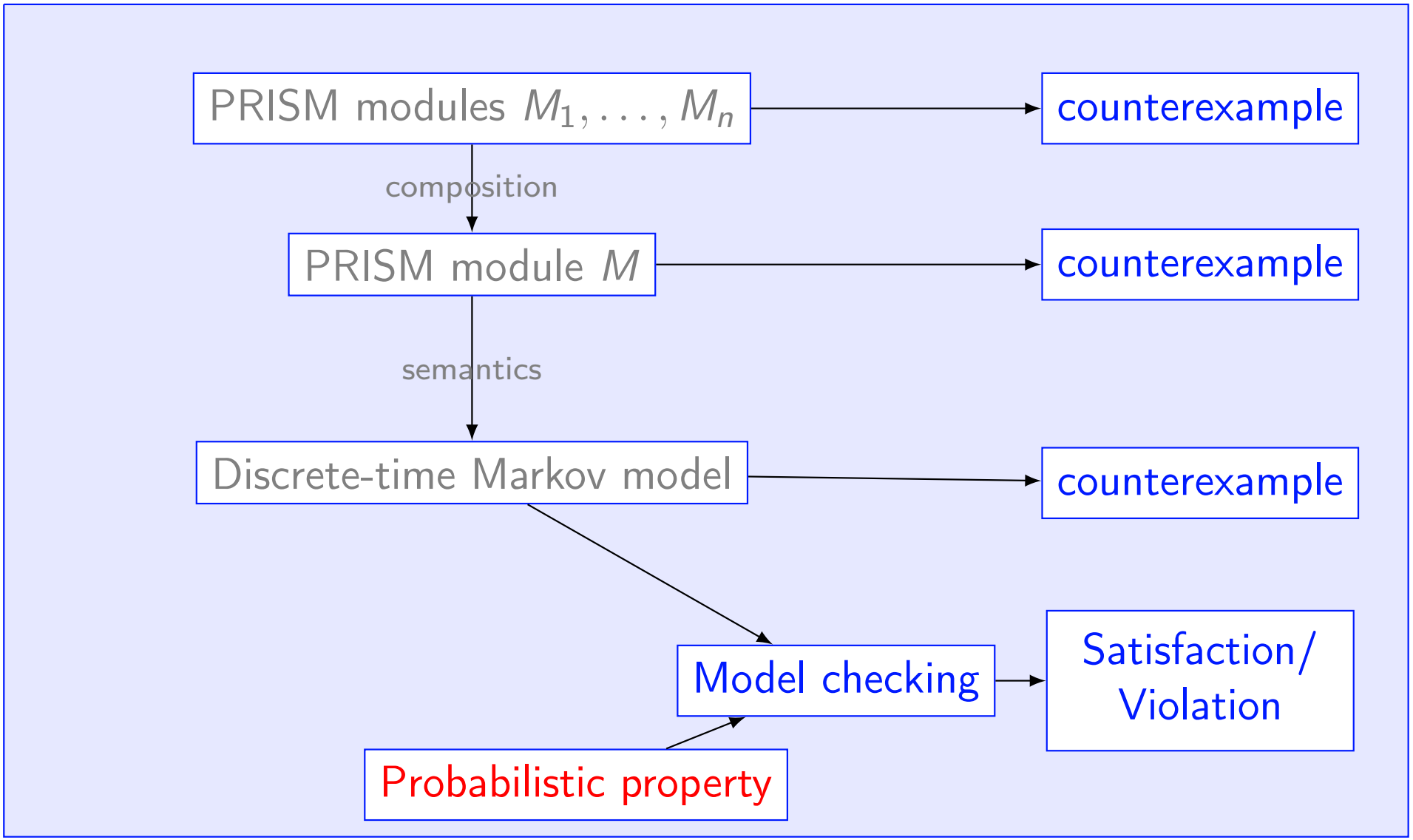


$$Pr_{s_0}(\Diamond s_4) = Pr_{s_0}(s_0 s_1 s_4) + Pr_{s_0}(s_0 s_1 s_2 s_1 s_4) + Pr_{s_0}(s_0 s_1 s_2 s_3 s_4) + Pr_{s_0}(s_0 s_5 s_6 s_4) + \ldots$$

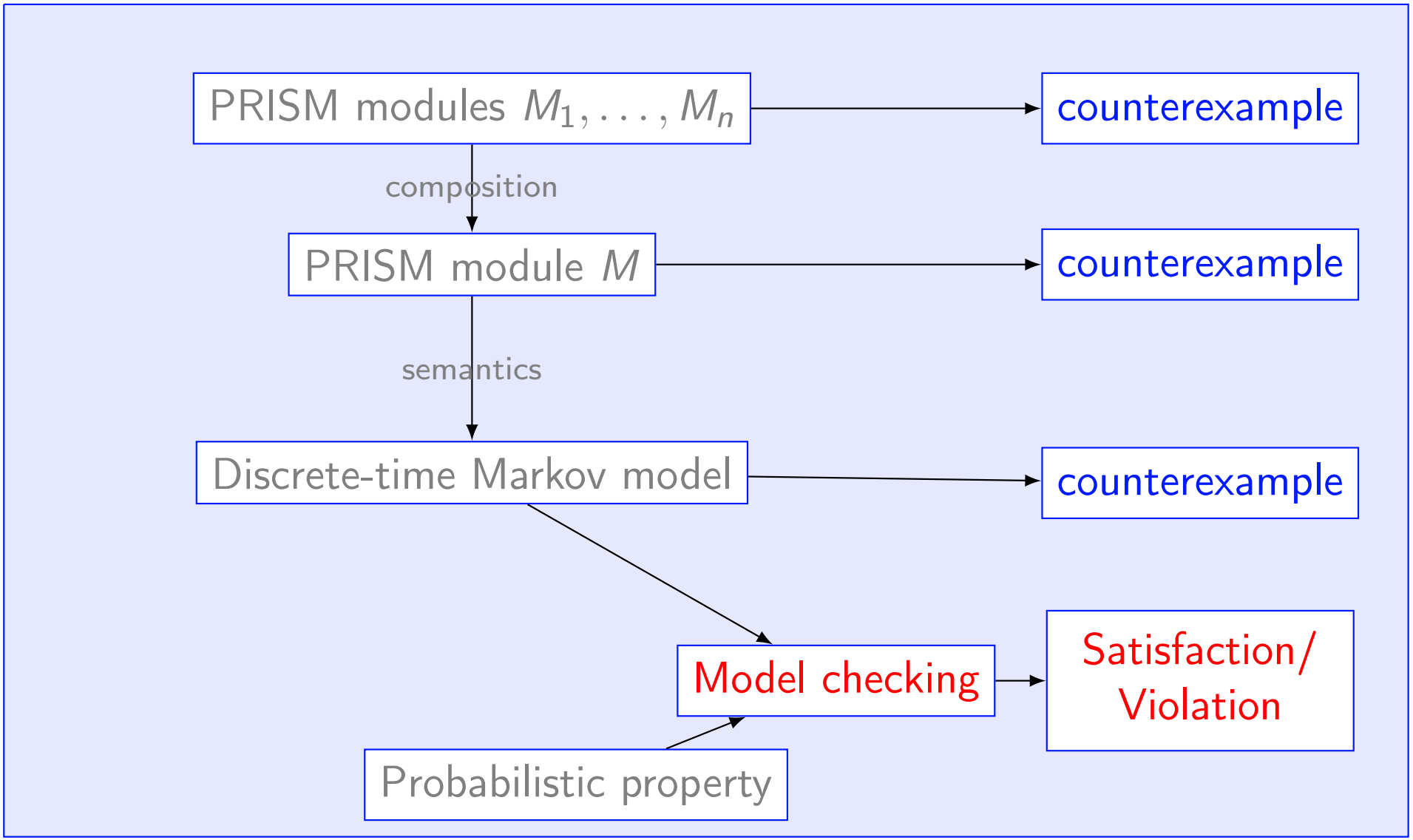
Reachability properties of MDPs and PAs

- A probabilistic reachability property is satisfied by an MDP/PA if it is satisfied by the induced DTMCs of all schedulers.
- It is sufficient to check memoryless deterministic schedulers.

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

PCTL model checking for DTMCs

Recursively determine the satisfaction sets for sub-formulas inside-out:

- a:
- $\varphi_1 \wedge \varphi_1$:
- $\neg\varphi$:
- $\mathbb{P}_J(\bigcirc\varphi)$:

PCTL model checking for DTMCs

Recursively determine the satisfaction sets for sub-formulas inside-out:

- a: $Sat(a) = \{s \in S \mid a \in L(s)\}$
- $\varphi_1 \wedge \varphi_1$: $Sat(\varphi_1 \wedge \varphi_2) = Sat(\varphi_1) \cap Sat(\varphi_2)$
- $\neg\varphi$: $Sat(\neg\varphi_1) = S \backslash Sat(\varphi)$
- $\mathbb{P}_J(\bigcirc\varphi)$: Via matrix multiplication
  - given characteristic vector $\underline{\varphi}$
    ($0/1$-vector over S, 1 meaning the satisfaction of φ)
  - compute $P \cdot \underline{\varphi}$ (probabilities to satisfy $\bigcirc\varphi$)
  - collect states with probabilities in J

PCTL model checking for DTMCs

- $\mathbb{P}_J(\varphi_1 \mathcal{U}^{\leq k} \varphi_2)$:

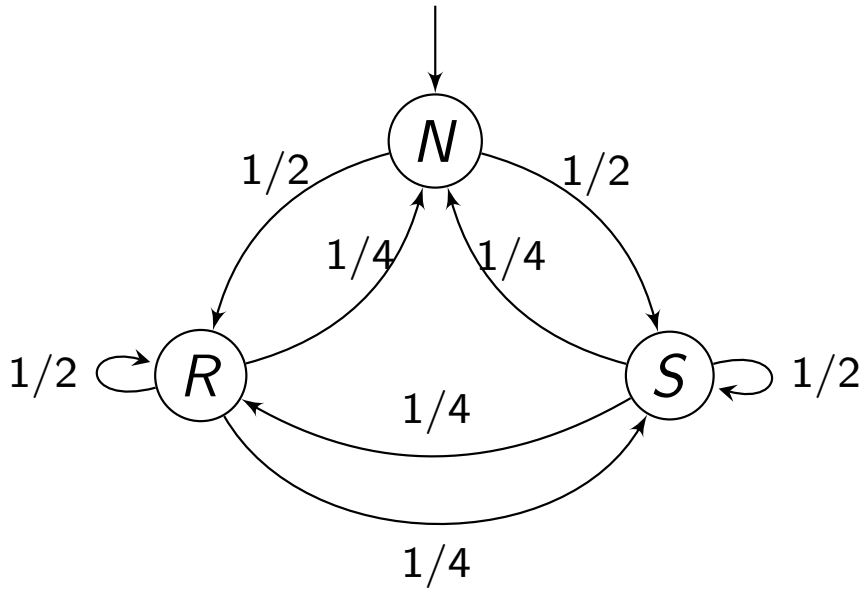
- $\mathbb{P}_J(\varphi_1 \mathcal{U} \varphi_2)$:

PCTL model checking for DTMCs

- $\mathbb{P}_J(\varphi_1 \,\mathcal{U}^{\leq k}\, \varphi_2)$: Via matrix multiplication
  - make states satisfying $\neg\varphi_1 \vee \varphi_2$ absorbing
  - remove states from which no φ_2-state is reachable
  - given characteristic vector $\underline{\varphi_2}$
  - compute $P^k \cdot \underline{\varphi_2}$
  - collect states with probabilities in J
- $\mathbb{P}_J(\varphi_1 \,\mathcal{U}\, \varphi_2)$: Via probabilistic reachability properties
  - make states satisfying $\neg\varphi_1 \vee \varphi_2$ absorbing
  - remove states from which no φ_2-state is reachable
  - compute reachability probabilities for $\Diamond\varphi_2$
  - collect states with probabilities in J

Weather properties in the Land of Oz

$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$



Model checking reachability properties of DTMCs

Model checking reachability properties of DTMCs

- S set of states, T set of absorbing target states
- States from which T is not reachable are irrelevant and get deleted.

Model checking reachability properties of DTMCs

- S set of states, T set of absorbing target states
- States from which T is not reachable are irrelevant and get deleted.

Probabilities of reaching T from $s \in S$

$$p_s = \begin{cases} 1 & \text{if } s \in T, \\ \sum_{s' \in S} P(s, s') \cdot p_{s'} & \text{otherwise.} \end{cases}$$

Model checking reachability properties of DTMCs

- S set of states, T set of absorbing target states
- States from which T is not reachable are irrelevant and get deleted.

Probabilities of reaching T from $s \in S$

$$p_s = \begin{cases} 1 & \text{if } s \in T, \\ \sum_{s' \in S} P(s, s') \cdot p_{s'} & \text{otherwise.} \end{cases}$$

- Option 1: Solve the linear equation system
- Option 2: Iterative approach to approximate the fixedpoint for $\mathit{probabilities} = P \cdot \mathit{probabilities}$
- Option 3: SCC-based model checking

Model checking reachability properties of DTMCs

- S set of states, T set of absorbing target states
- States from which T is not reachable are irrelevant and get deleted.

Probabilities of reaching T from $s \in S$

$$p_s = \begin{cases} 1 & \text{if } s \in T, \\ \sum_{s' \in S} P(s, s') \cdot p_{s'} & \text{otherwise.} \end{cases}$$

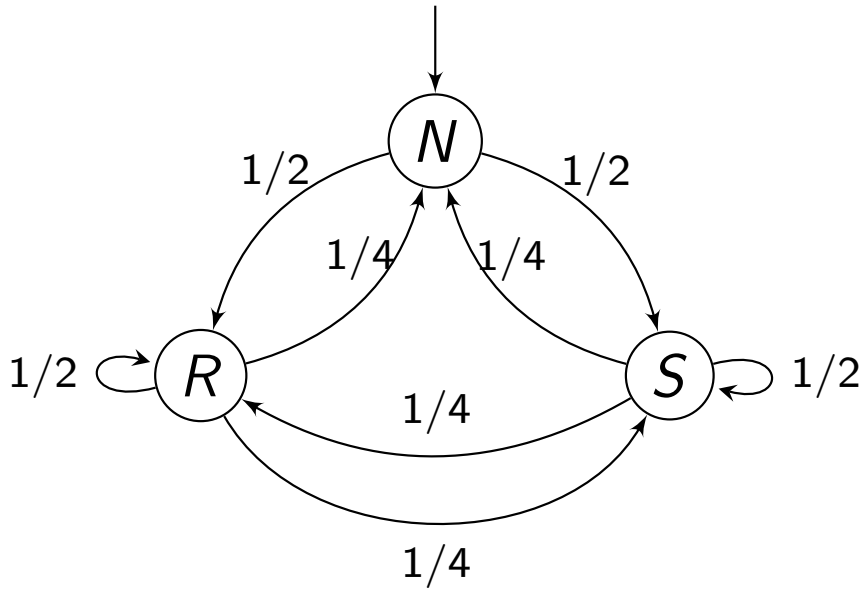
- Option 1: Solve the linear equation system
- Option 2: Iterative approach to approximate the fixedpoint for $\mathit{probabilities} = P \cdot \mathit{probabilities}$
- Option 3: SCC-based model checking

Probabilistic model checkers

- `PRISM` (Kwiatkowska, Norman, Parker)
- `MRMC` (Katoen, Zapreev, Hahn, Hermanns, Jansen (David N.))
- `Comics` (Jansen, Ábrahám, Wimmer, Katoen, Becker)

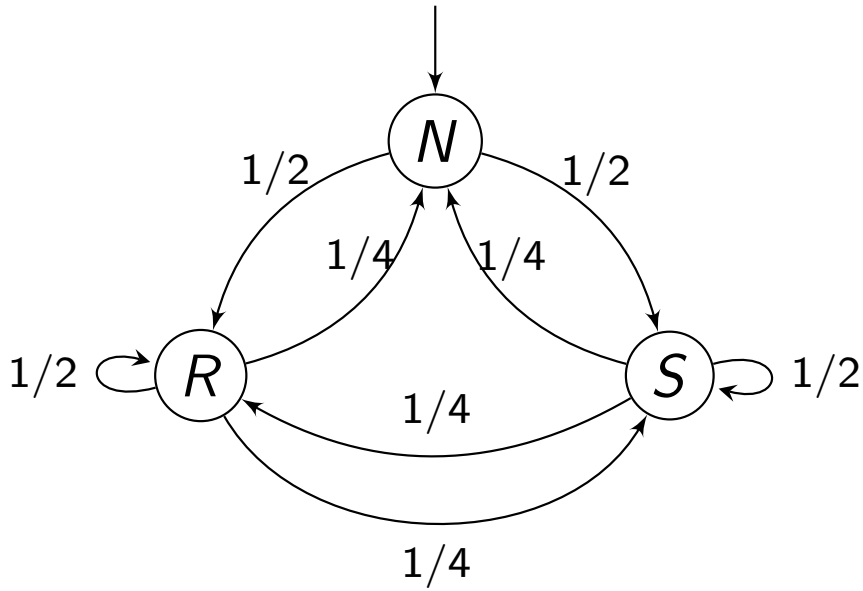
Option 1: Example

$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$

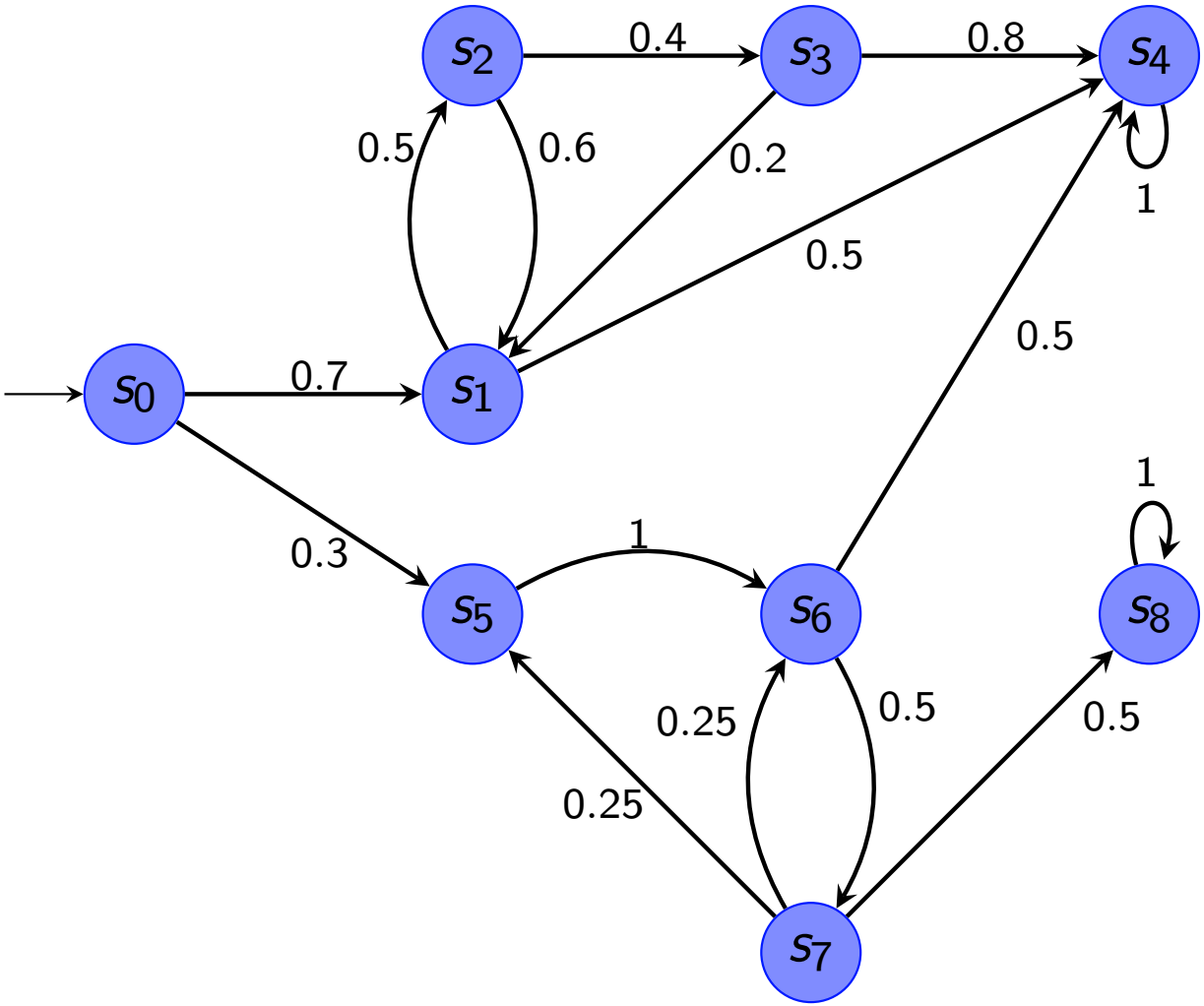


Option 2: Example

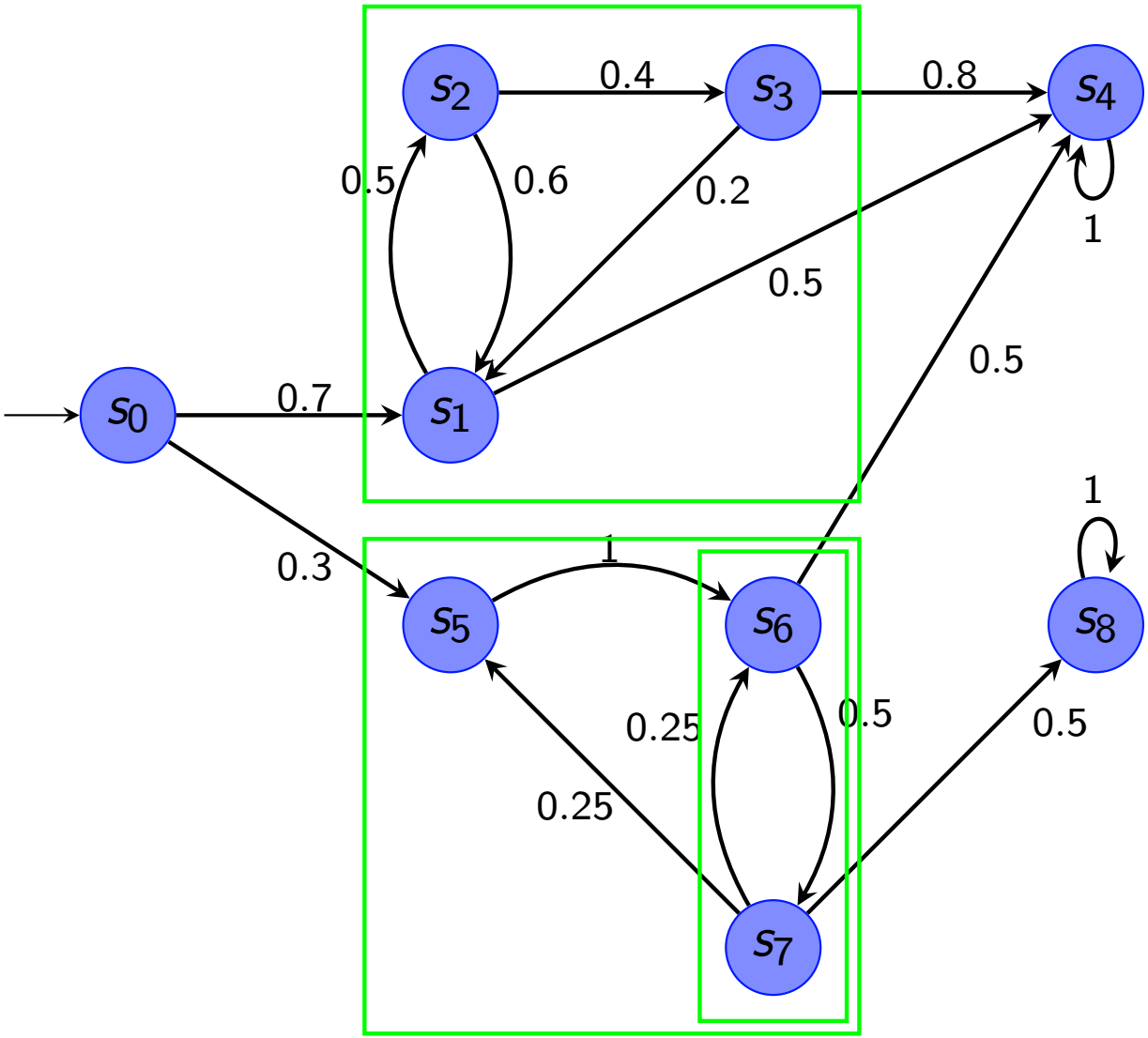
$$P = \begin{array}{c} \\ N \\ R \\ S \end{array} \begin{array}{c} \begin{array}{ccc} N & R & S \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$



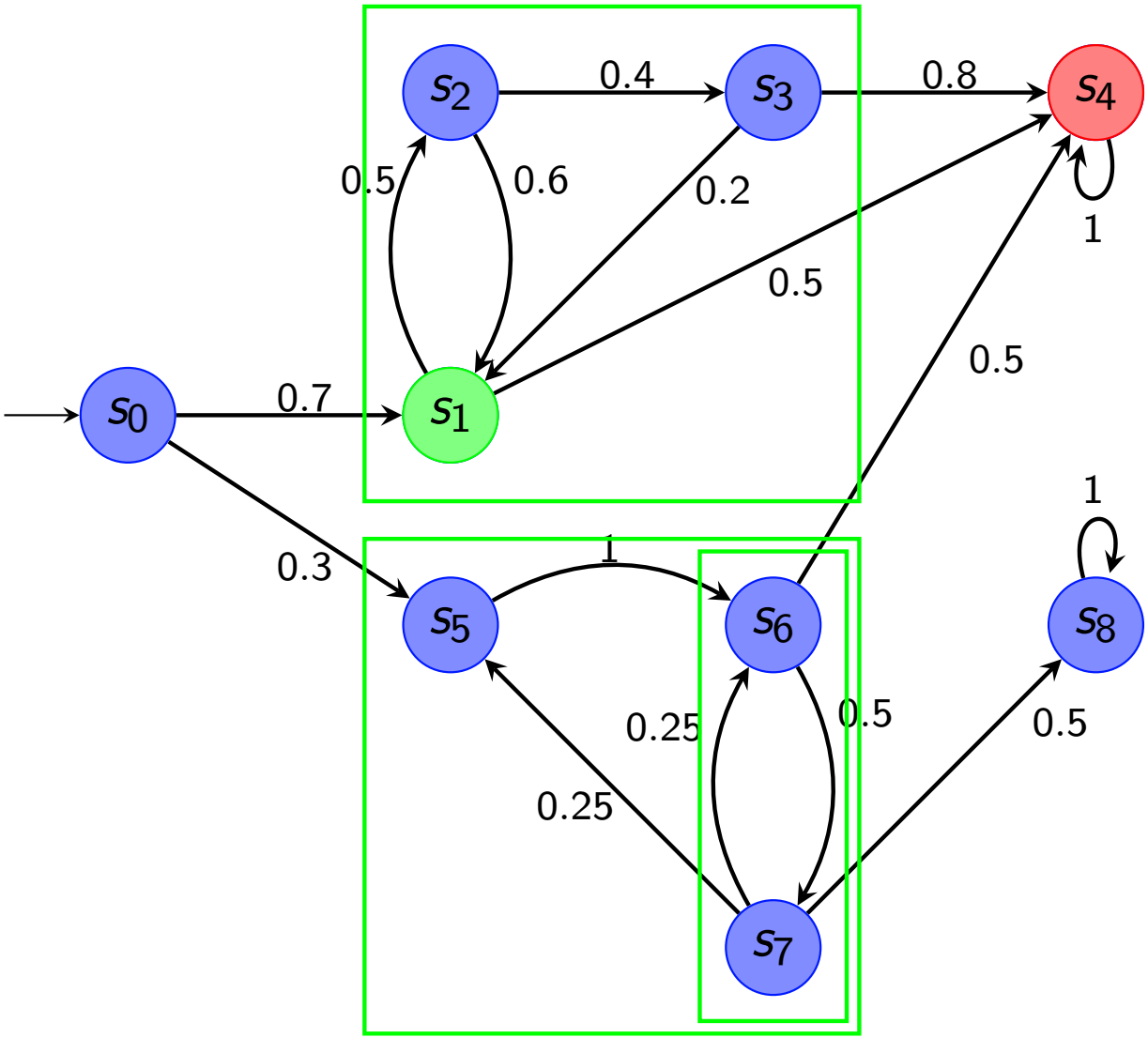
Option 3: SCC-based model checking for DTMCs



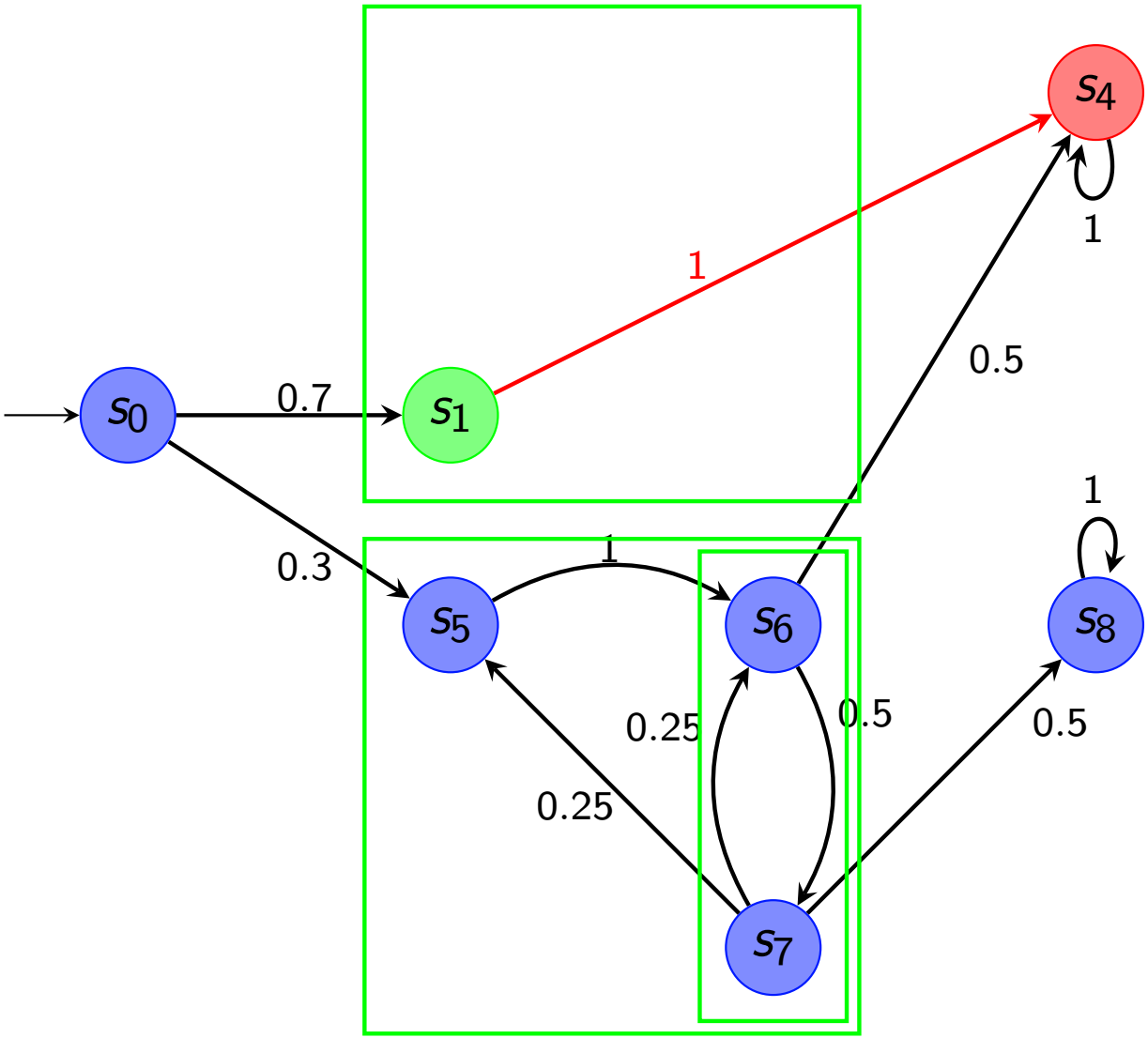
Option 3: SCC-based model checking for DTMCs



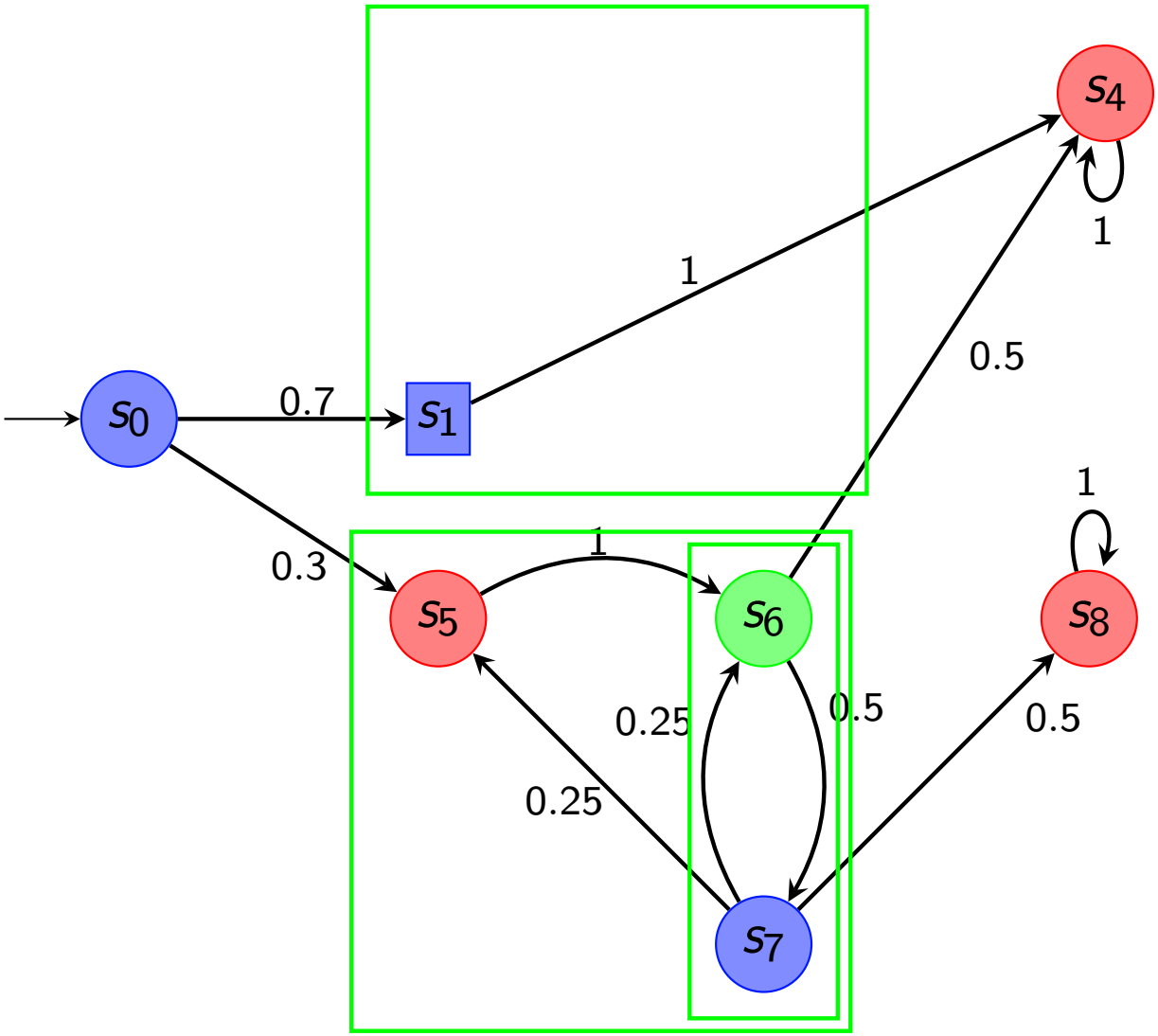
Option 3: SCC-based model checking for DTMCs



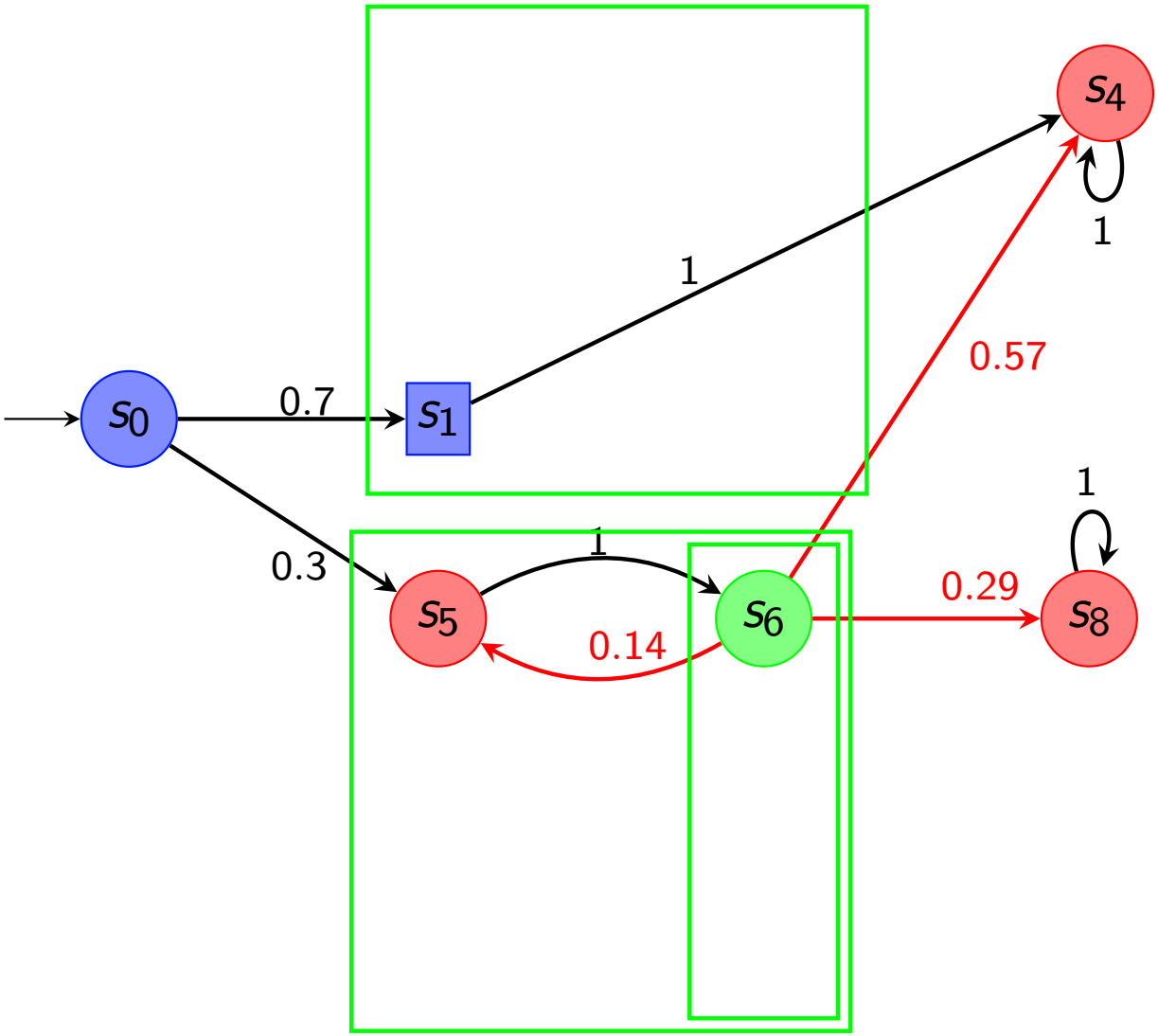
Option 3: SCC-based model checking for DTMCs



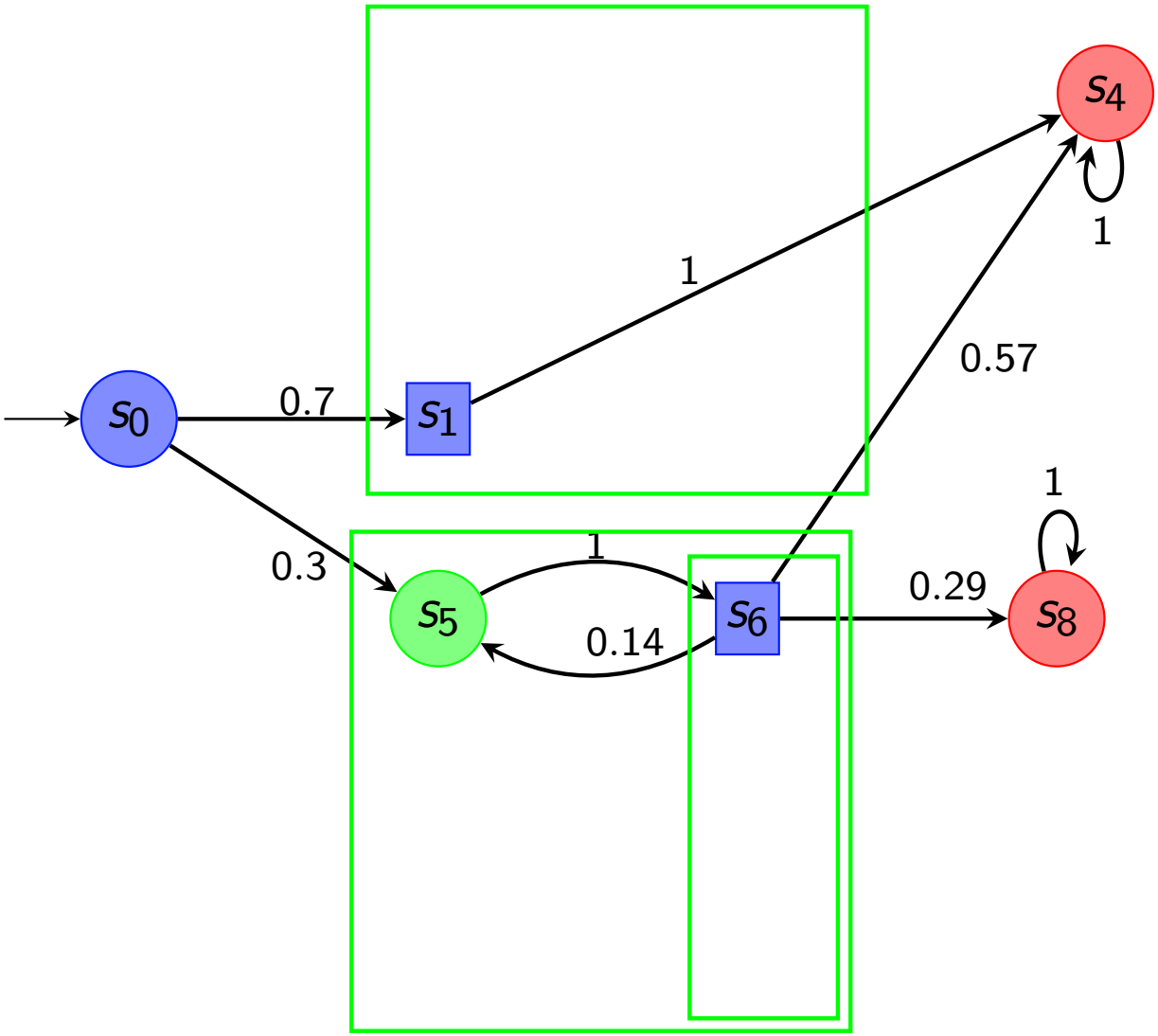
Option 3: SCC-based model checking for DTMCs



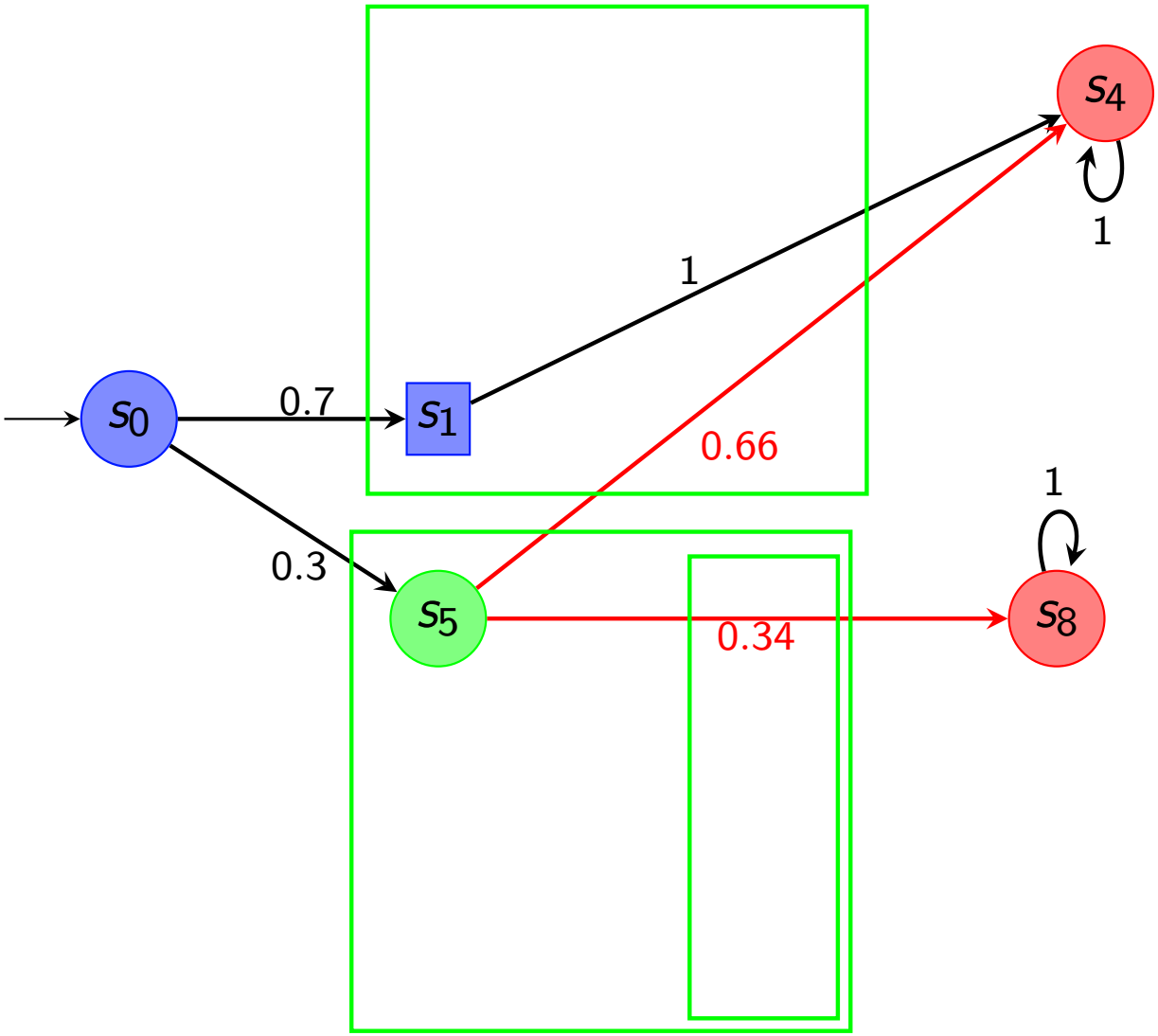
Option 3: SCC-based model checking for DTMCs



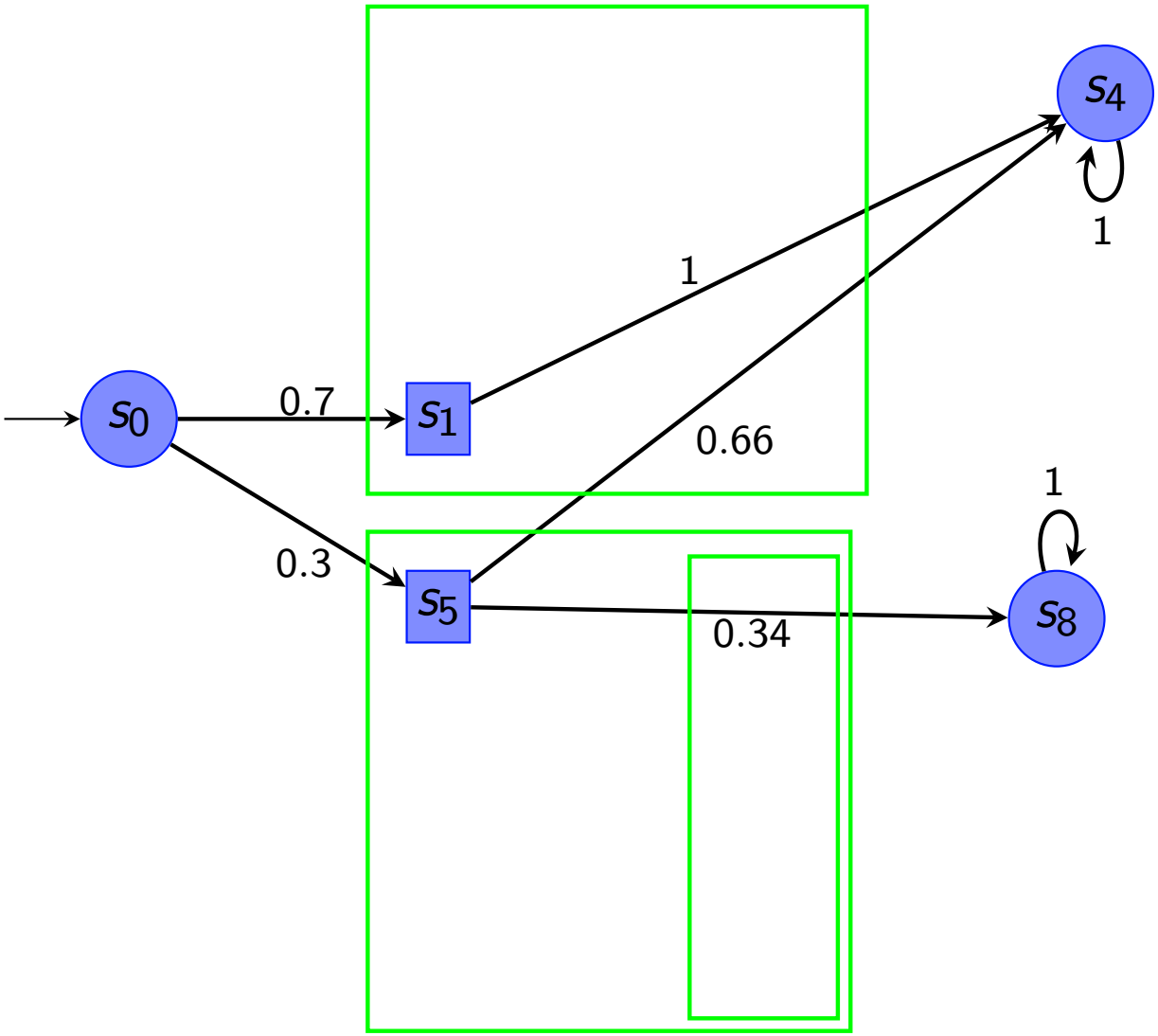
Option 3: SCC-based model checking for DTMCs



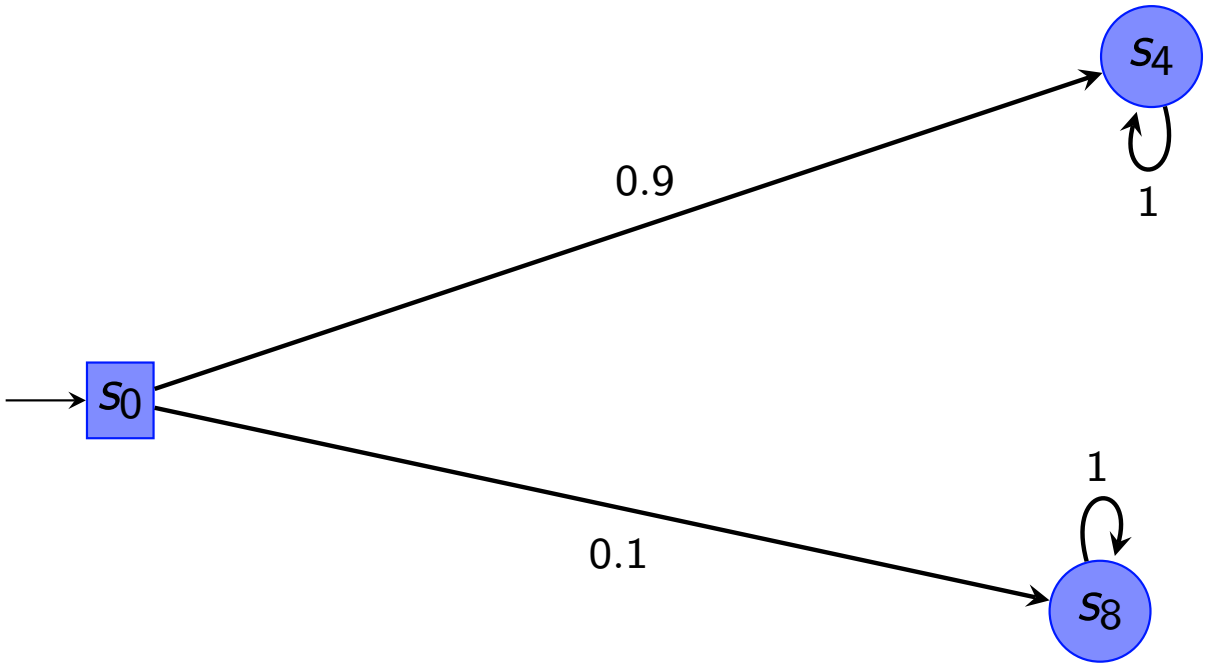
Option 3: SCC-based model checking for DTMCs



Option 3: SCC-based model checking for DTMCs



Option 3: SCC-based model checking for DTMCs



Model checking reachability properties of MDPs/PAs

- MDP/PA $\overset{scheduler}{\longrightarrow}$ DTMC

An MDP/PA satisfies a probabilistic reachability property if every DTMC induced by any scheduler satisfies it.

Model checking reachability properties of MDPs/PAs

- MDP/PA $\overset{scheduler}{\longrightarrow}$ DTMC

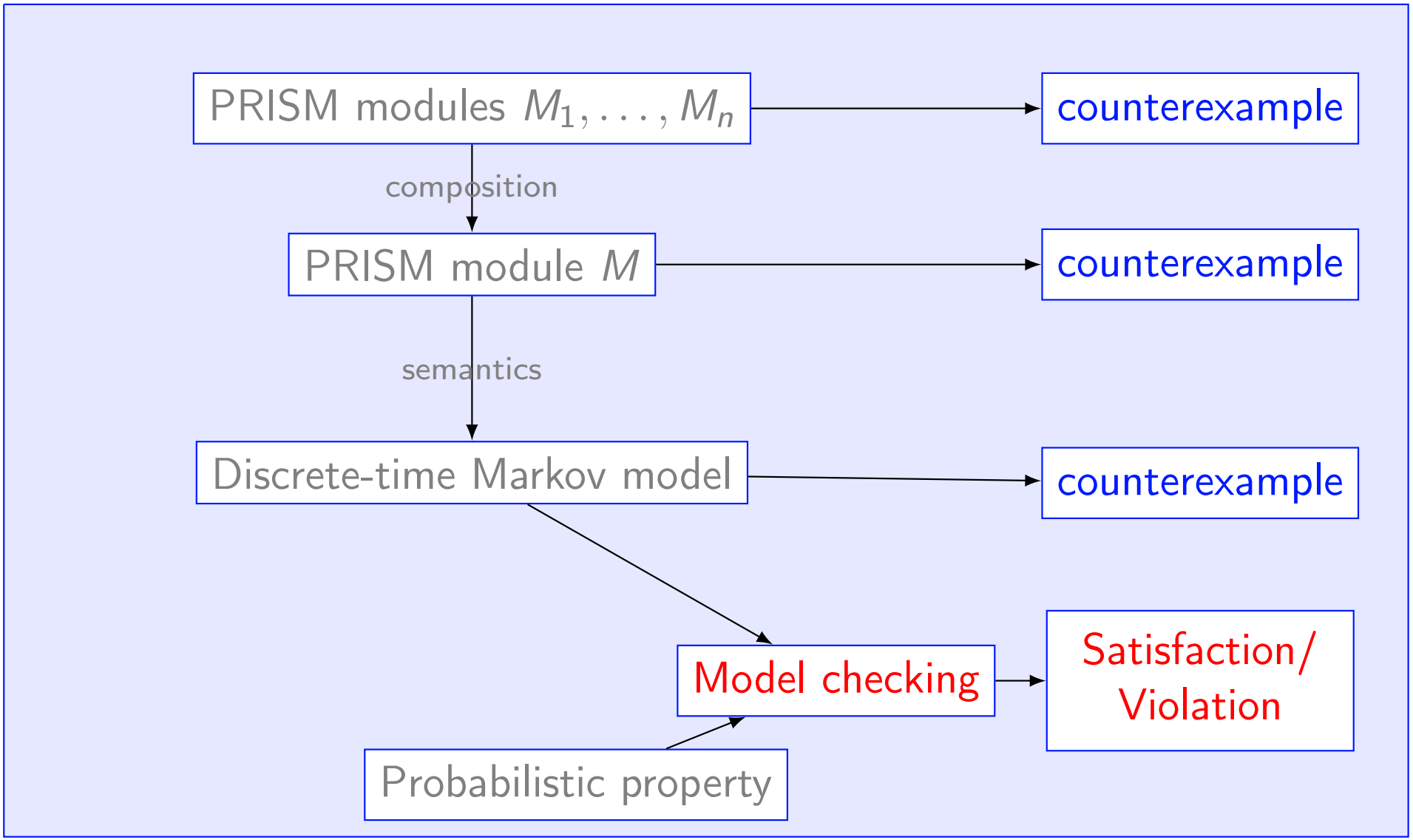
An MDP/PA satisfies a probabilistic reachability property if every DTMC induced by any scheduler satisfies it.

- Compute minimal/maximal scheduler under the memoryless deterministic ones
- Model check the induced DTMC

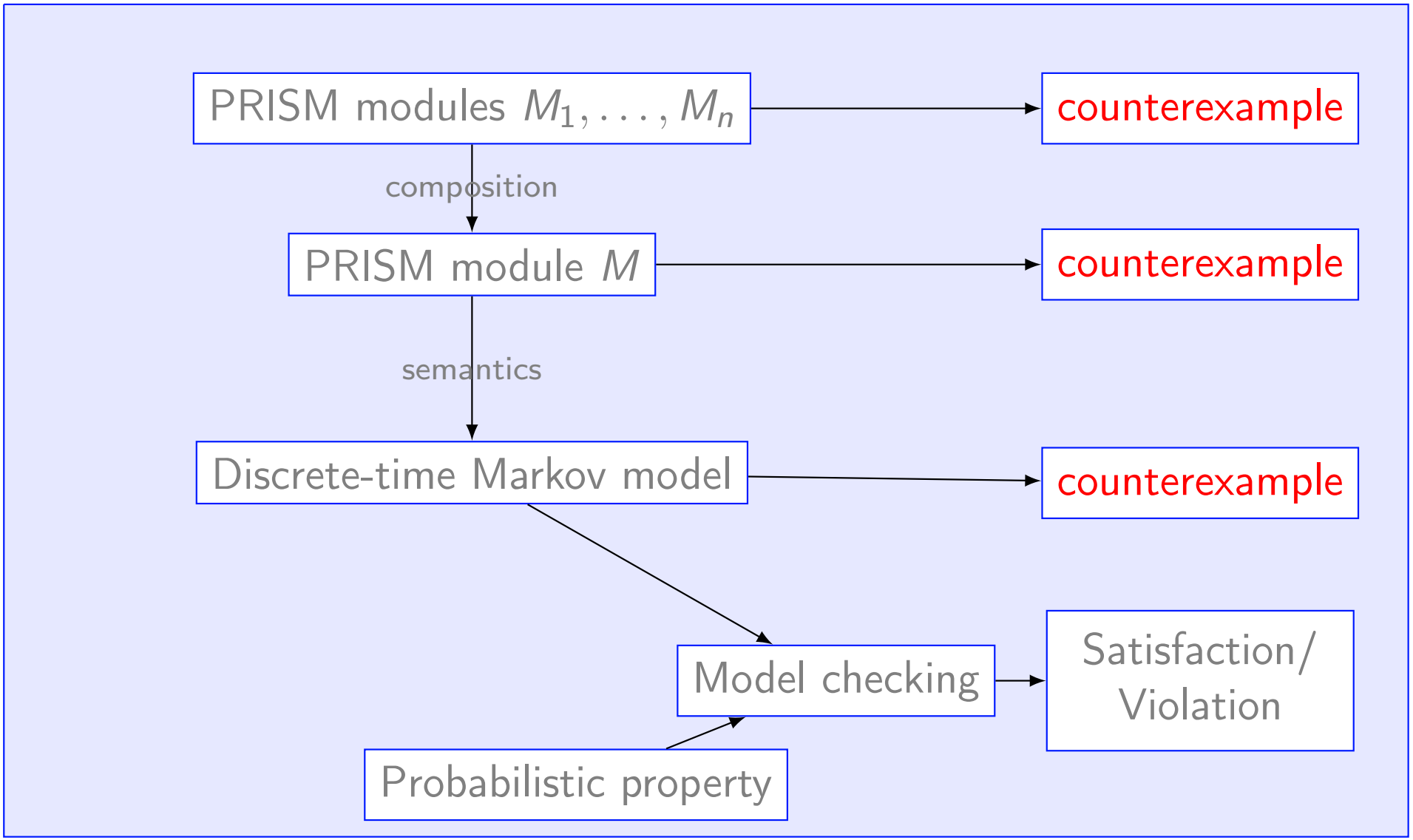
Model checking reachability properties of PRISM models

Model check the underlying DTMC/MDP/PA models.

Contents

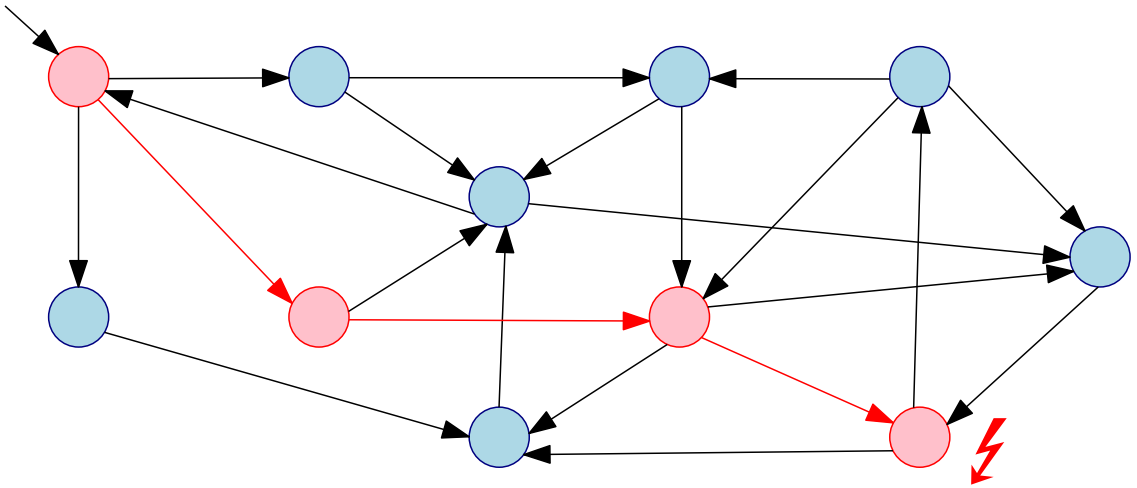

PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Probabilistic counterexamples

Counterexamples



"It is impossible to overestimate the importance of the counterexample feature. The counterexamples are invaluable in debugging complex systems. Some people use model checking just for this feature."

Edmund Clarke, Turing-Award Winner 2007

- Application areas:
  - Guide for debugging erroneous systems (error reproduction)
  - Counterexample-guided abstraction refinement (CEGAR)

Counterexamples for DTMCs

- Digital systems:
  - Safety property: $\mathcal{AG}\, safe$
  - Violation: $\mathcal{EF}\, \neg safe$
  - Counterexample: Path from the initial state to a $\neg safe$ state

Counterexamples for DTMCs

- Digital systems:
  - Safety property: $\mathcal{A}\mathcal{G}\, safe$
  - Violation: $\mathcal{E}\mathcal{F}\, \neg safe$
  - Counterexample: Path from the initial state to a $\neg safe$ state

- Probabilistic systems:
  - Safety property: $\mathcal{P}_{\geq\lambda}(\mathcal{G}\, safe)$
  - Violation: $\mathcal{P}_{>1-\lambda}(\mathcal{F}\, \neg safe)$
  - Counterexample: Set C of paths from the initial state to a $\neg safe$ state with $Pr(C) > 1 - \lambda$
  - Not computed as a by-product of model checking

Counterexamples for DTMCs

- Digital systems:
  - Safety property: $\mathcal{AG}\, safe$
  - Violation: $\mathcal{EF}\, \neg safe$
  - Counterexample: Path from the initial state to a $\neg safe$ state

- Probabilistic systems:
  - Safety property: $\mathcal{P}_{\geq\lambda}(\mathcal{G}\, safe)$
  - Violation: $\mathcal{P}_{>1-\lambda}(\mathcal{F}\, \neg safe)$
  - Counterexample: Set C of paths from the initial state to a $\neg safe$ state with $Pr(C) > 1 - \lambda$
  - Not computed as a by-product of model checking

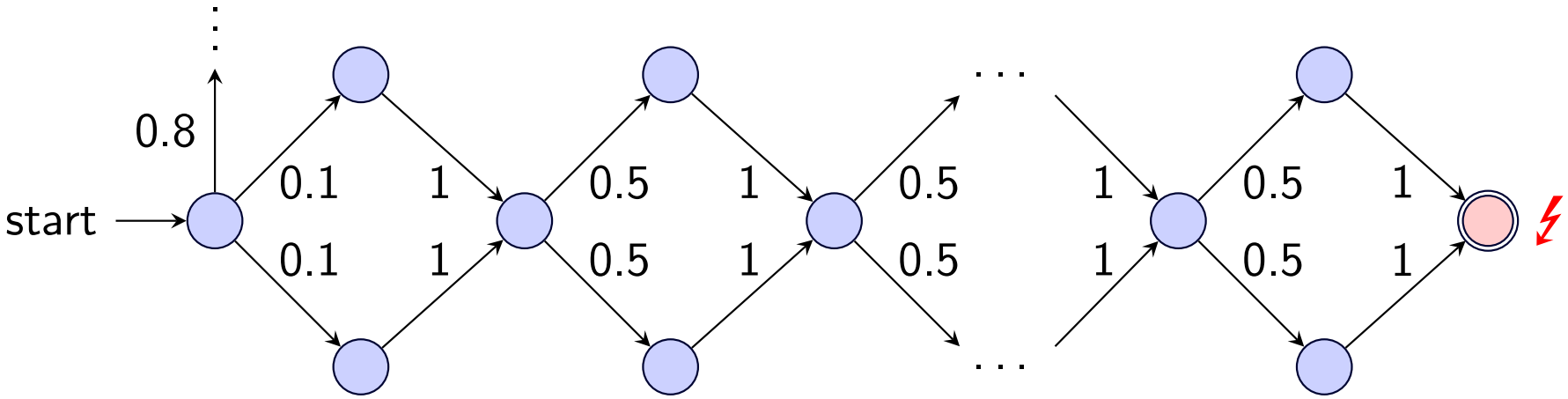
Existing Tools:

- DiPro *(Aljazzar et al.)*
- COMICS *(Jansen et al.)*

Problem

The number of paths in C can be VERY large – much larger than the system itself!

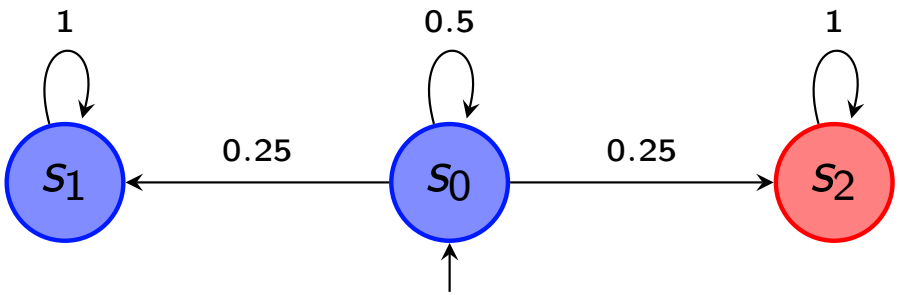
Large counterexamples



Property $\mathbb{P}_{<0.2}(\Diamond \neg \mathit{safe})$

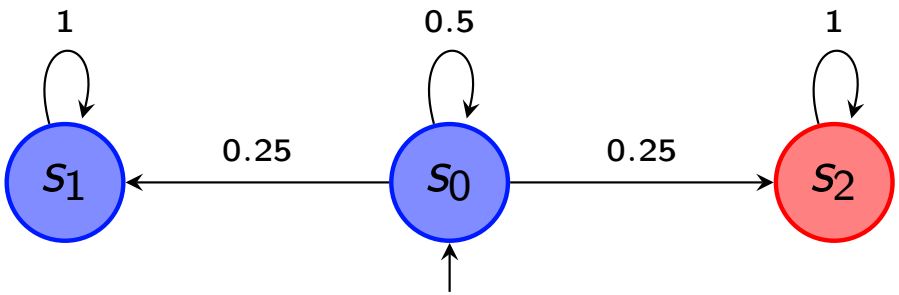
$\Rightarrow$ Number of paths in each counterexample is exponential in the number of states

Counterexamples can be even infinite sets



Property: $\mathbb{P}_{<0.5}(\Diamond s_2)$

Counterexamples can be even infinite sets



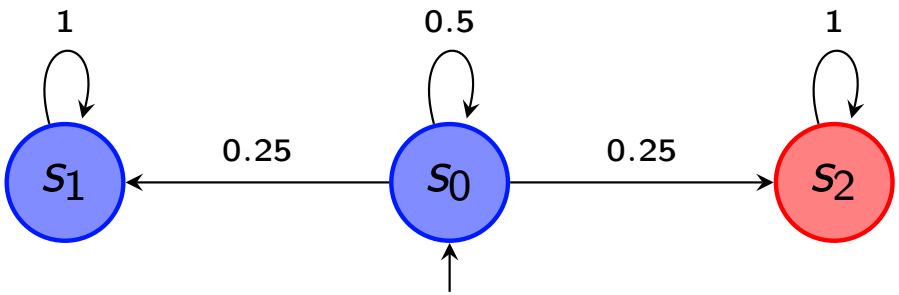
Property: $\mathbb{P}_{<0.5}(\Diamond s_2)$

Consider set C of all paths leading to state s_2:

$$C = \{(s_0) \to s_2, (s_0)^2 \to s_2, (s_0)^3 \to s_2, \ldots\}$$

Probability of C: $\sum_{i=0}^{\infty}(0.5)^i \cdot 0.25$

Counterexamples can be even infinite sets



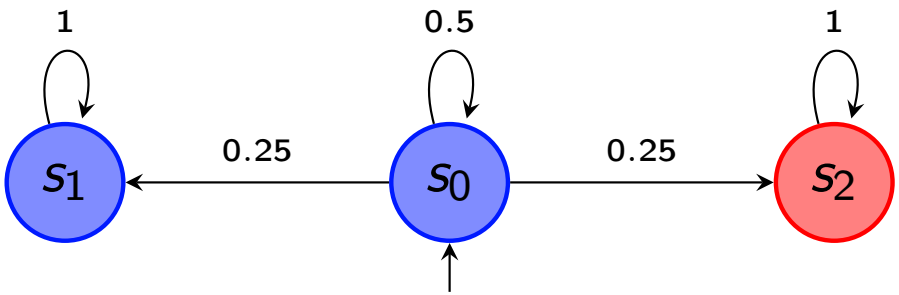
Property: $\mathbb{P}_{<0.5}(\Diamond s_2)$

Consider set C of all paths leading to state s_2:

$$C = \{(s_0) \to s_2, (s_0)^2 \to s_2, (s_0)^3 \to s_2, \ldots\}$$

Probability of C: $\sum_{i=0}^{\infty}(0.5)^i \cdot 0.25 \overset{\text{geom. ser.}}{=} \frac{1}{1-0.5} \cdot 0.25$

Counterexamples can be even infinite sets



Property is violated!

Property: $\mathbb{P}_{<0.5}(\Diamond s_2)$

Consider set C of all paths leading to state s_2:

$$C = \{(s_0) \to s_2, (s_0)^2 \to s_2, (s_0)^3 \to s_2, \ldots\}$$

Probability of C: $\sum_{i=0}^{\infty}(0.5)^i \cdot 0.25 \overset{\text{geom. ser.}}{=} \frac{1}{1-0.5} \cdot 0.25 = 0.5$

Critical subsystems for DTMCs

Counterexamples can be represented

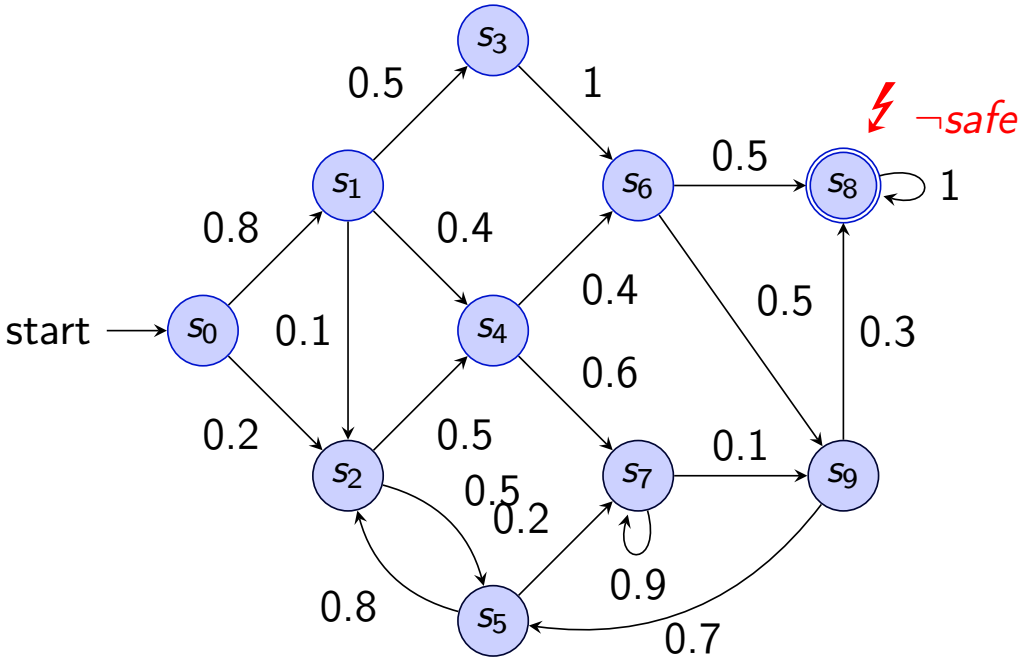
- by enumeration of the paths,
- by regular expressions, trees,...
- critical subsystems [Aljazzar/Leue, 2009; Jansen et al., 2011].

Critical subsystem

Subset S' of the states such that the probability of reaching a $\neg$*safe*-state visiting only states from S' is already beyond $1 - \lambda$.

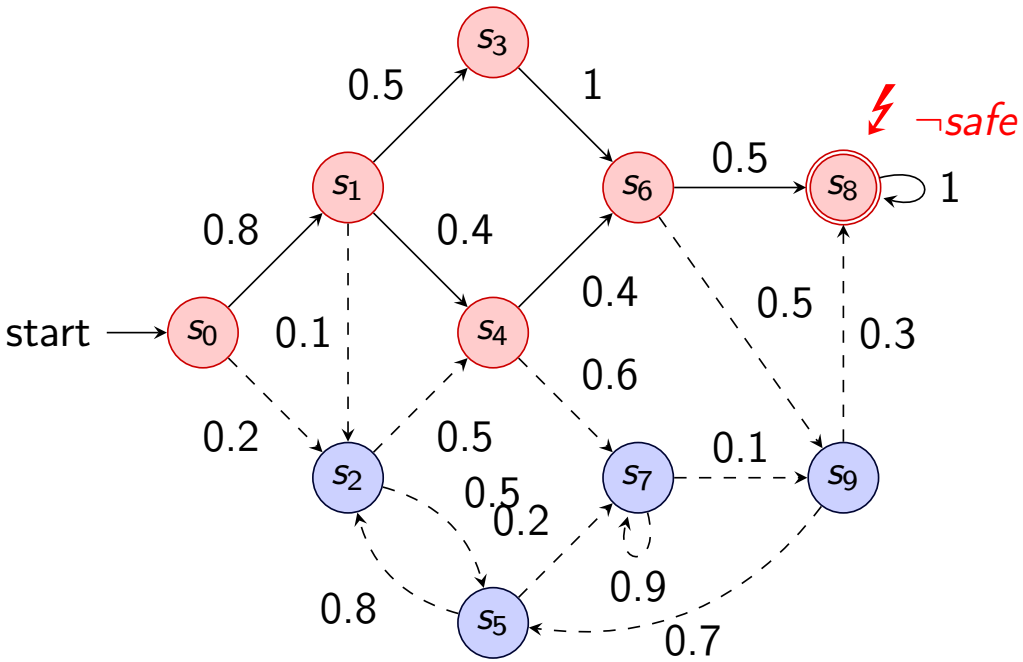
Critical subsystems for DTMCs: Example

$$\mathcal{P}_{\geq 0.75}(\mathcal{G}\, safe) \quad \equiv \quad \mathcal{P}_{\leq 0.25}(\mathcal{F}\, \neg safe)$$



Critical subsystems for DTMCs: Example

$$\mathcal{P}_{\geq 0.75}(\mathcal{G}\, safe) \quad \equiv \quad \mathcal{P}_{\leq 0.25}(\mathcal{F}\, \neg safe)$$



Path-based counterexamples

Path-based computation of hierarchical critical subsystems for DTMCs

Representation:

- Critical subsystem

Path-based computation of hierarchical critical subsystems for DTMCs

Representation:

- Critical subsystem

Method:

- SCC-based model checking
- If property was falsified:
  - Search for counterexample on abstract system
  - Hierarchical concretization

Path-based computation of hierarchical critical subsystems for DTMCs

Representation:

- Critical subsystem

Method:

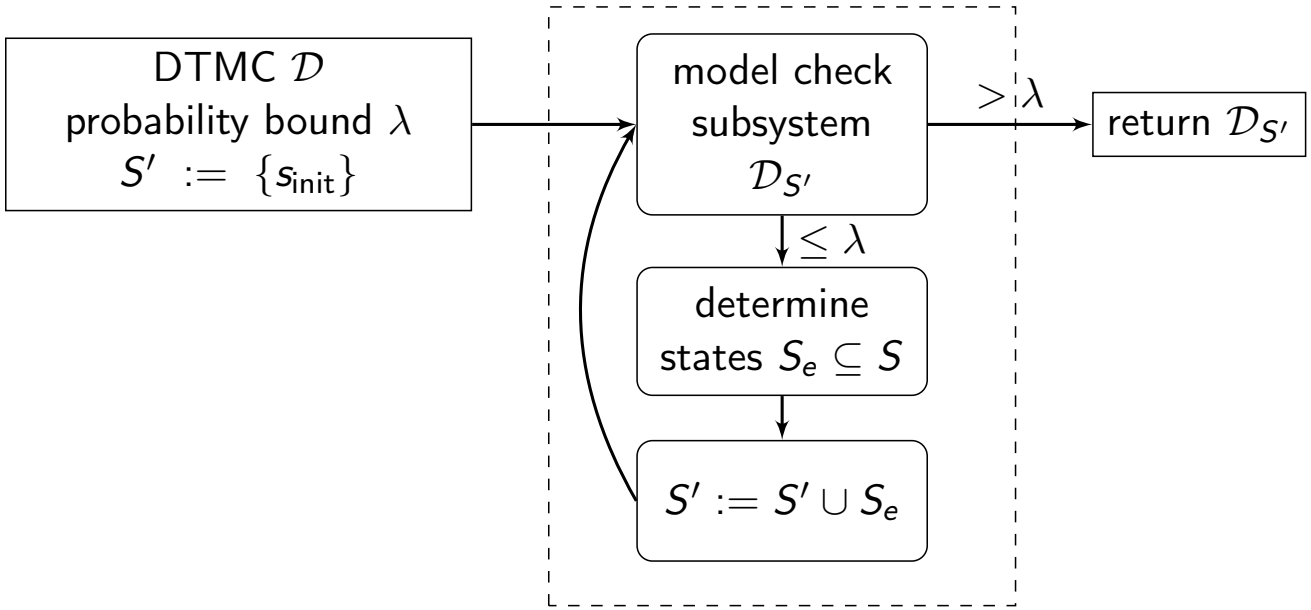
- SCC-based model checking
- If property was falsified:
  - Search for counterexample on abstract system
  - Hierarchical concretization

Advantages:

- Compact representation $\rightsquigarrow$ Usability
- Abstract counterexamples $\rightsquigarrow$ Treatment of large systems
- Hierarchical approach $\rightsquigarrow$ Omission of irrelevant system parts

Hierarchical algorithm for DTMCs - Overview

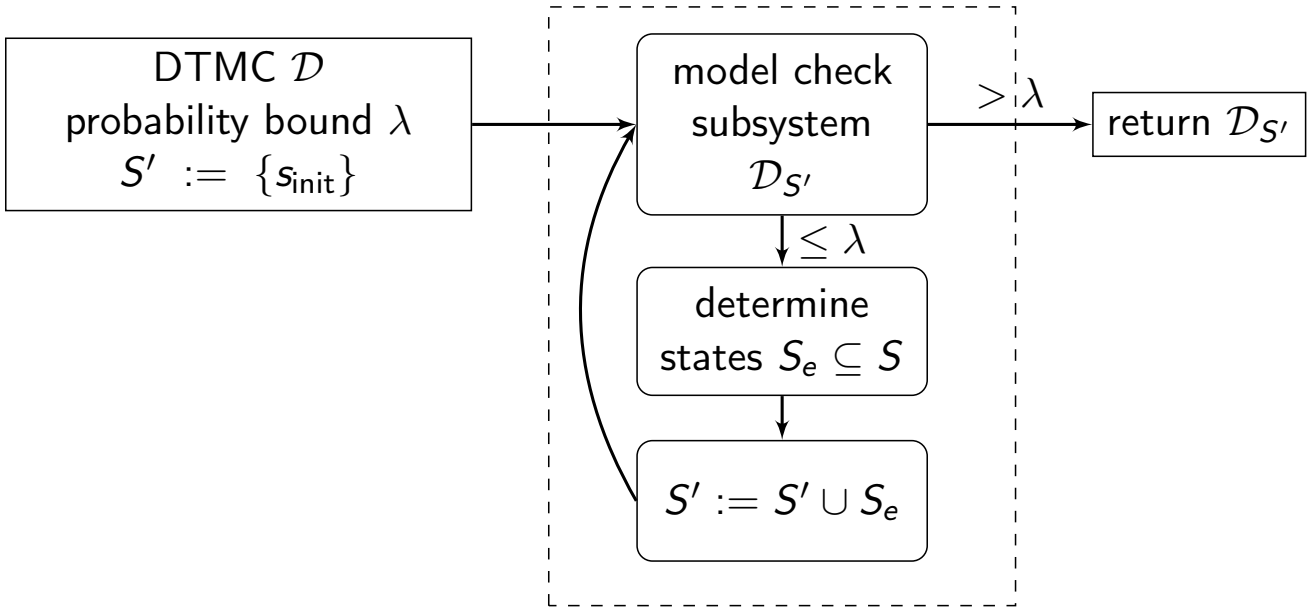
- Input: Abstract DTMC, target states, probability bound
- Find a critical subsystem



- Concretize one or more abstract states and refine the critical subsystem.

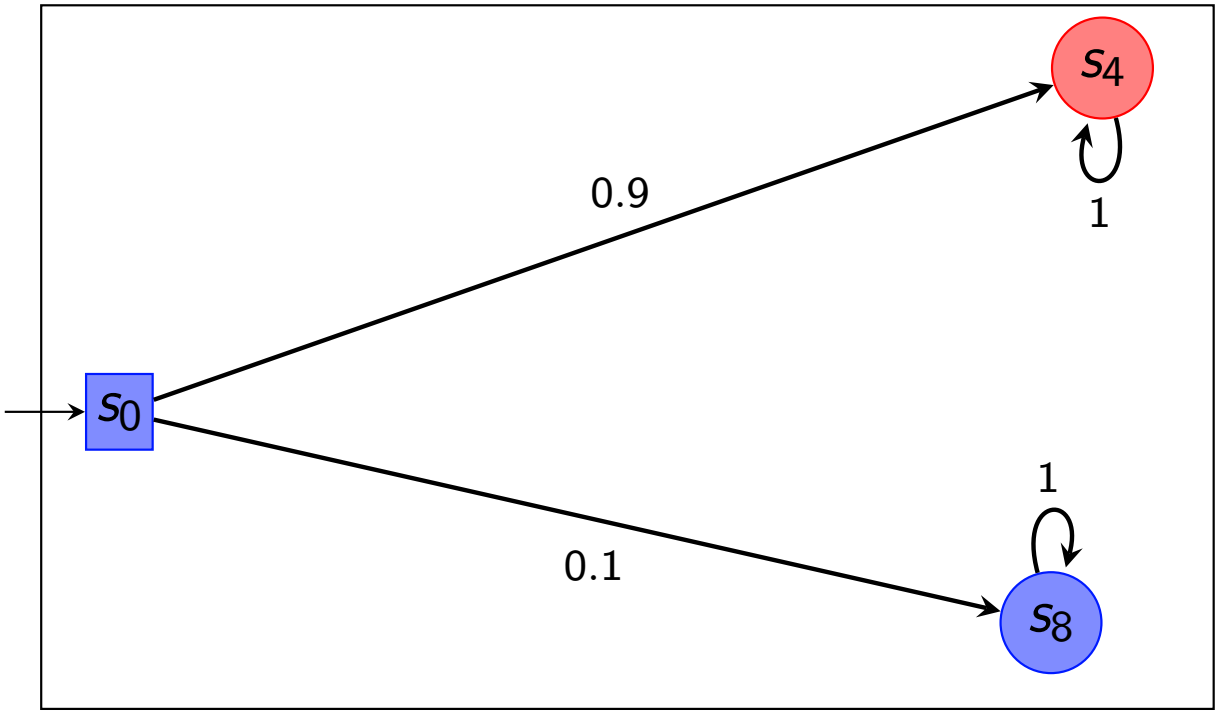
Hierarchical algorithm for DTMCs - Overview

- Input: Abstract DTMC, target states, probability bound
- Find a critical subsystem



- Concretize one or more abstract states and refine the critical subsystem.
- Global Search
  - Searches for the most probable paths
- Local Search
  - Searches for most probable path fragments connecting parts of previously found paths

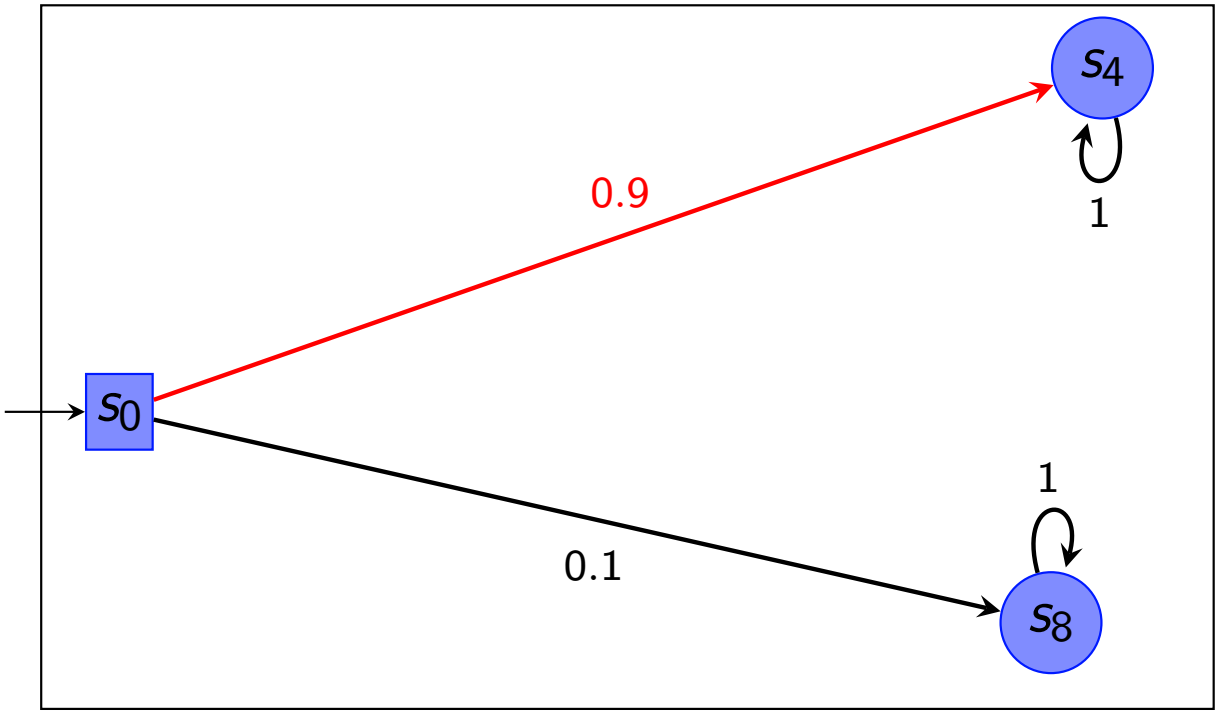
Hierarchical counterexamples for DTMCs - Global search



Task: Find a critical subsystem that violates

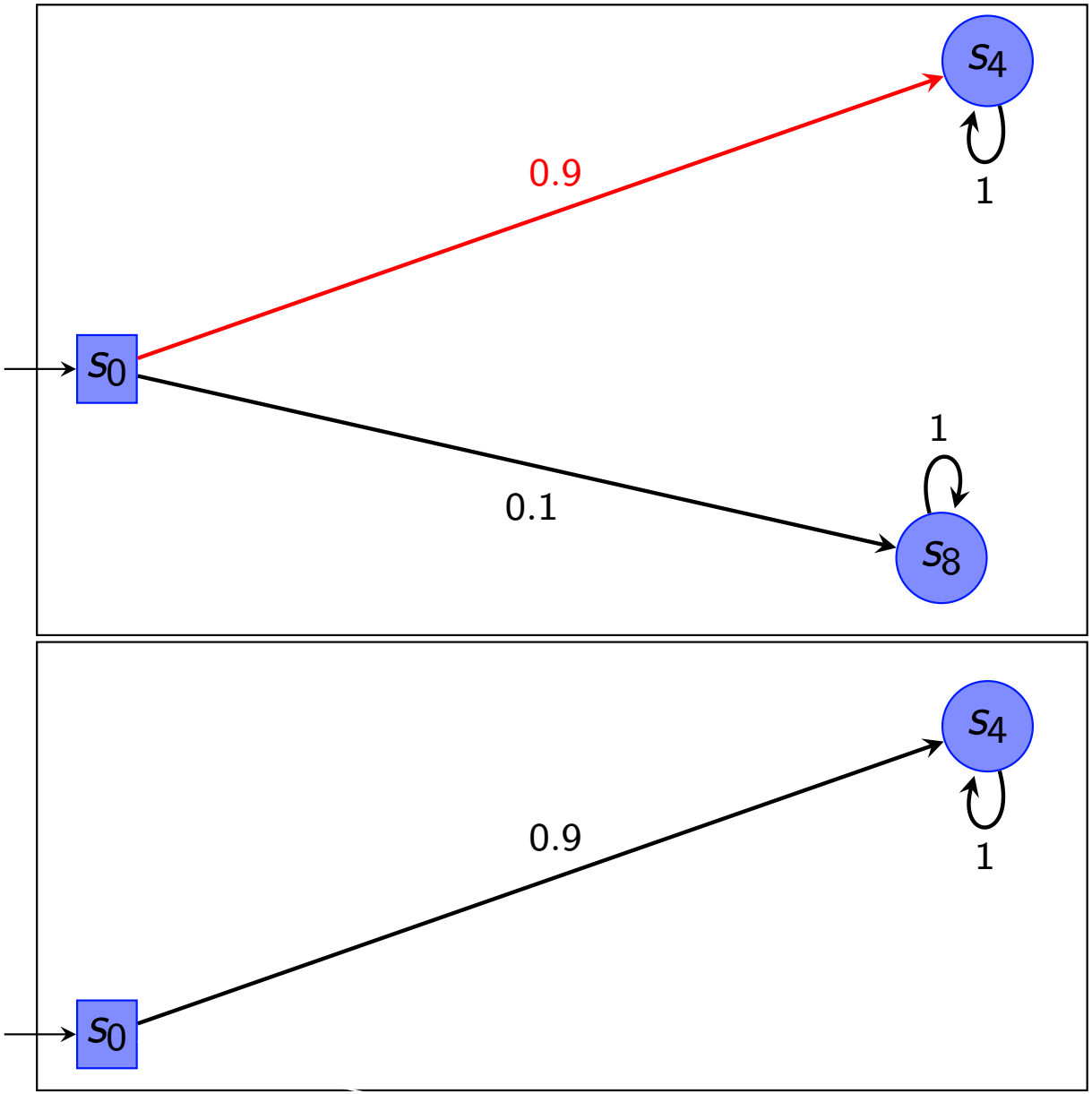
$$\mathbb{P}_{<0.5}(\Diamond 4)$$

Hierarchical counterexamples for DTMCs - Global search



Initial Search

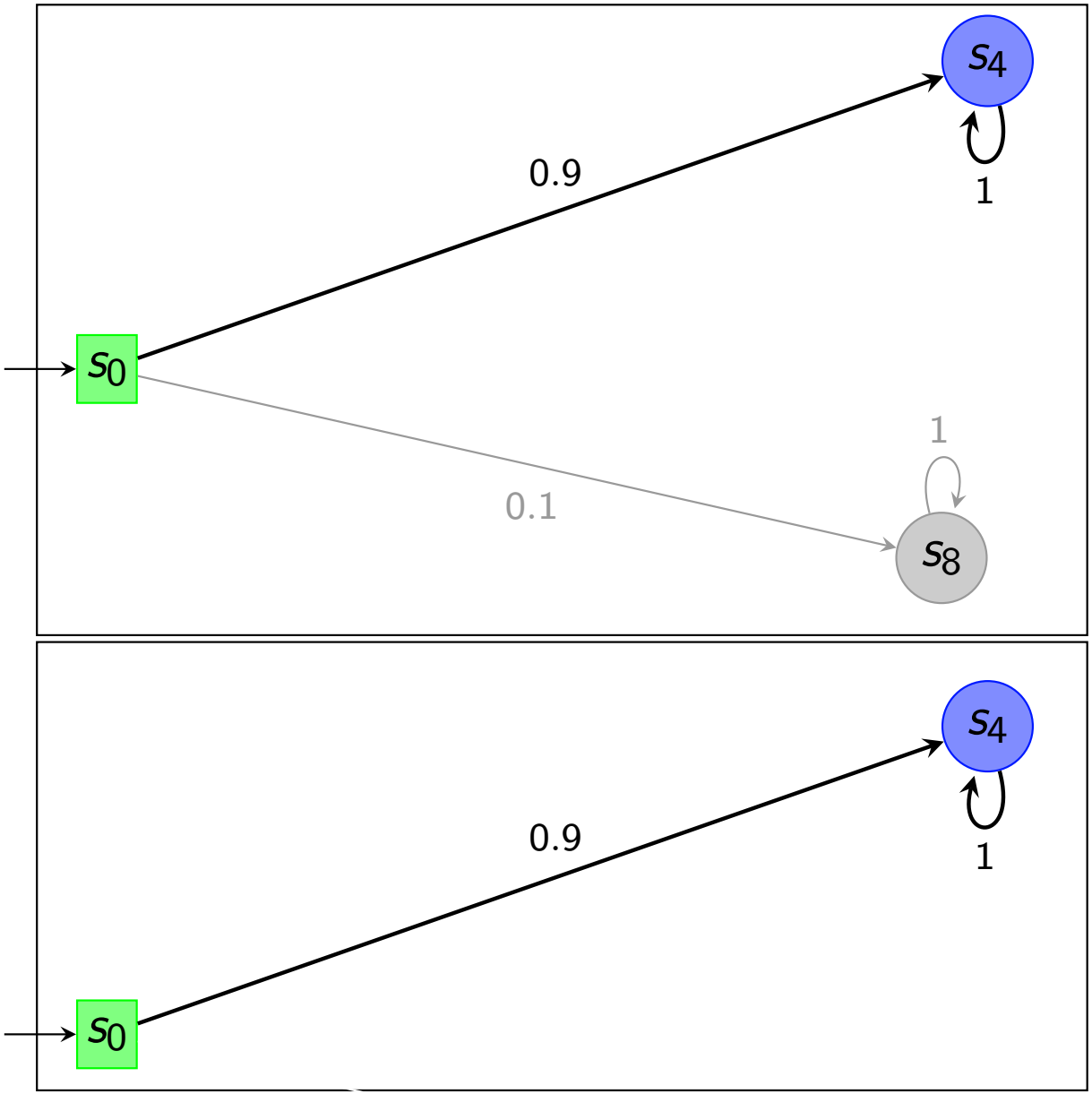
Hierarchical counterexamples for DTMCs - Global search



Critical subsystem

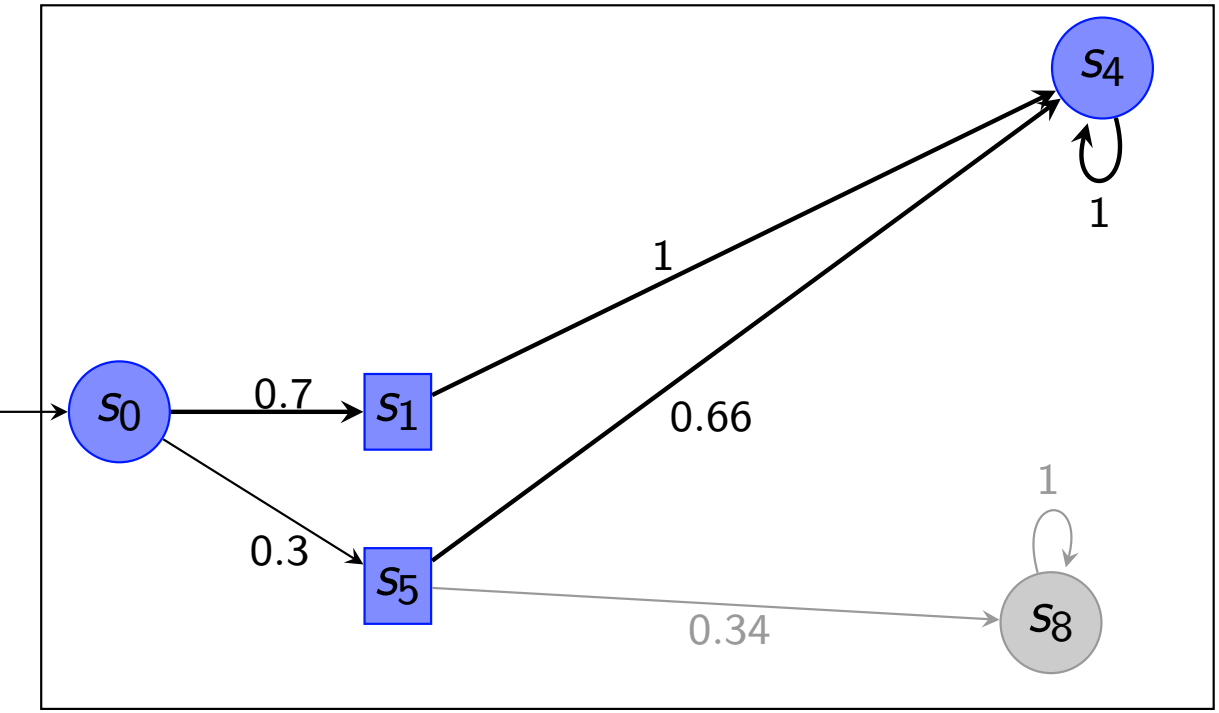
Probability 0.9

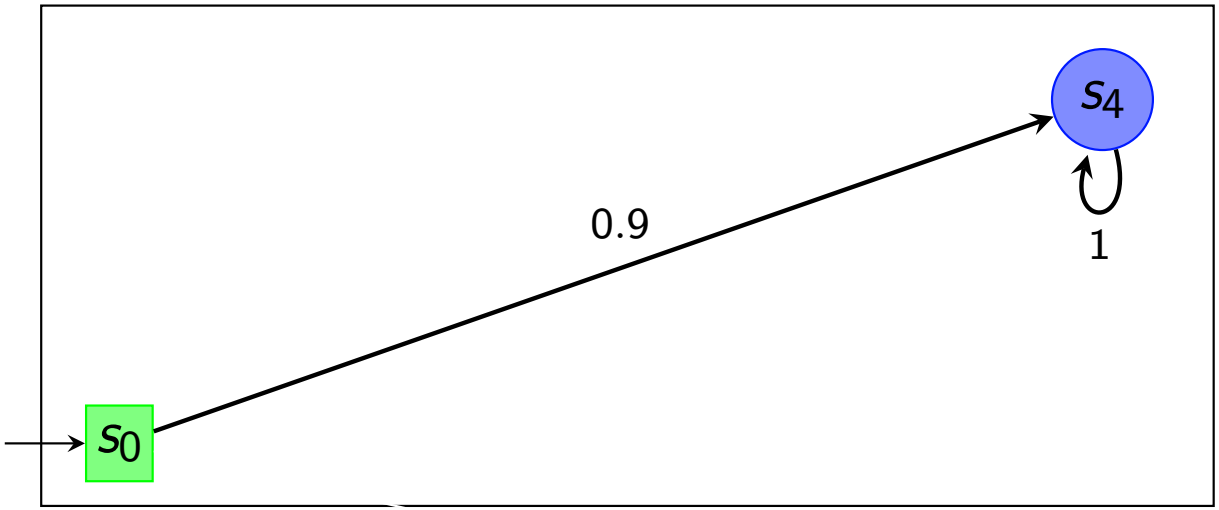
Hierarchical counterexamples for DTMCs - Global search



Choose state 0 for concretization

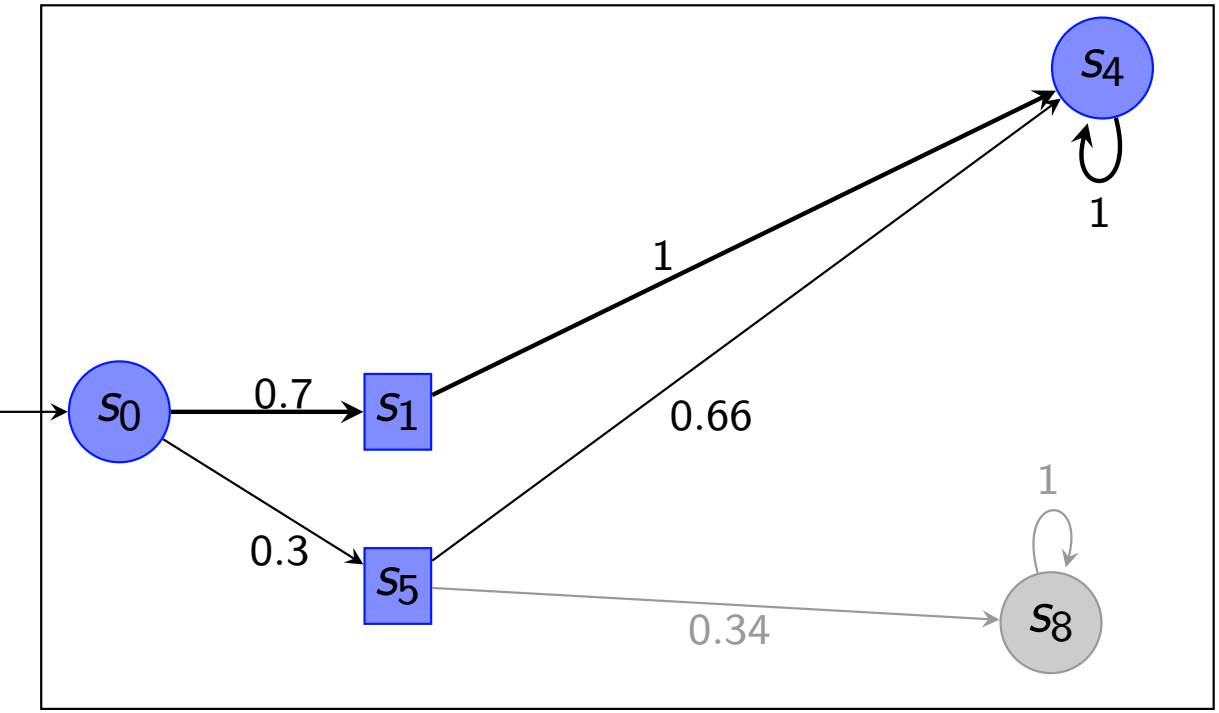
Hierarchical counterexamples for DTMCs - Global search

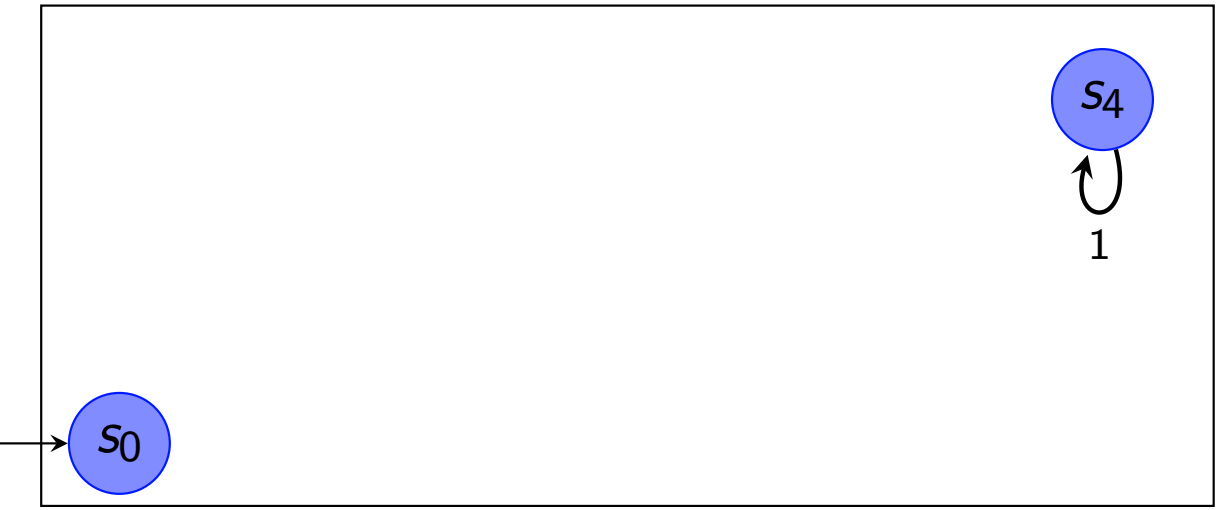




Concretize state 0

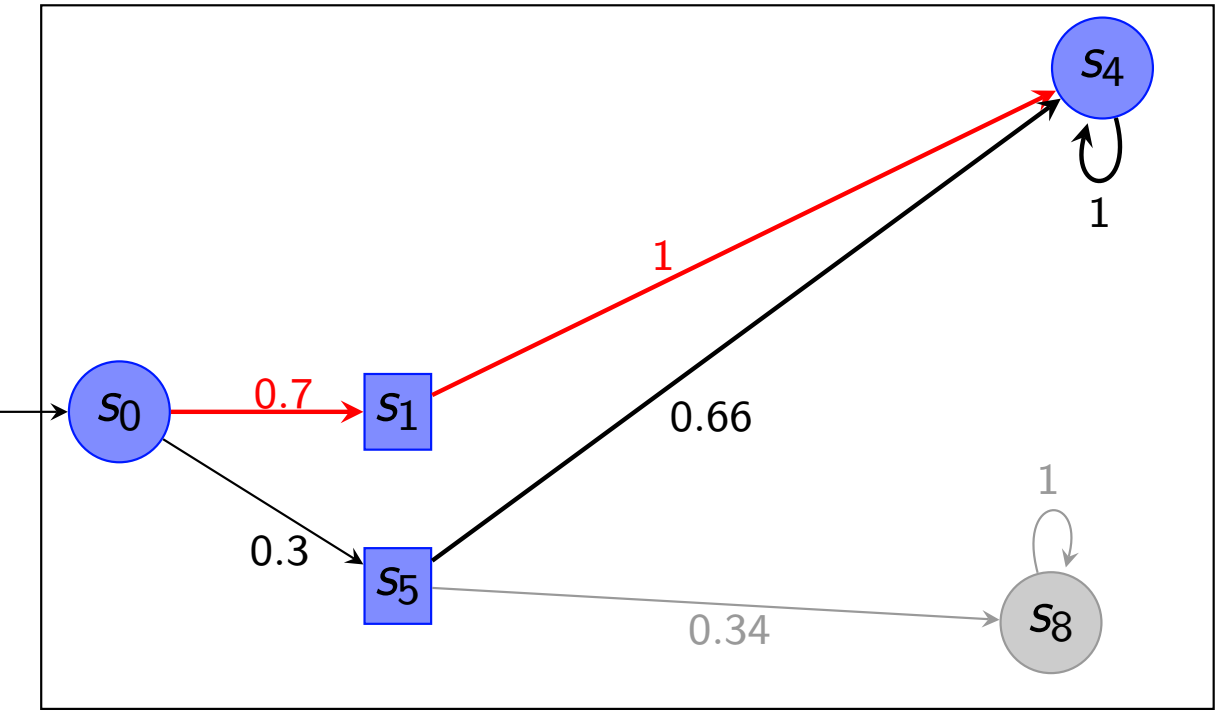
Hierarchical counterexamples for DTMCs - Global search

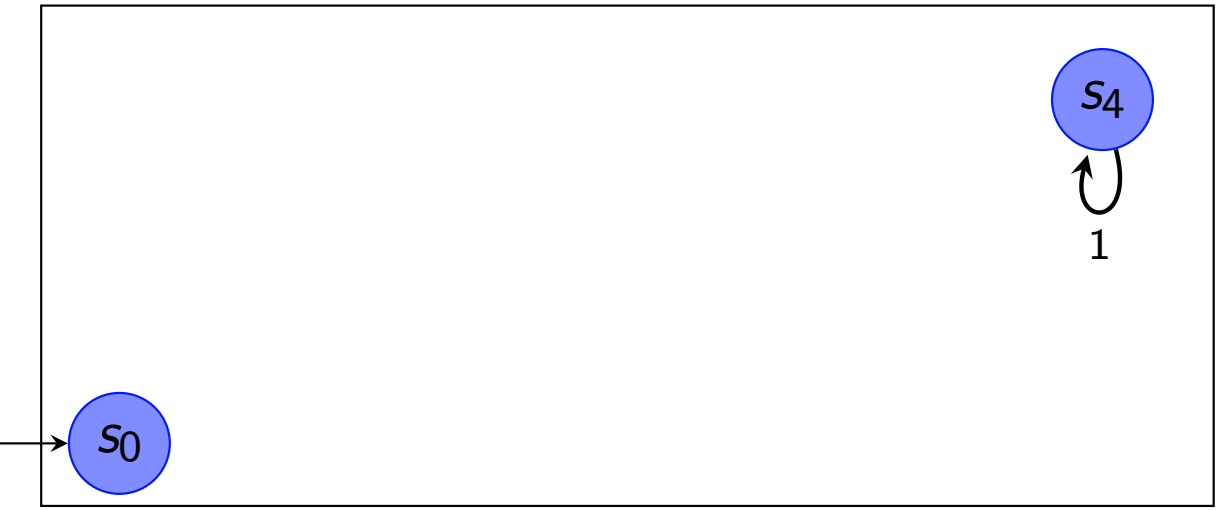




Concretize state 0

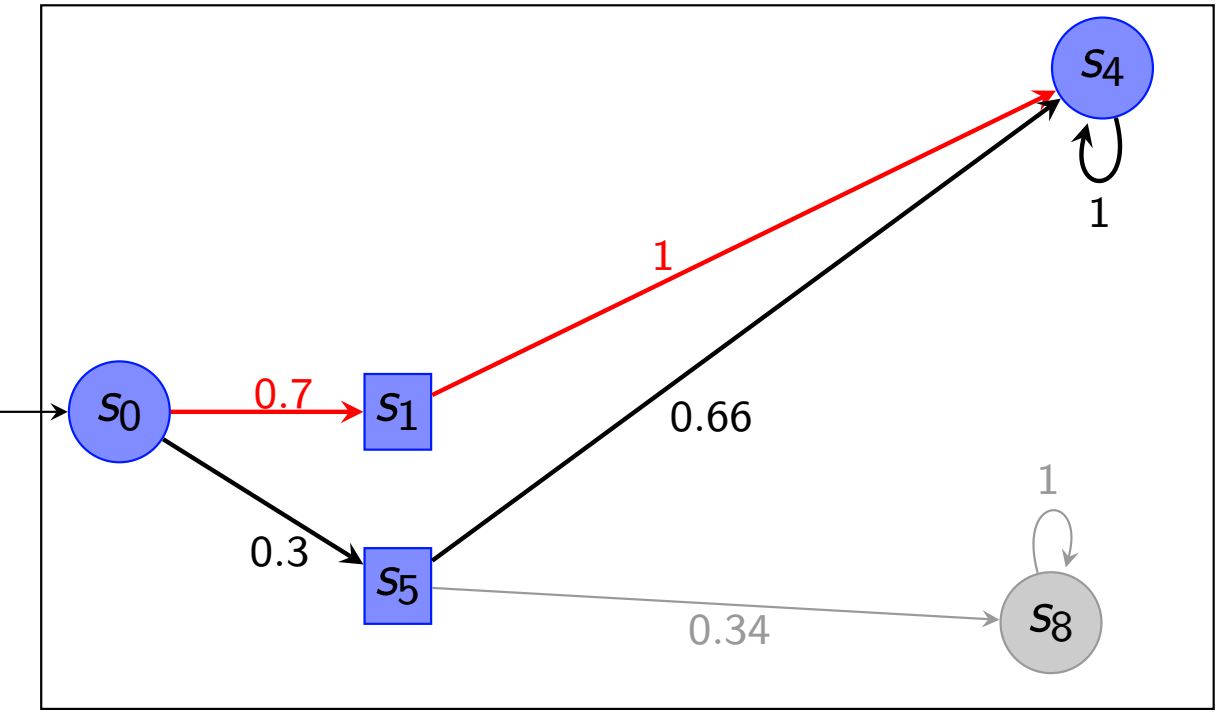
Hierarchical counterexamples for DTMCs - Global search

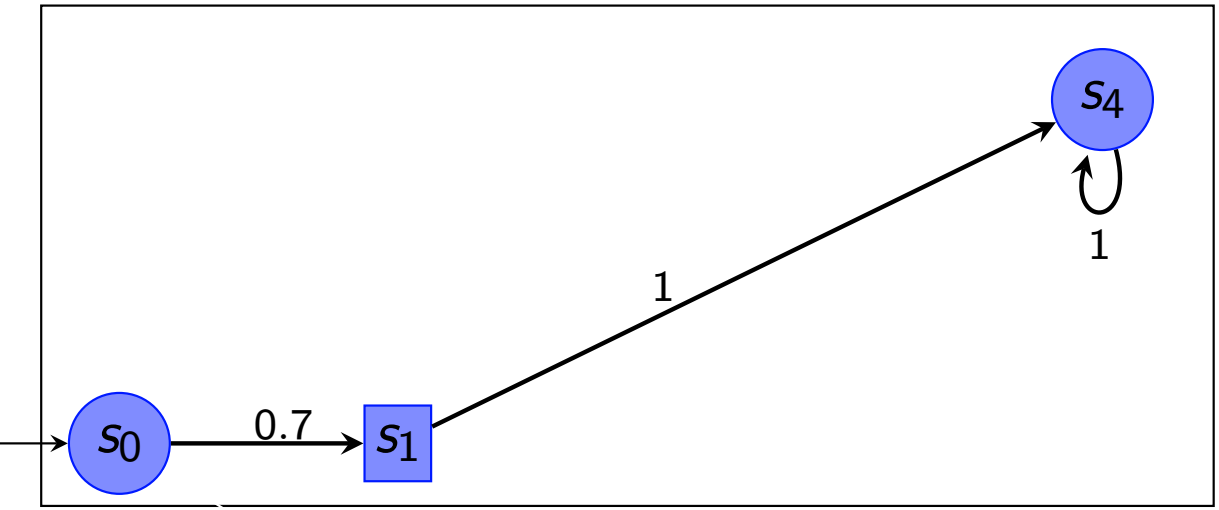




Search for most probable path

Hierarchical counterexamples for DTMCs - Global search

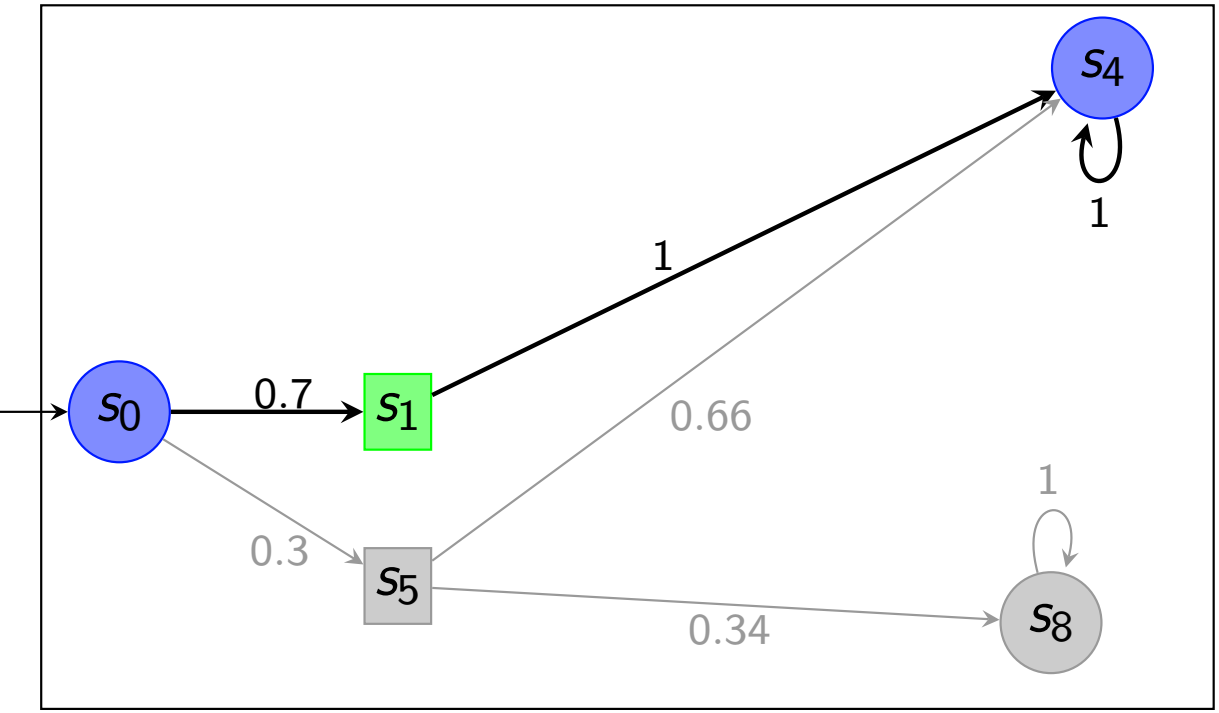


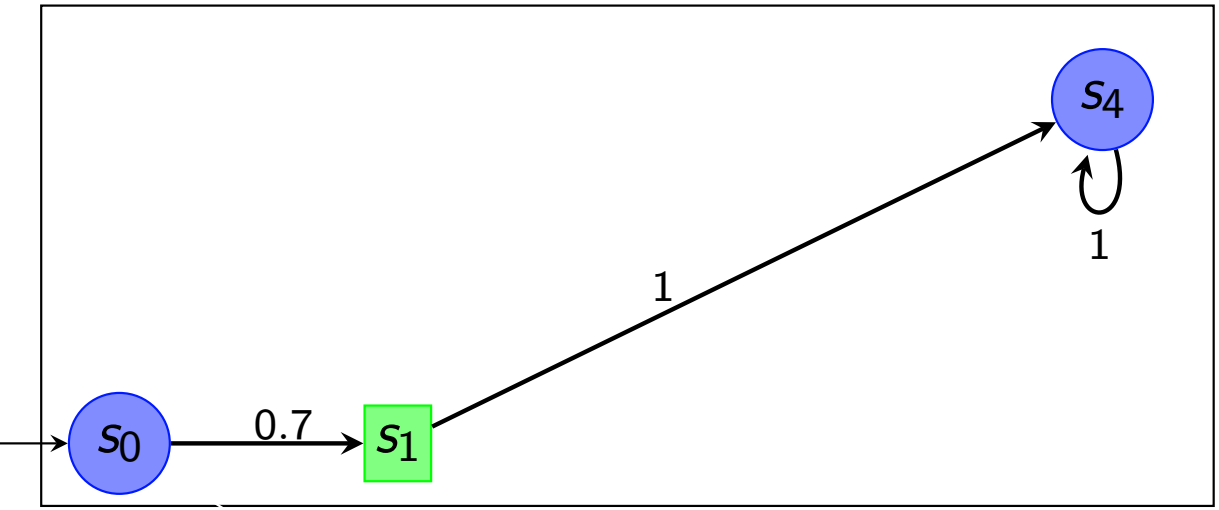


Critical subsystem updated

Probability 0.7

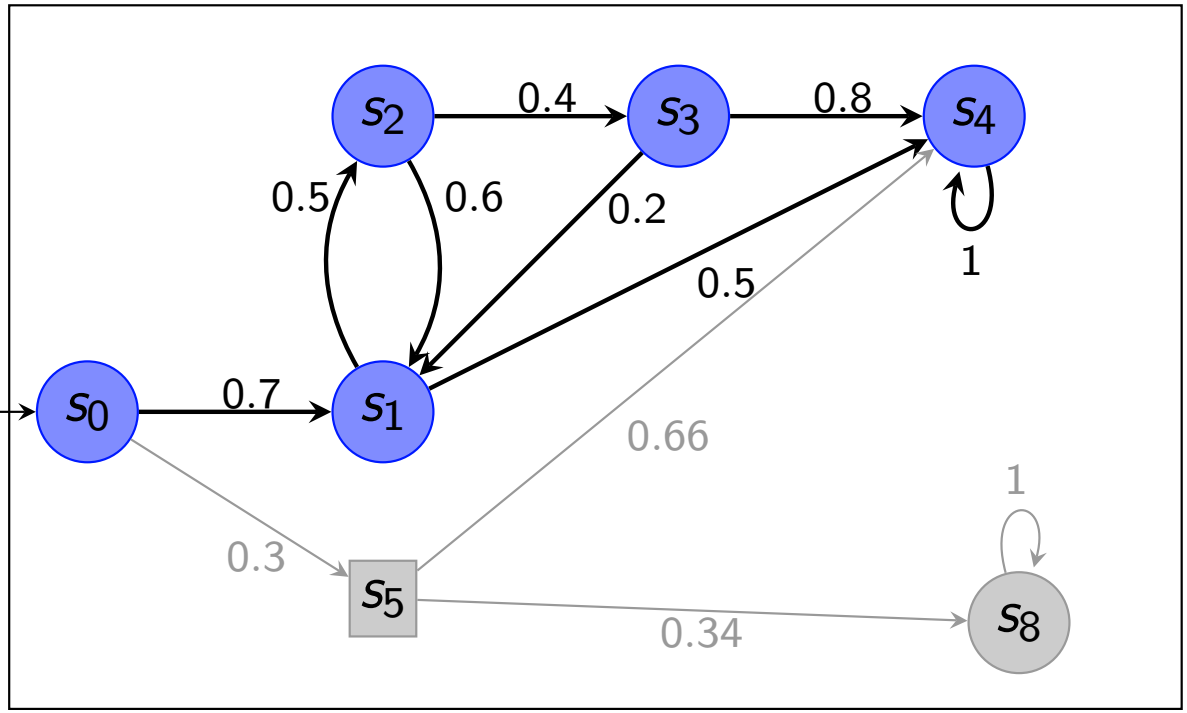
Hierarchical counterexamples for DTMCs - Global search

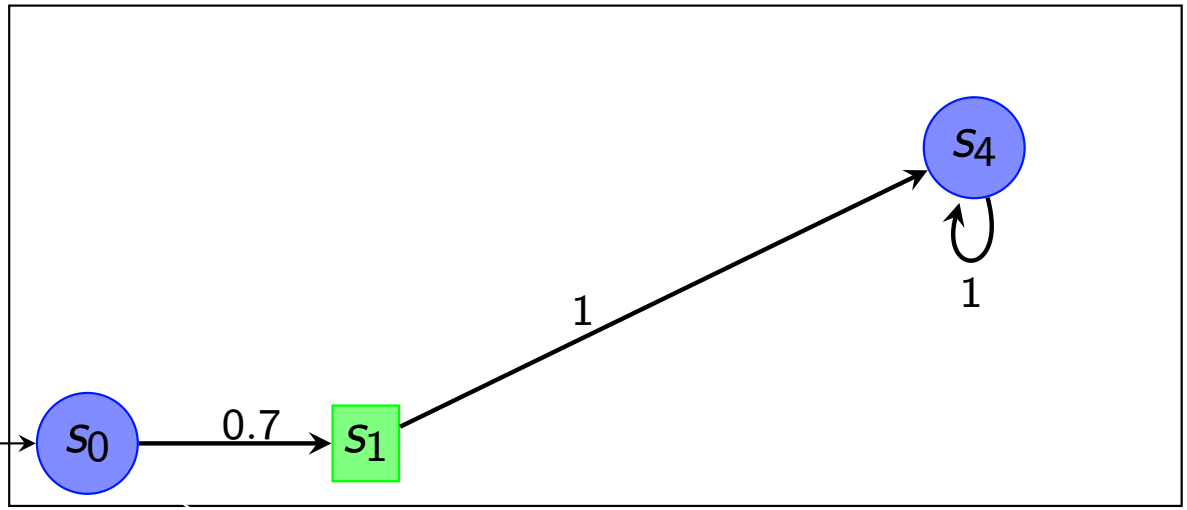




Choose state 1 for concretization

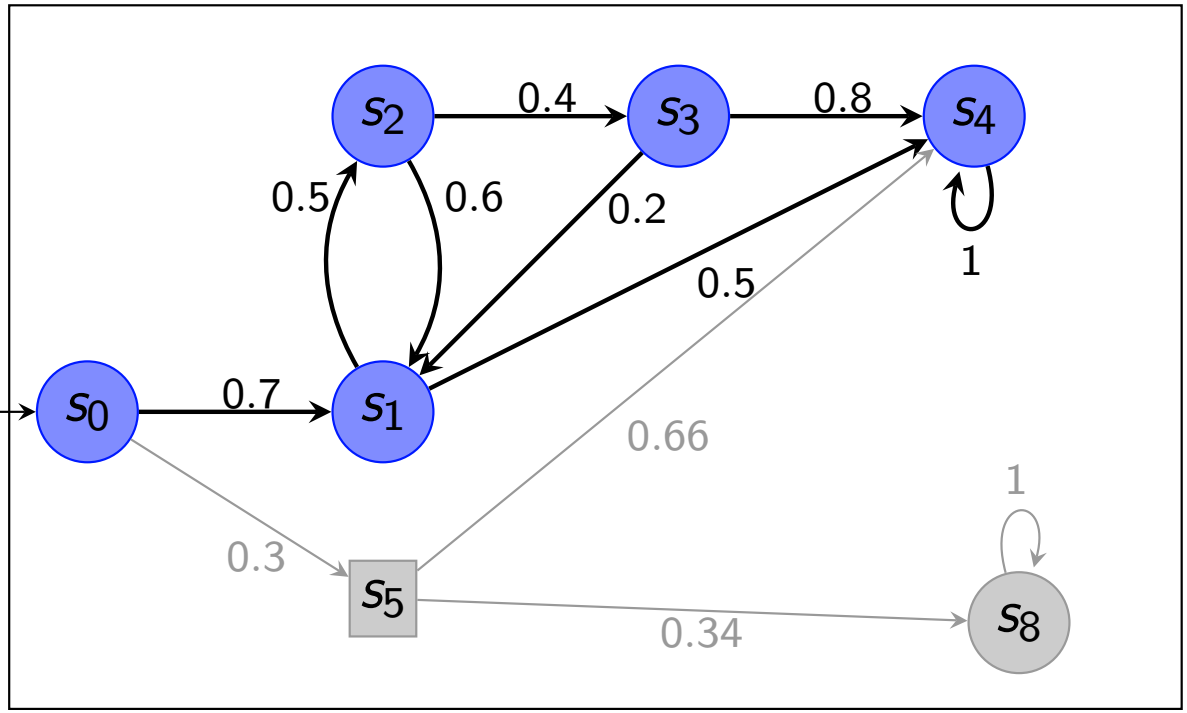
Hierarchical counterexamples for DTMCs - Global search

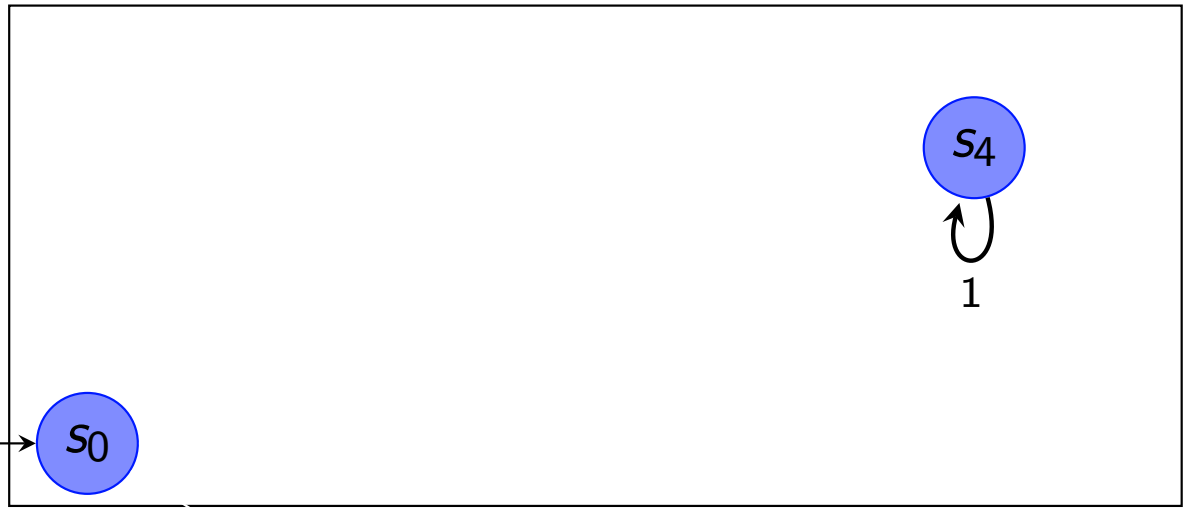




Concretize state 1

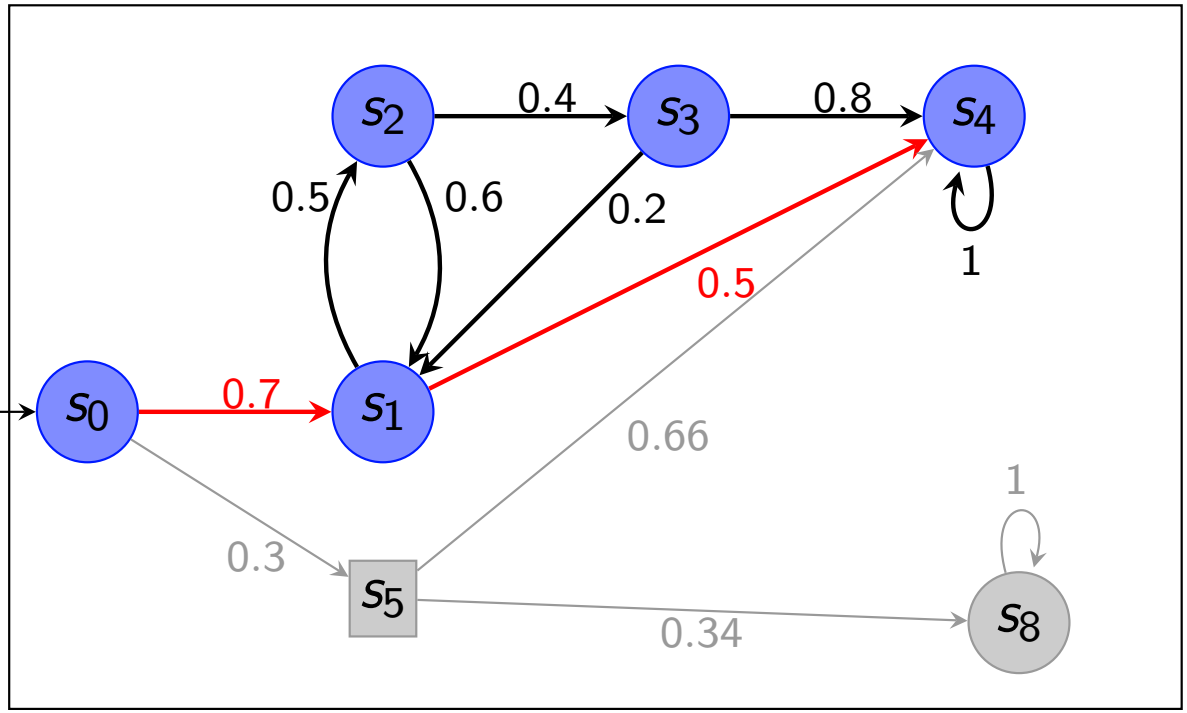
Hierarchical counterexamples for DTMCs - Global search

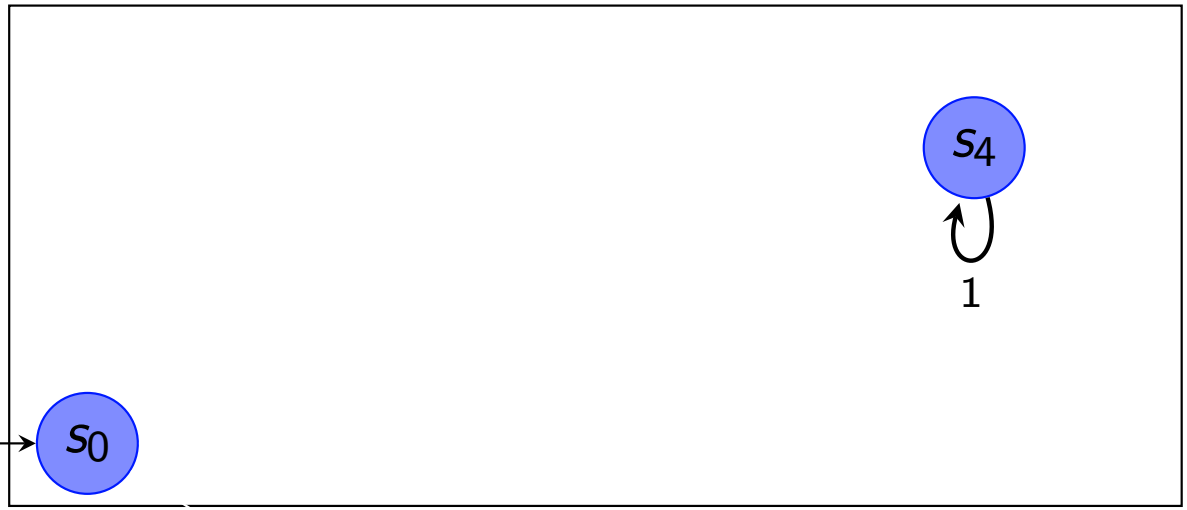




Concretize state 1

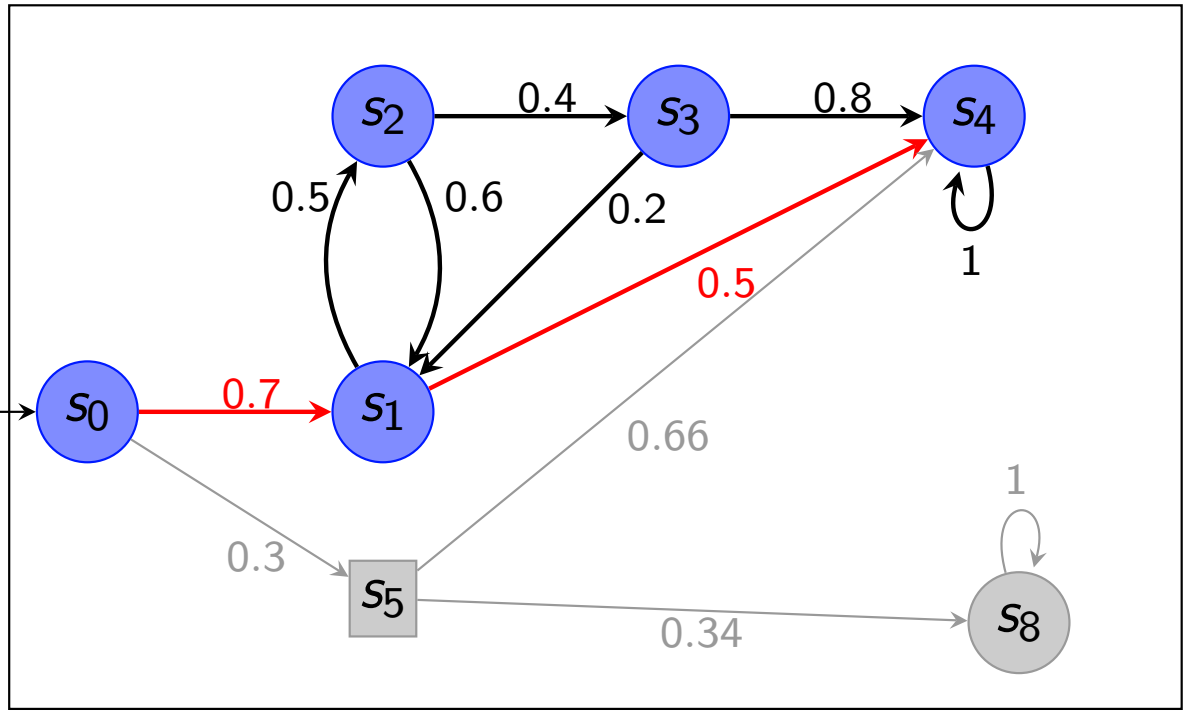
Hierarchical counterexamples for DTMCs - Global search

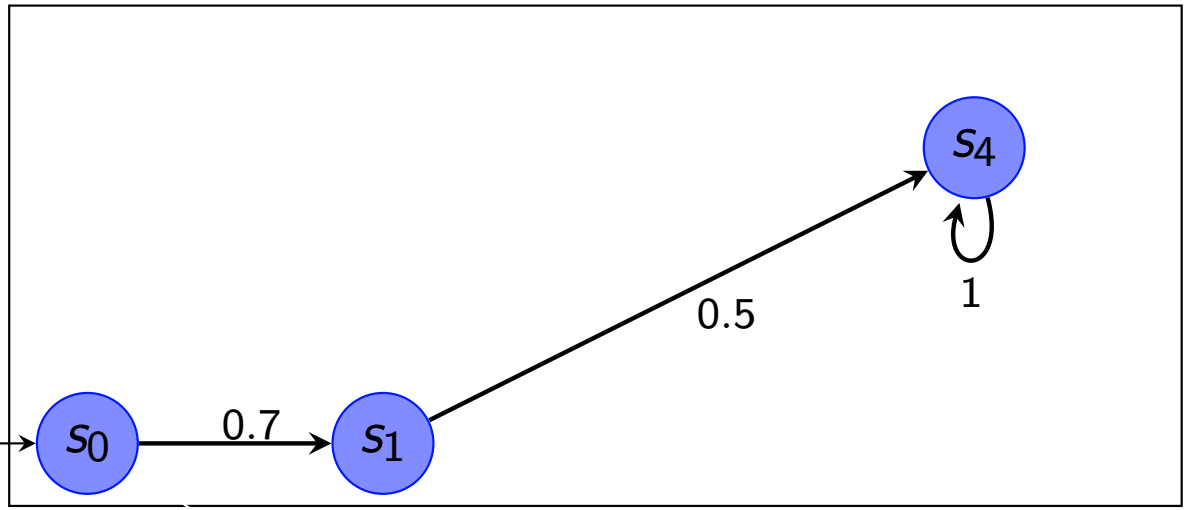




Search for most probable path

Hierarchical counterexamples for DTMCs - Global search

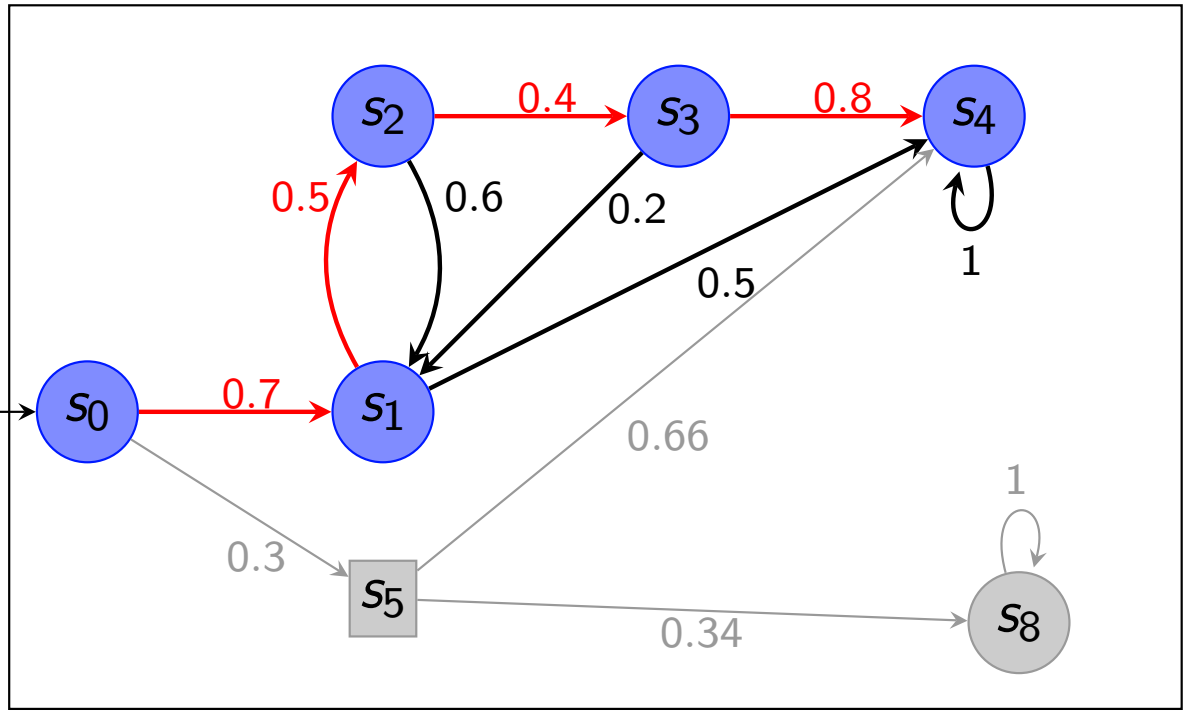


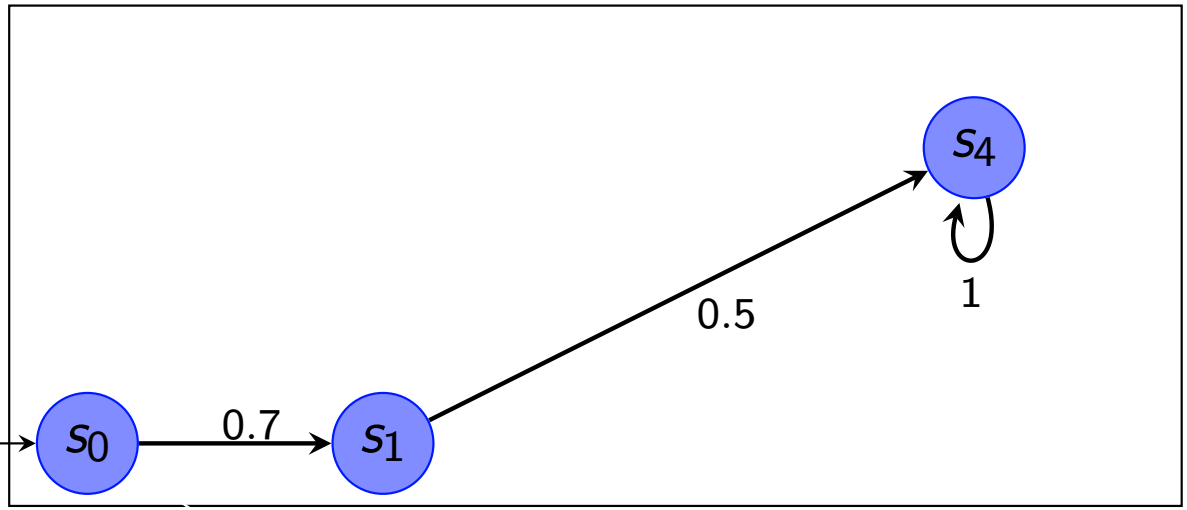


Critical subsystem updated

Probability 0.35

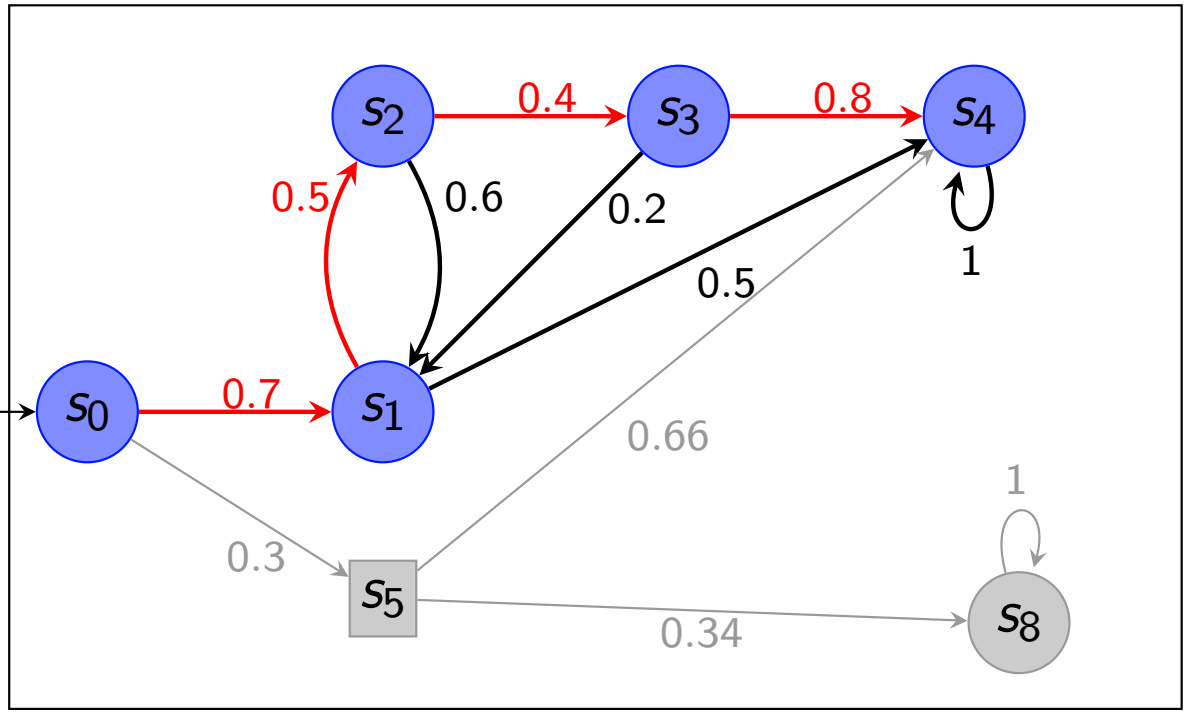
Hierarchical counterexamples for DTMCs - Global search

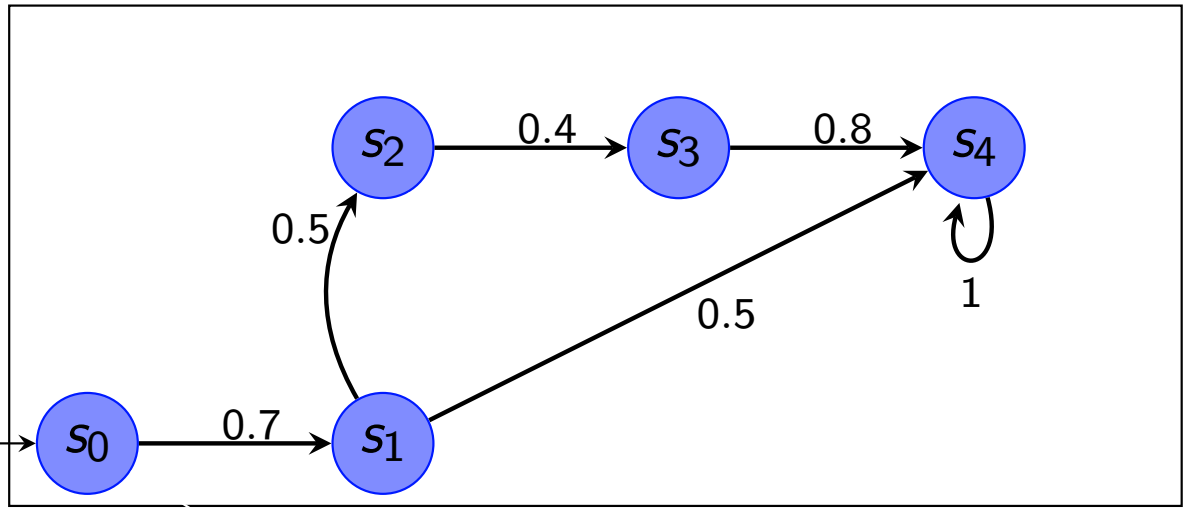




Search for next most probable path

Hierarchical counterexamples for DTMCs - Global search

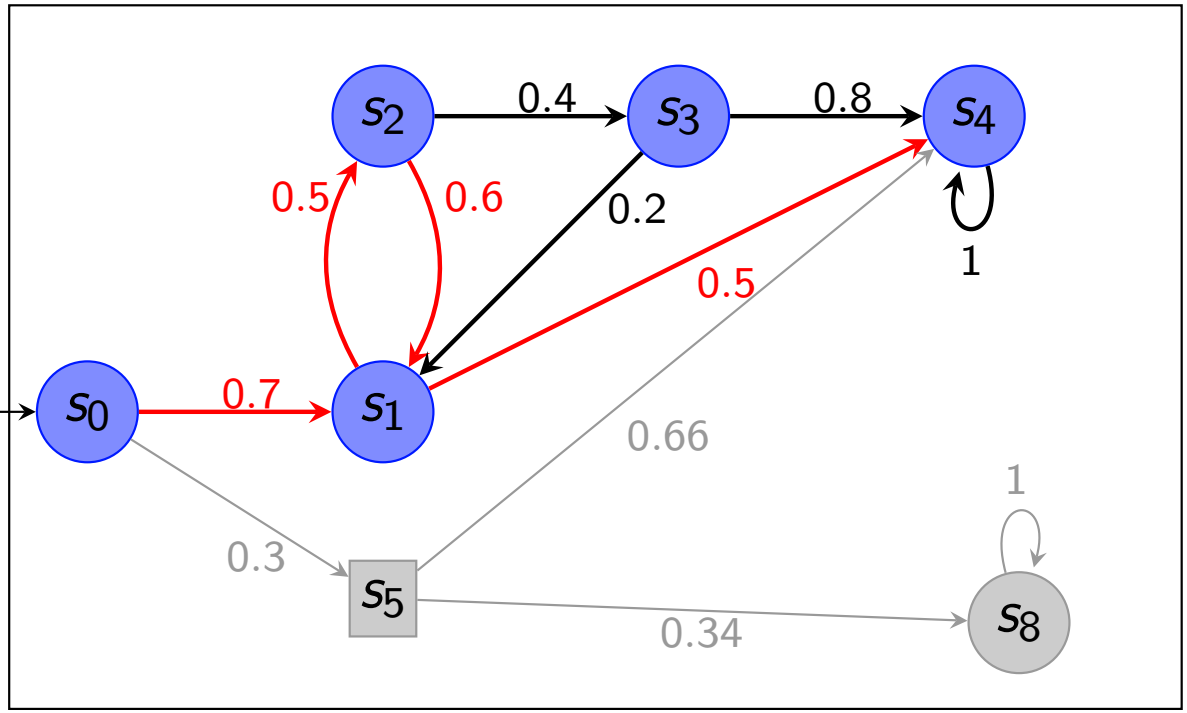


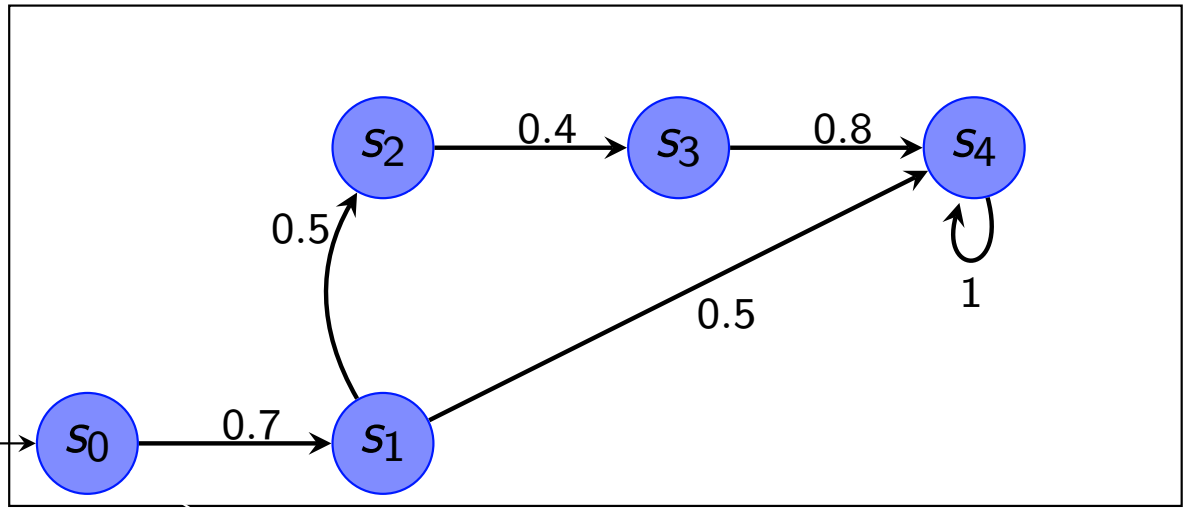


Critical subsystem updated

Probability 0.462

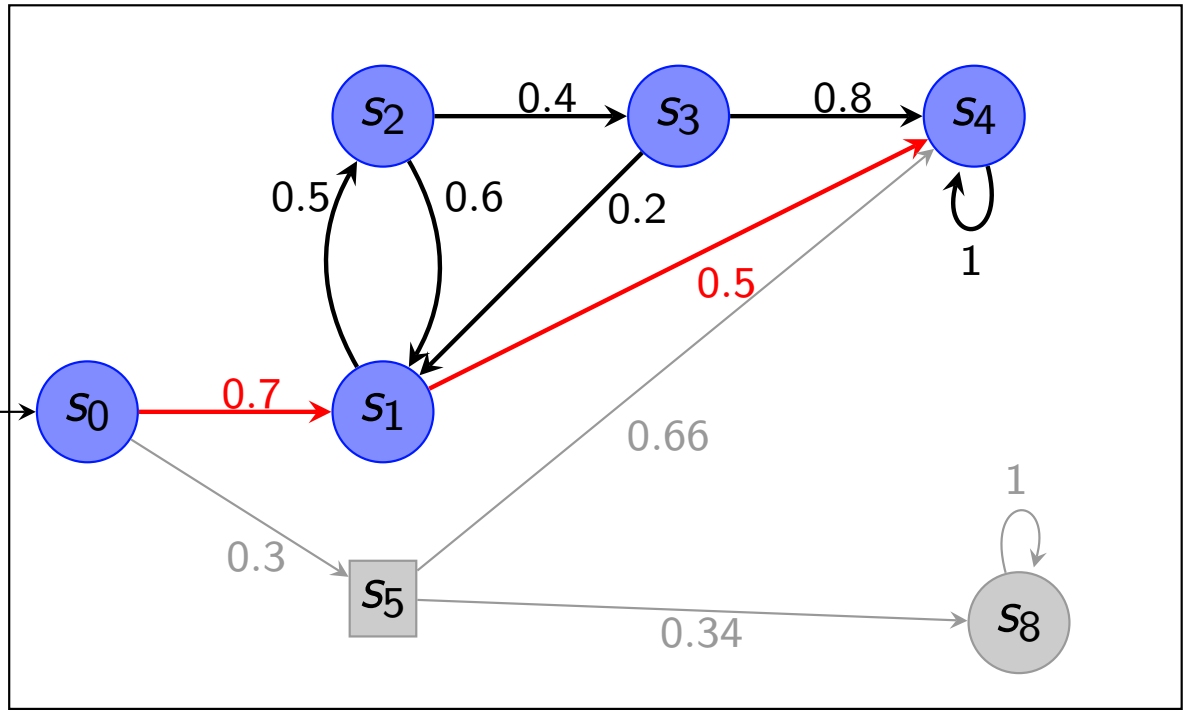
Hierarchical counterexamples for DTMCs - Global search

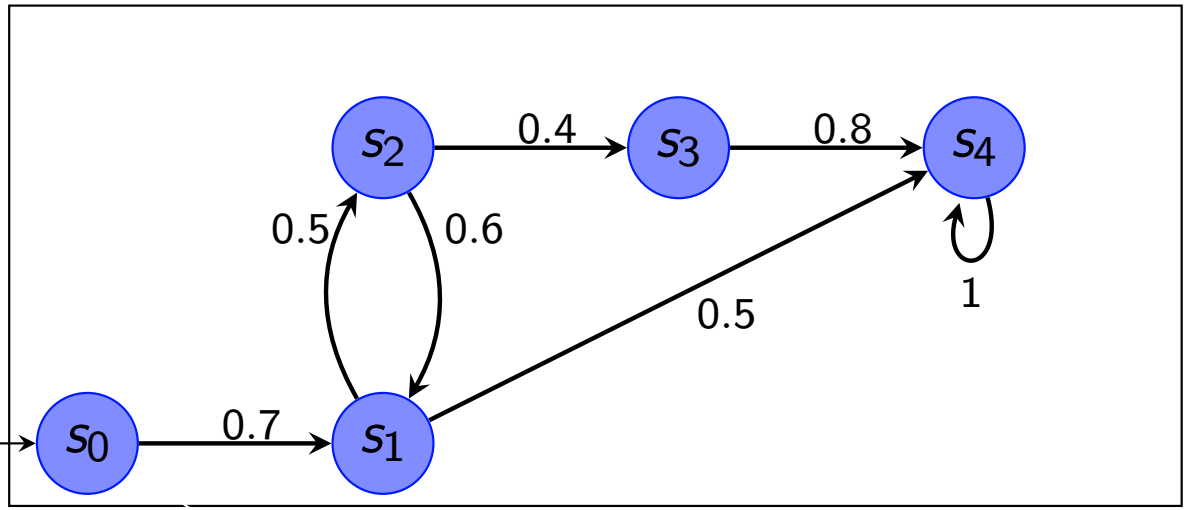




Search for next most probable path

Hierarchical counterexamples for DTMCs - Global search

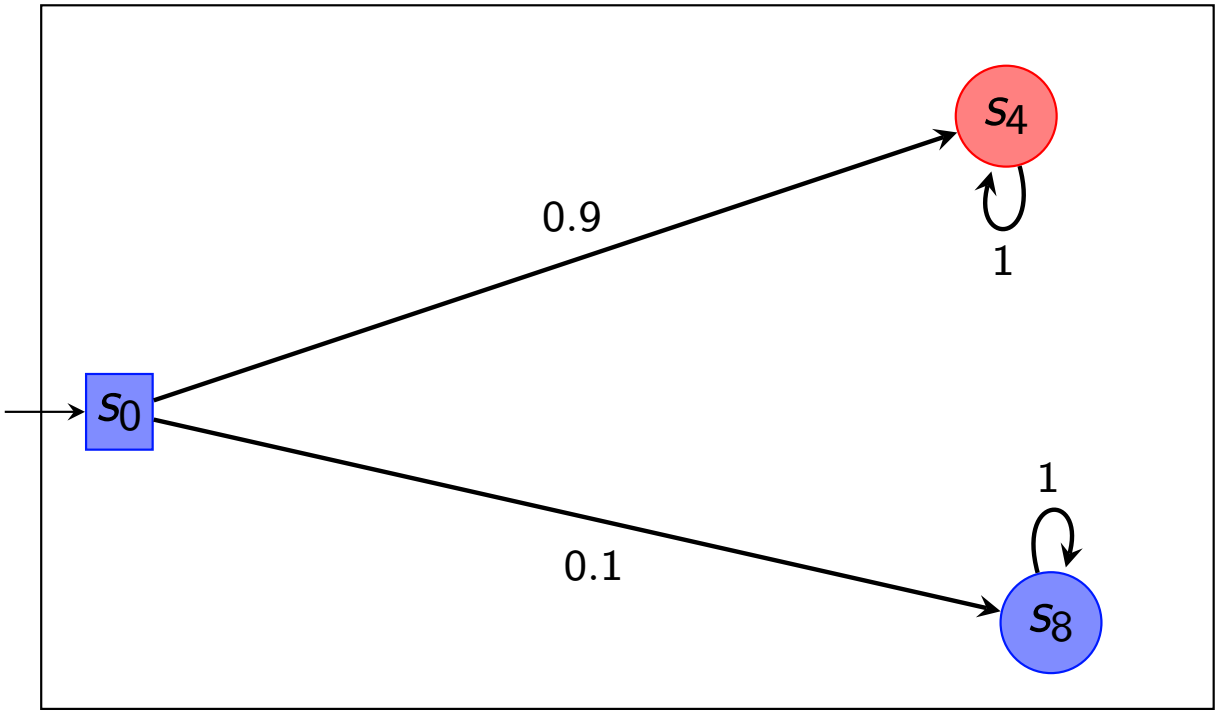




Critical subsystem updated

Probability 0.66

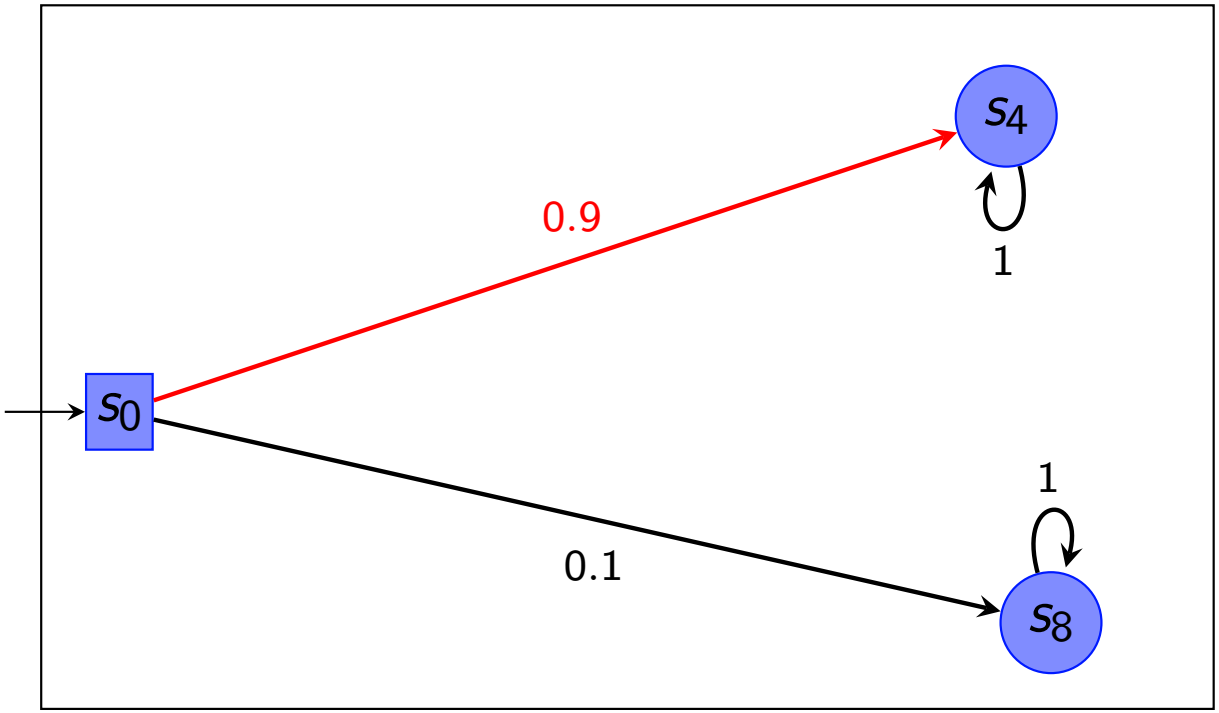
Hierarchical counterexamples for DTMCs - Local search



Task: Find a critical subsystem that violates

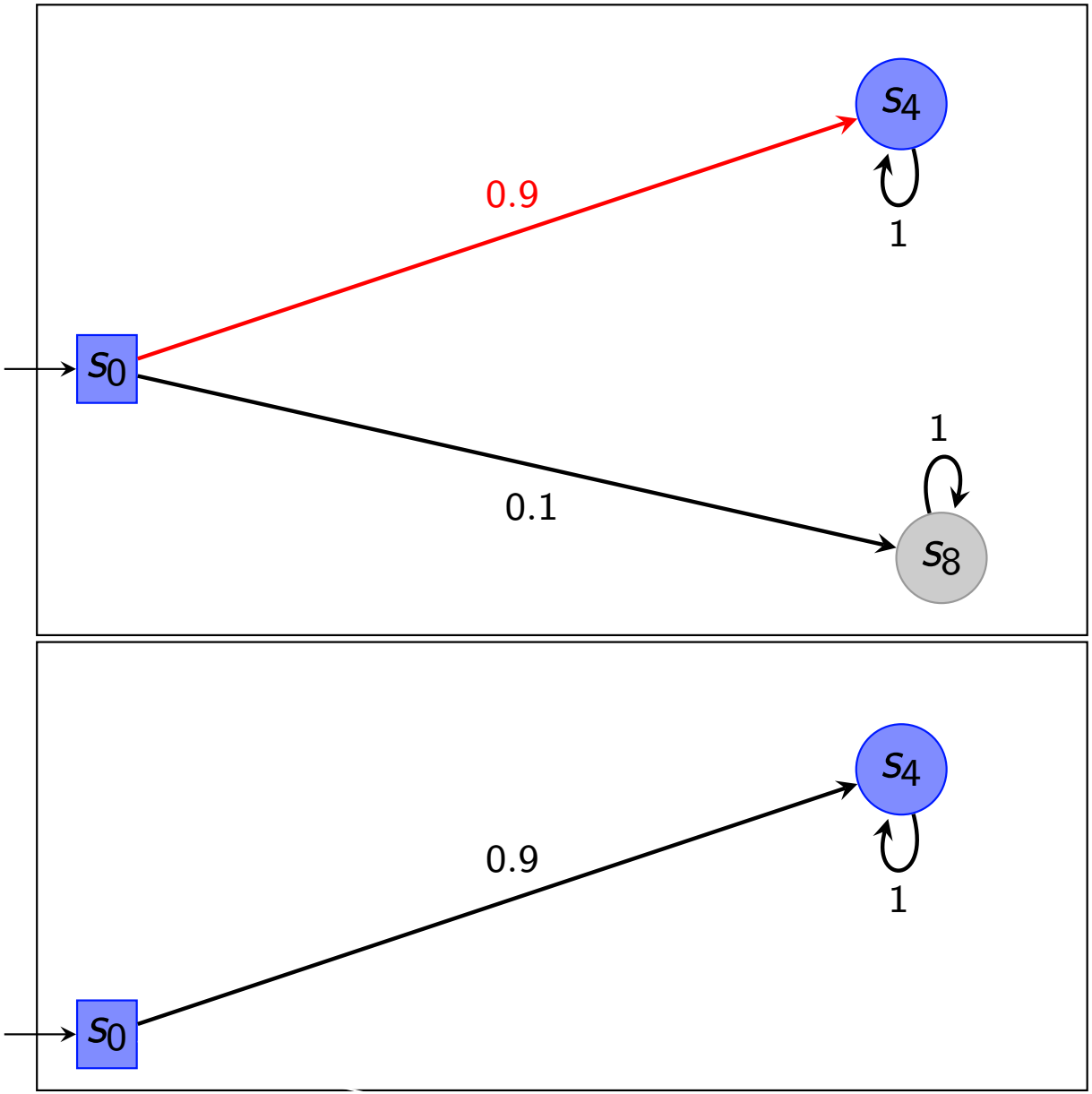
$$\mathbb{P}_{<0.5}(\Diamond 4)$$

Hierarchical counterexamples for DTMCs - Local search



Initial search

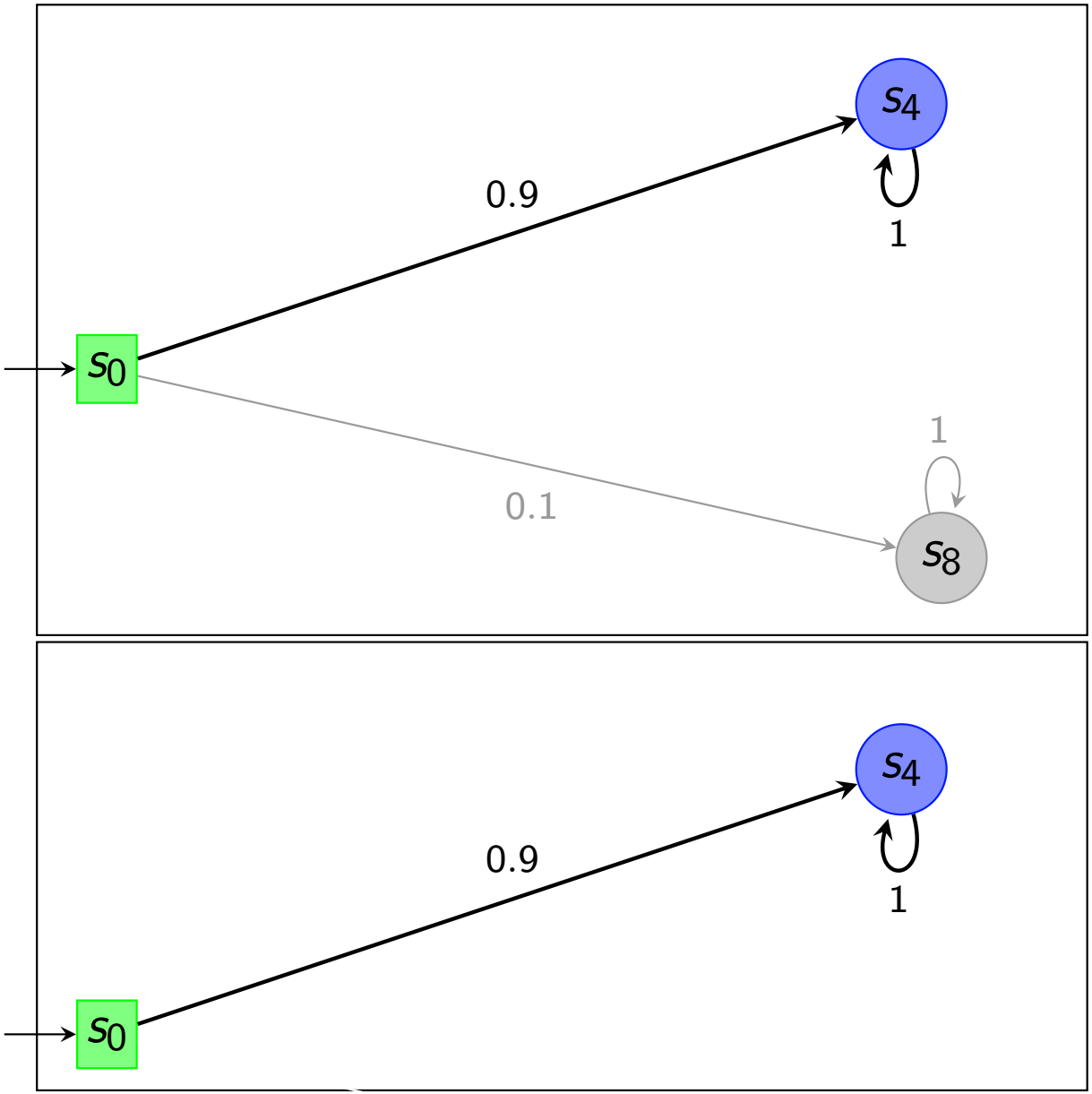
Hierarchical counterexamples for DTMCs - Local search



Critical subsystem

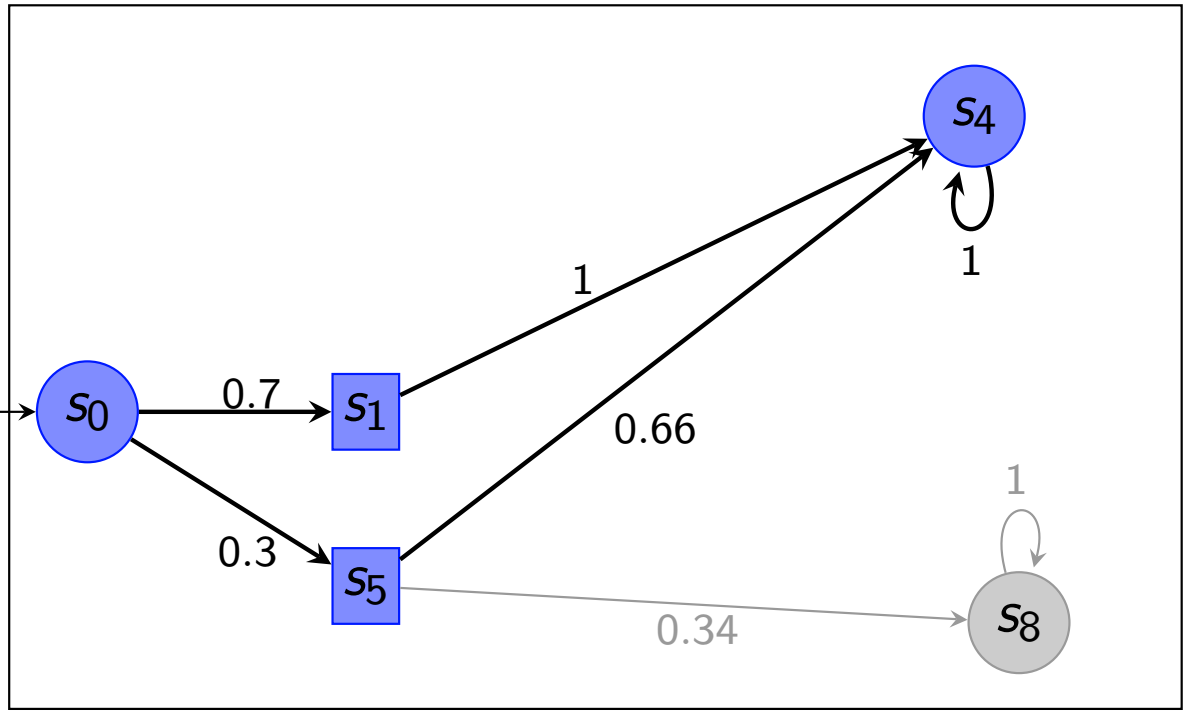
Probability 0.9

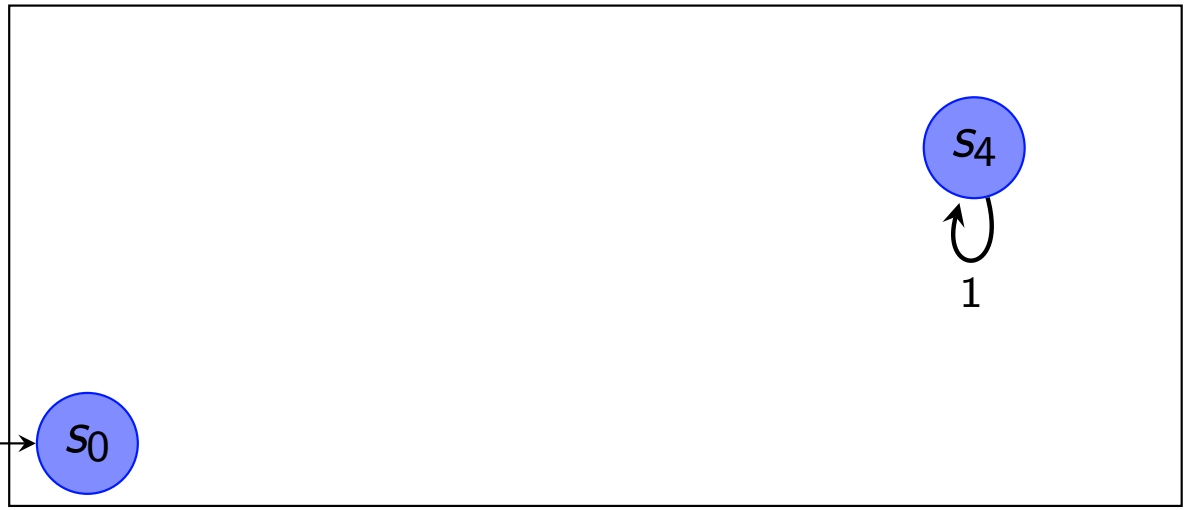
Hierarchical counterexamples for DTMCs - Local search



Choose state 0 for concretization

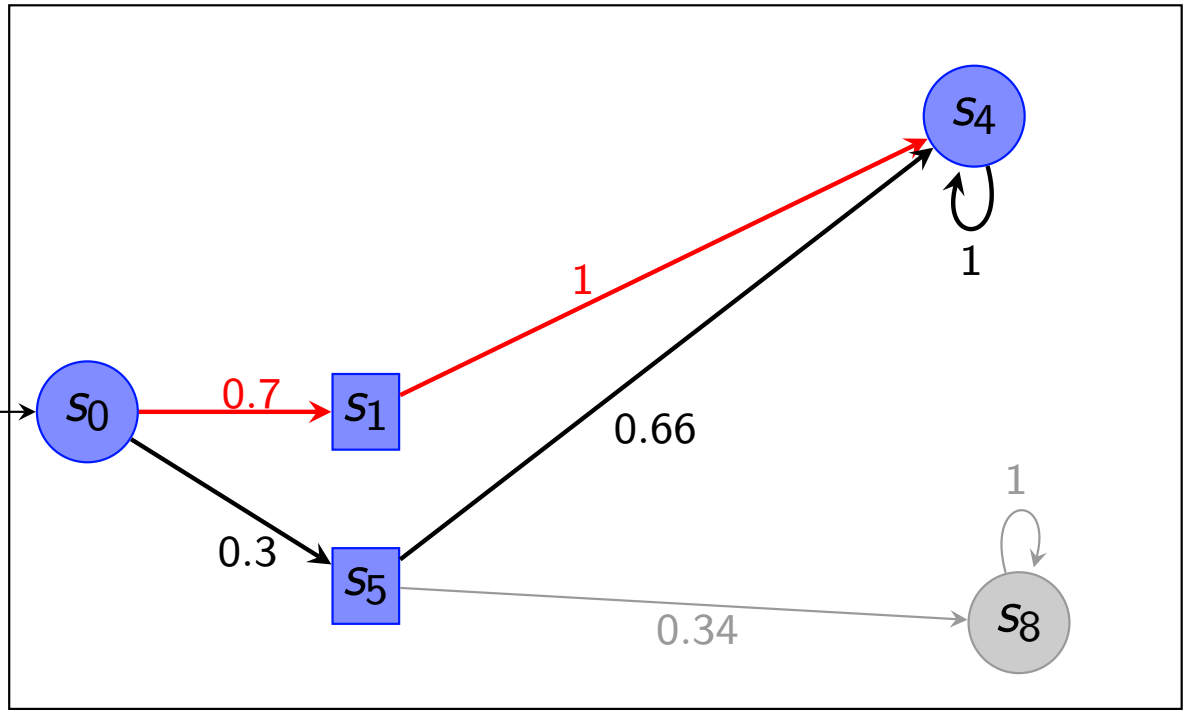
Hierarchical counterexamples for DTMCs - Local search

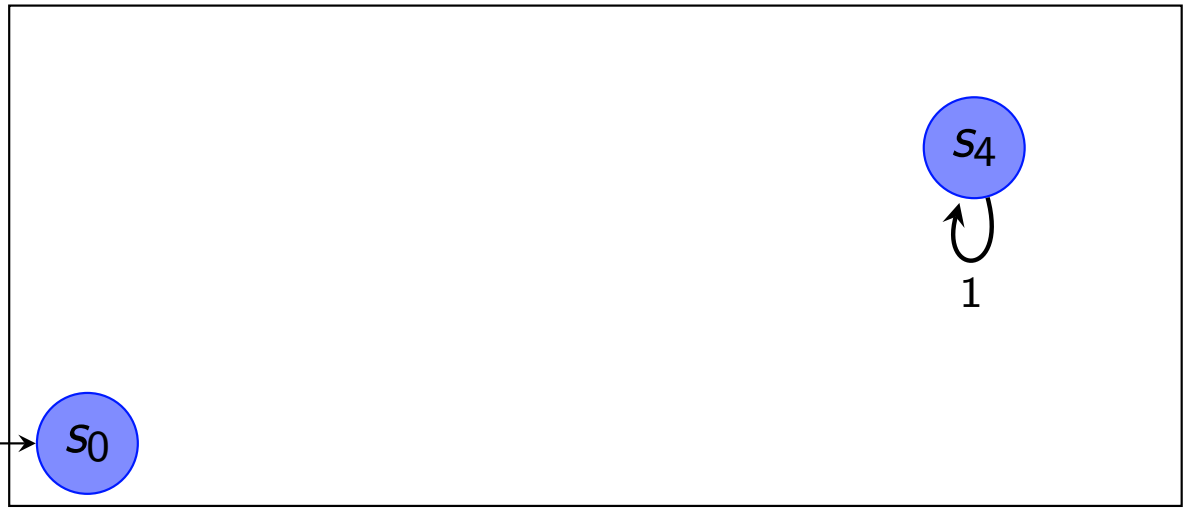




Concretize state 0

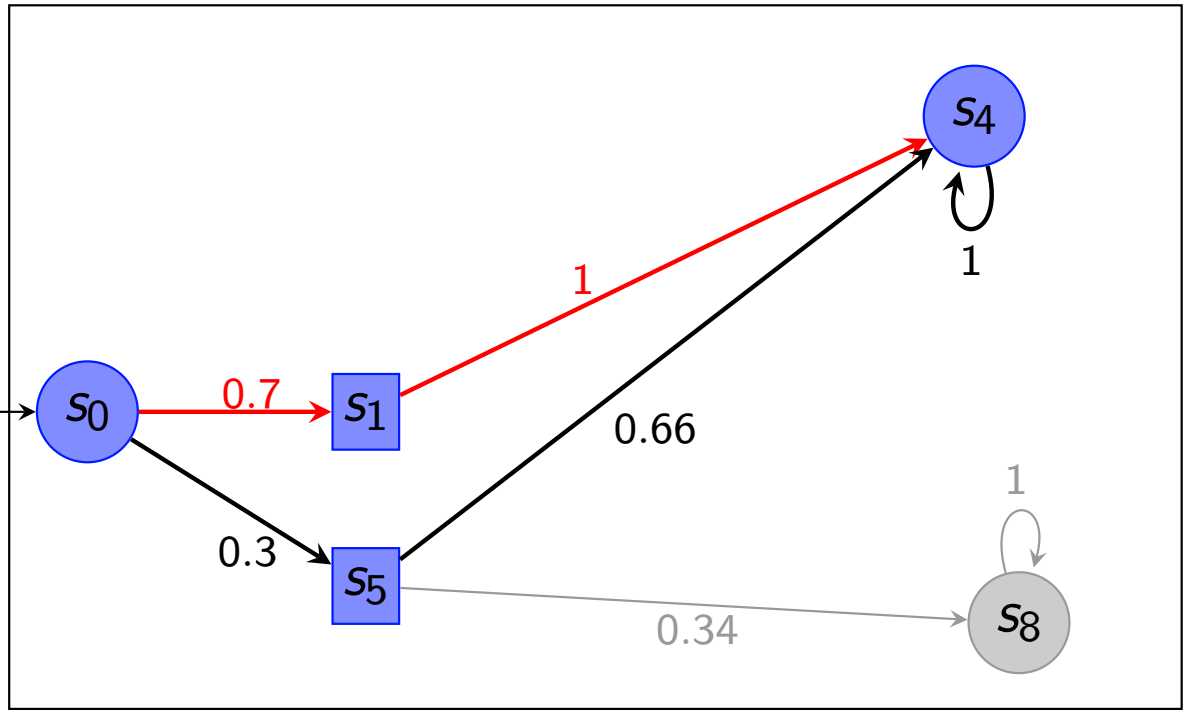
Hierarchical counterexamples for DTMCs - Local search

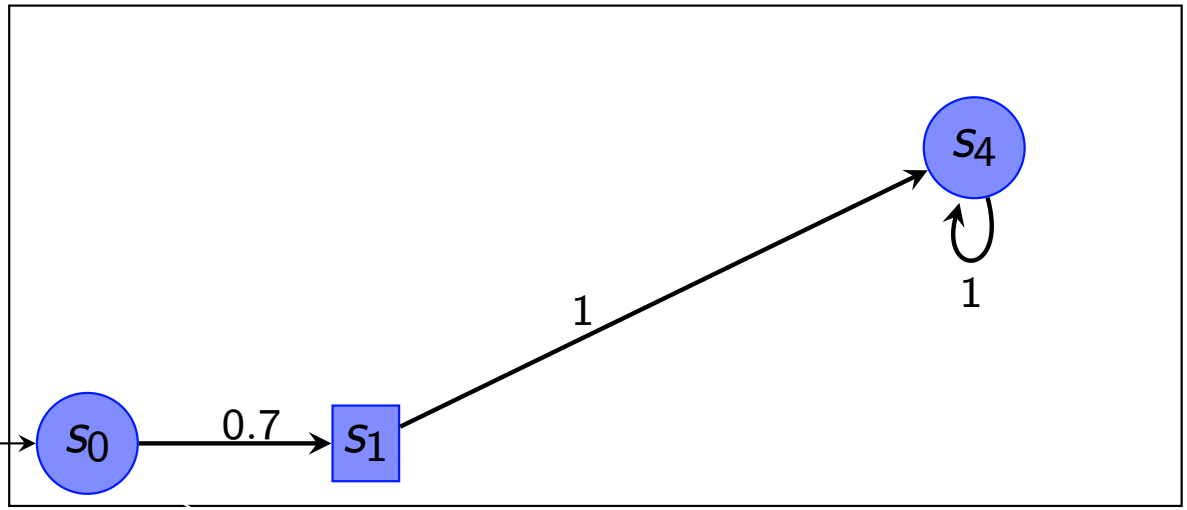




Search for next most probable path fragments connecting nodes of the subsystem

Hierarchical counterexamples for DTMCs - Local search

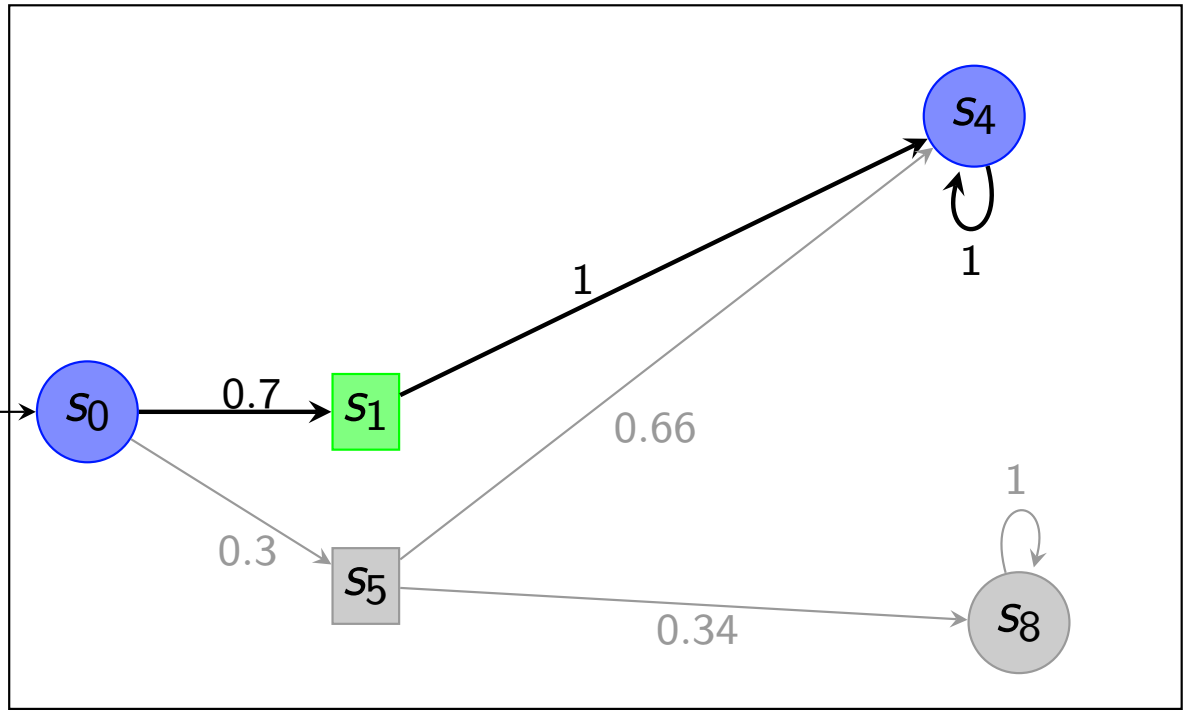


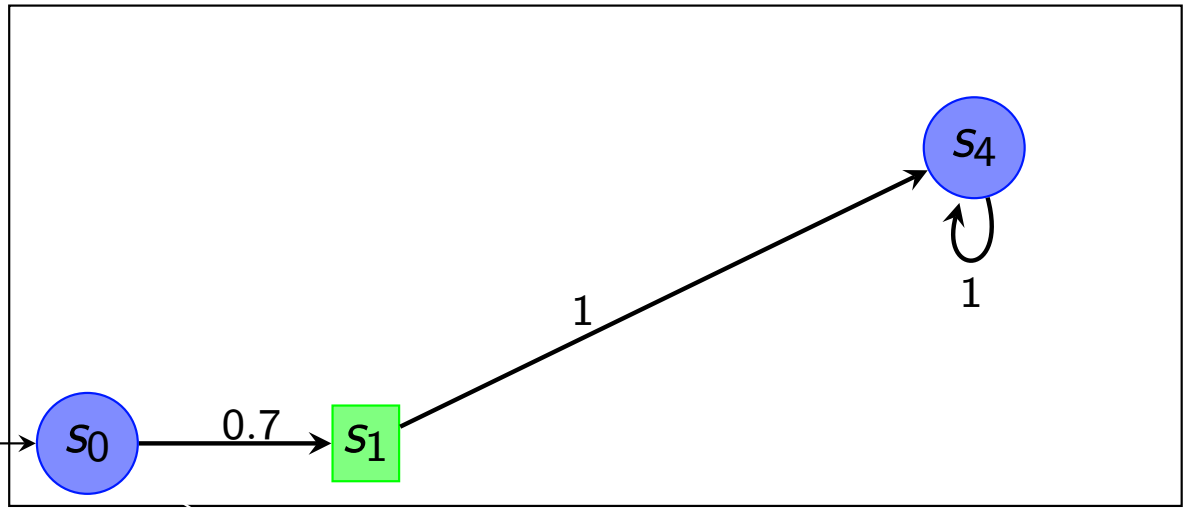


Critical subsystem updated

Probability 0.7

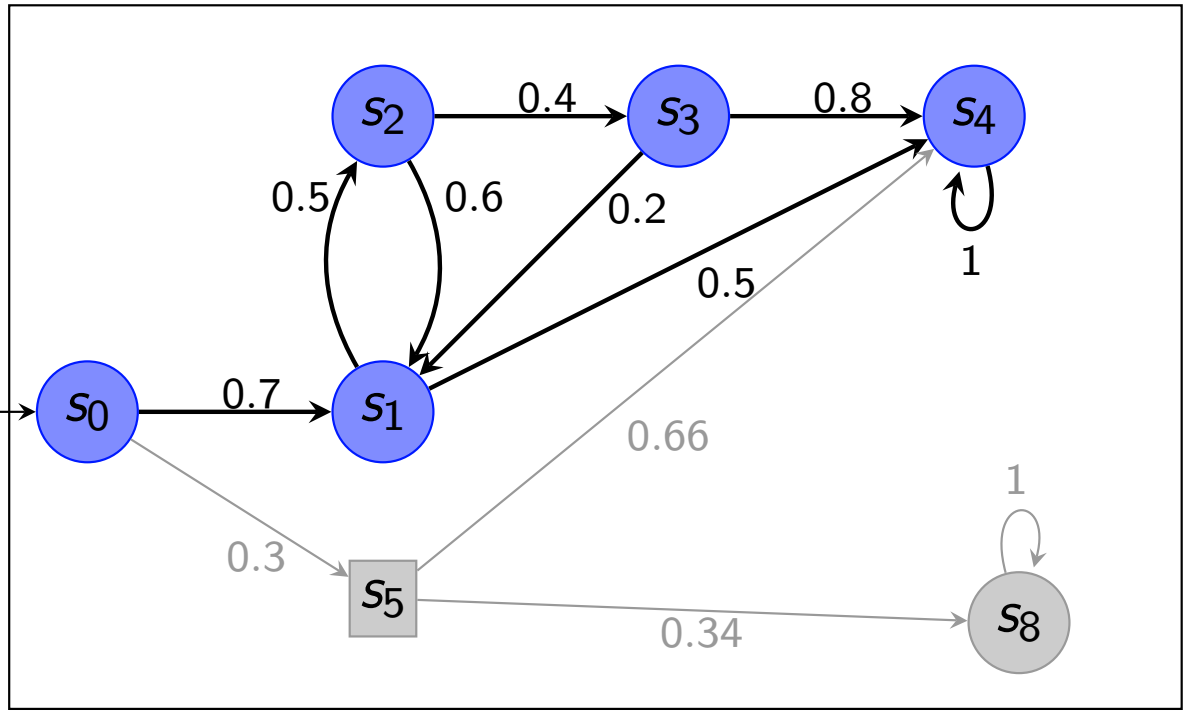
Hierarchical counterexamples for DTMCs - Local search

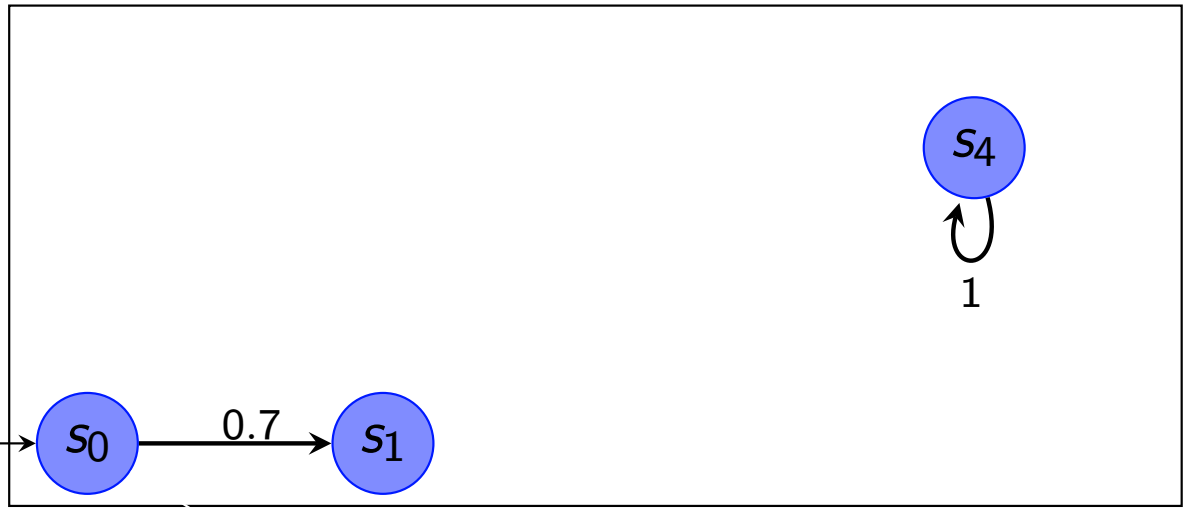




Choose state 1 for concretization

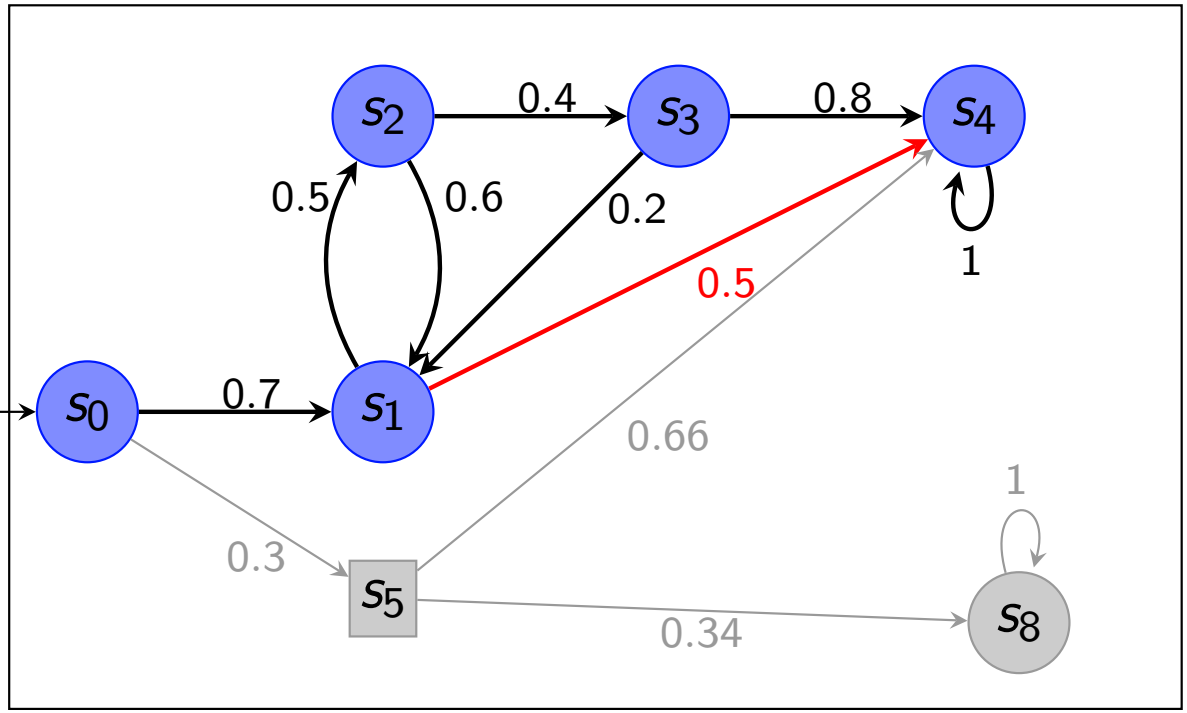
Hierarchical counterexamples for DTMCs - Local search

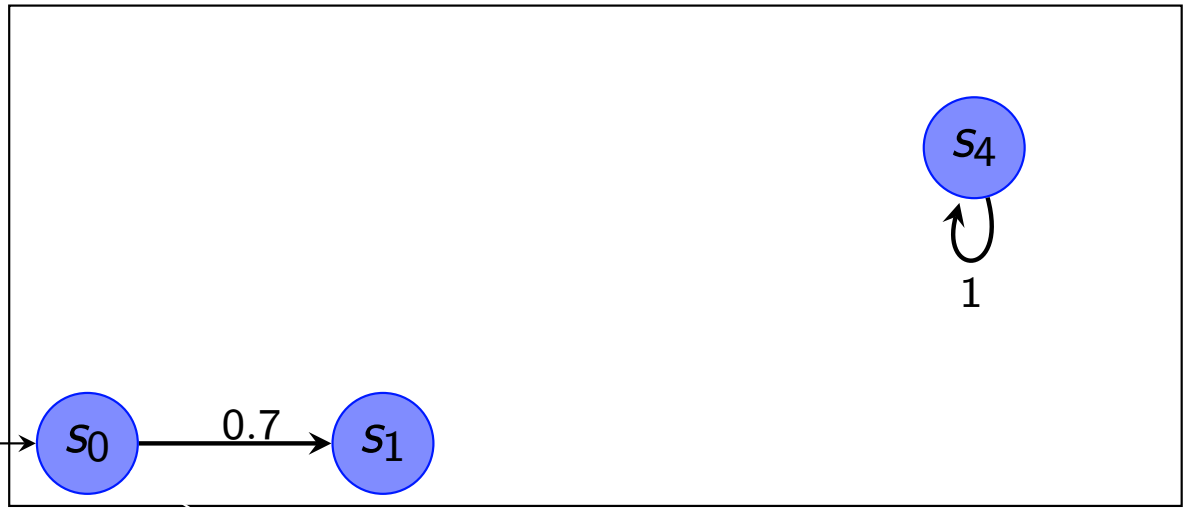




Concretize state 1

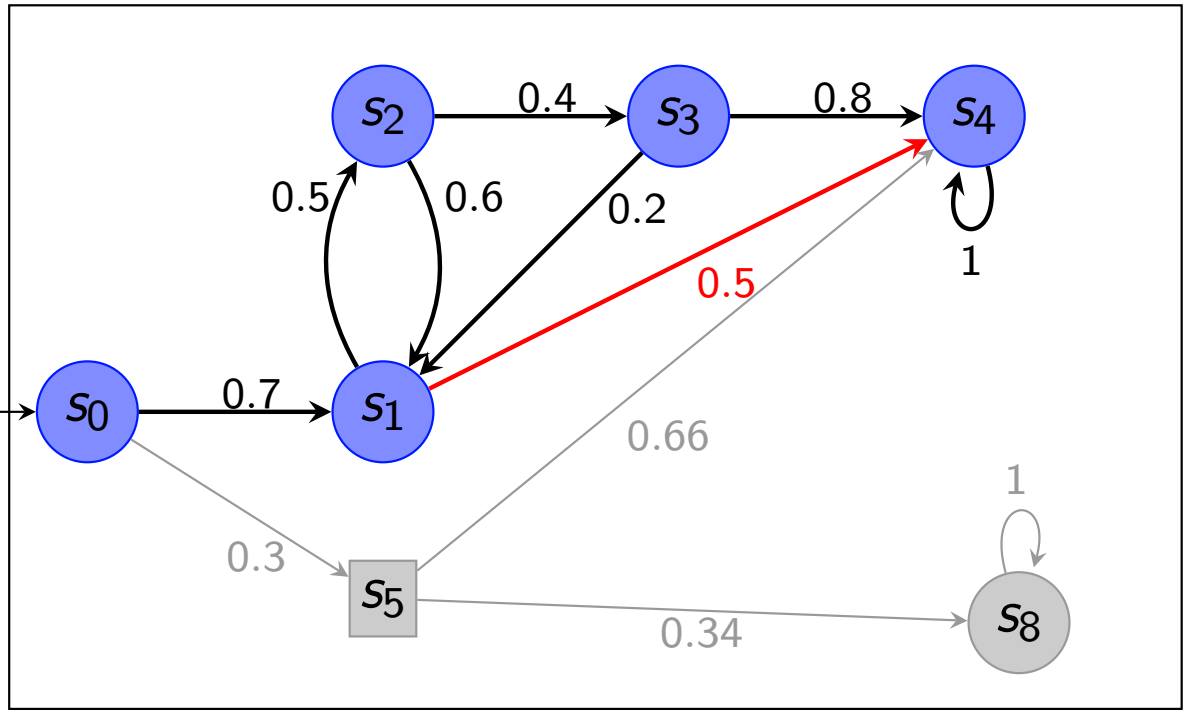
Hierarchical counterexamples for DTMCs - Local search

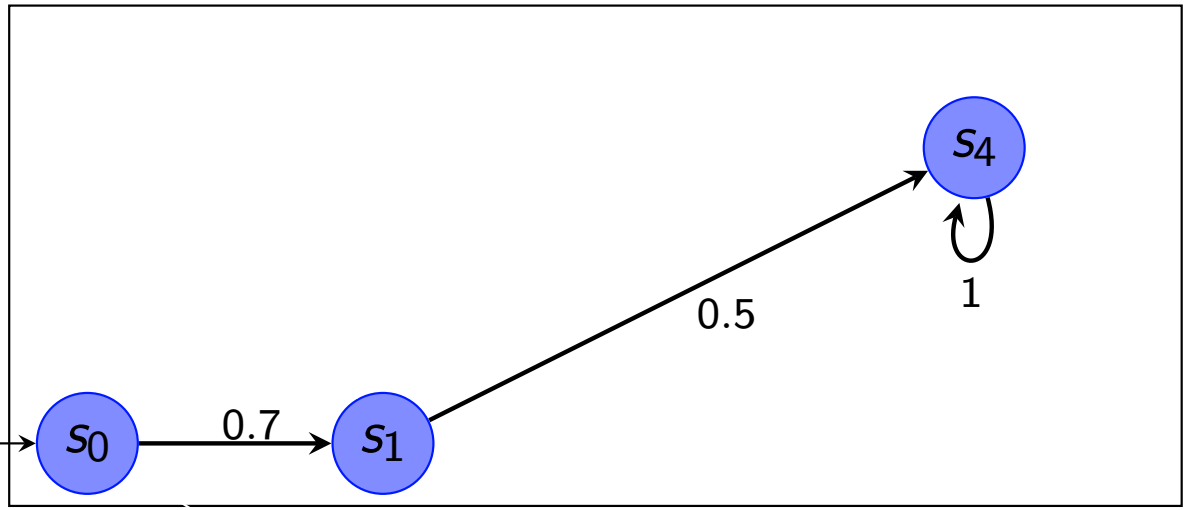




Search for next most probable path fragments connecting nodes of the subsystem

Hierarchical counterexamples for DTMCs - Local search

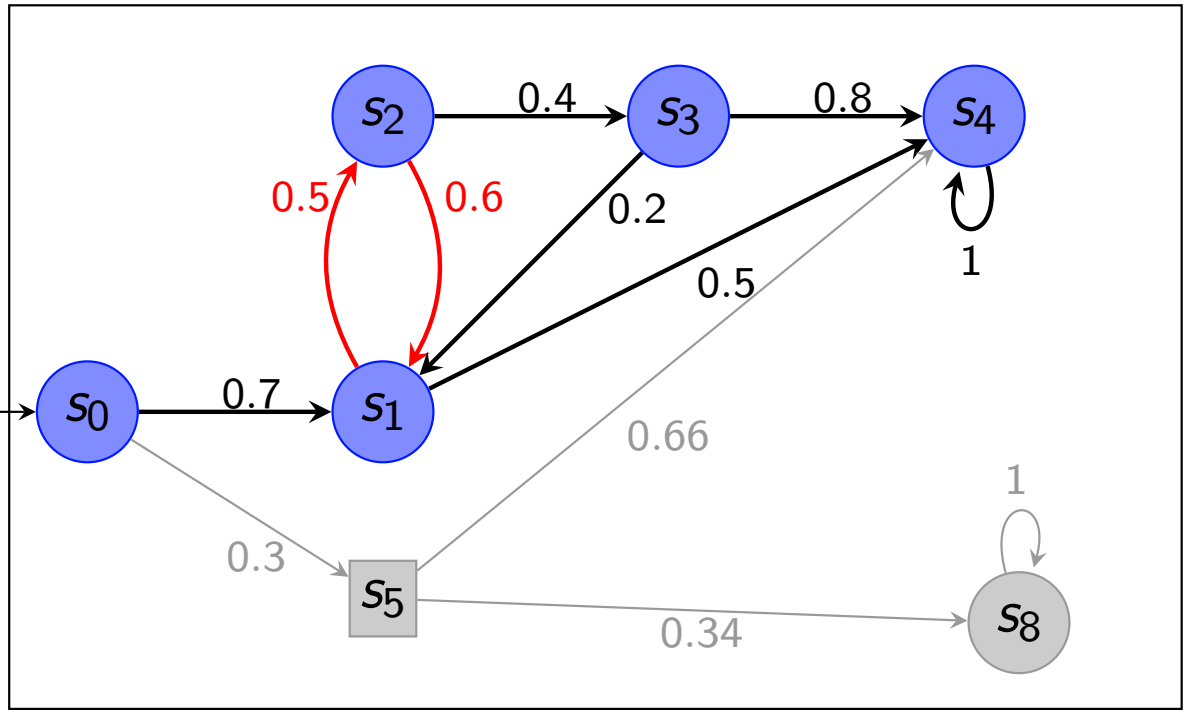


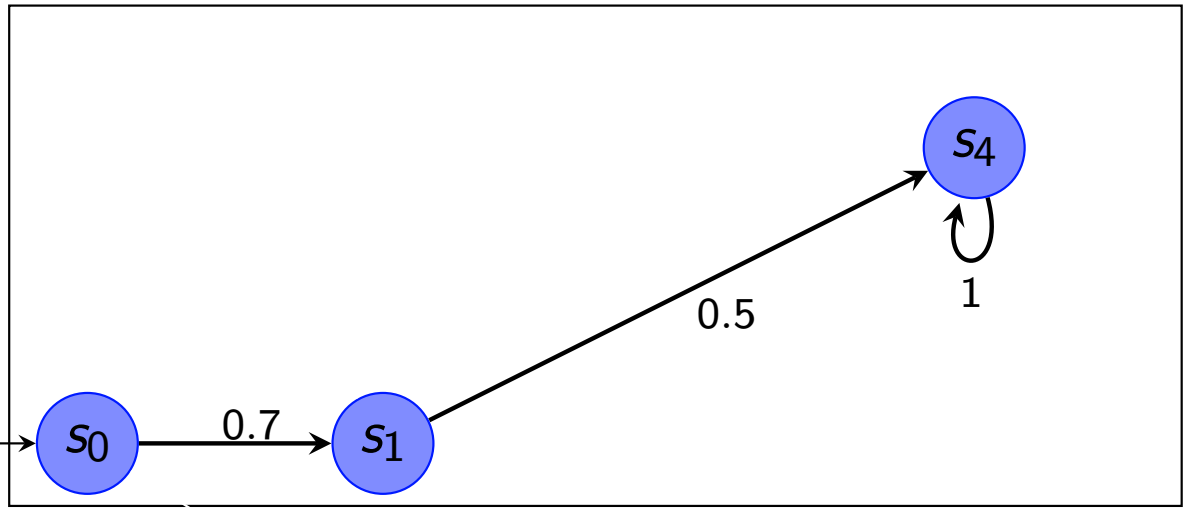


Critical subsystem updated

Probability 0.35

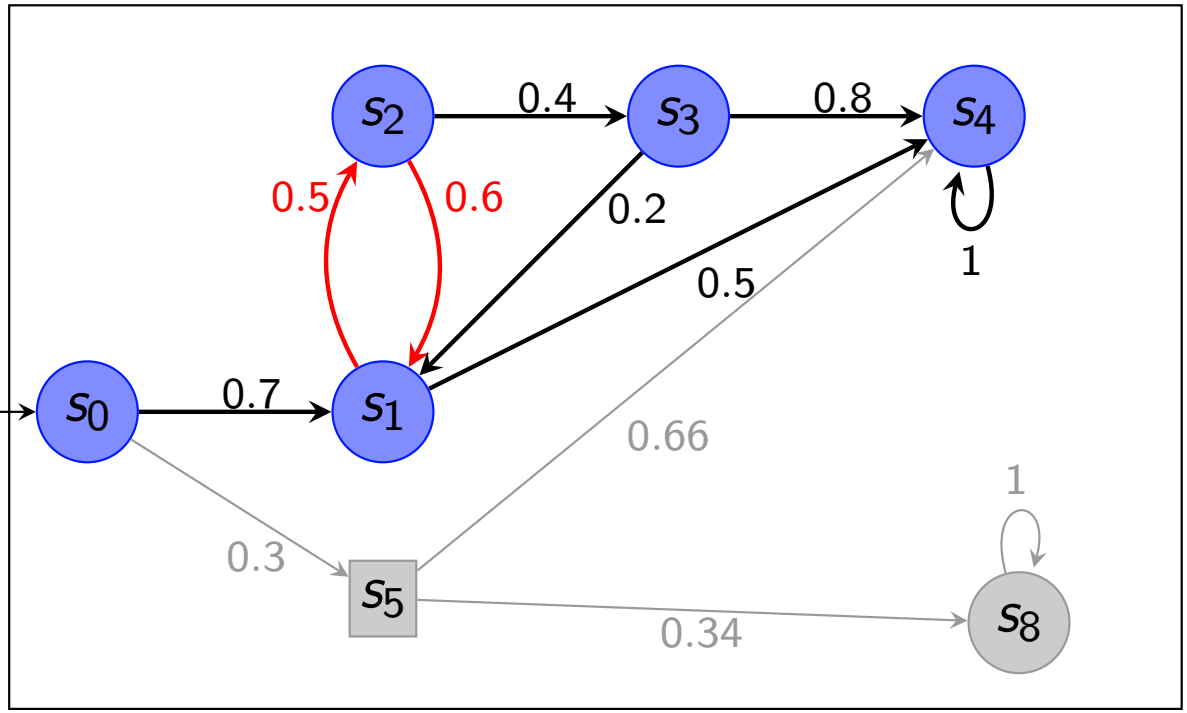
Hierarchical counterexamples for DTMCs - Local search

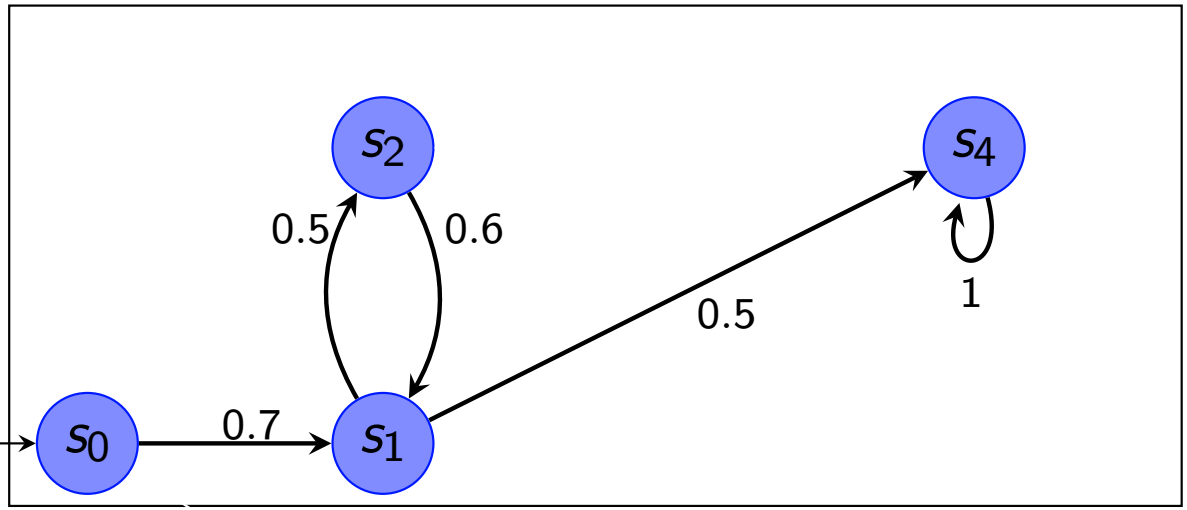




Search for next most probable path fragments connecting nodes of the subsystem

Hierarchical counterexamples for DTMCs - Local search



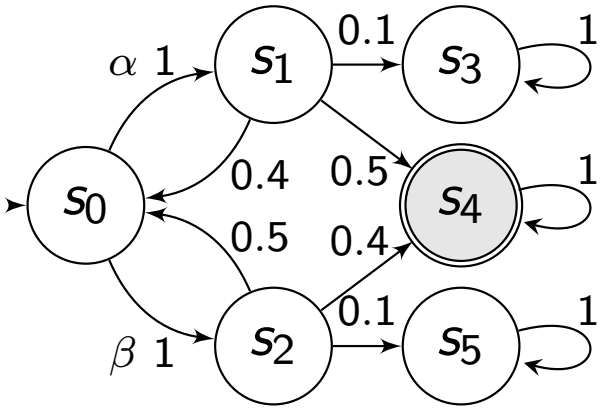


Critical subsystem updated

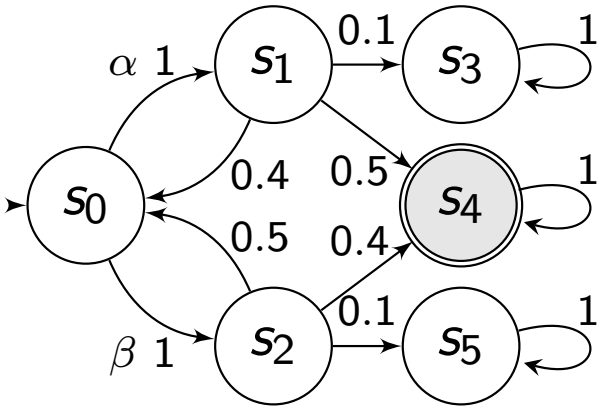
Probability 0.538

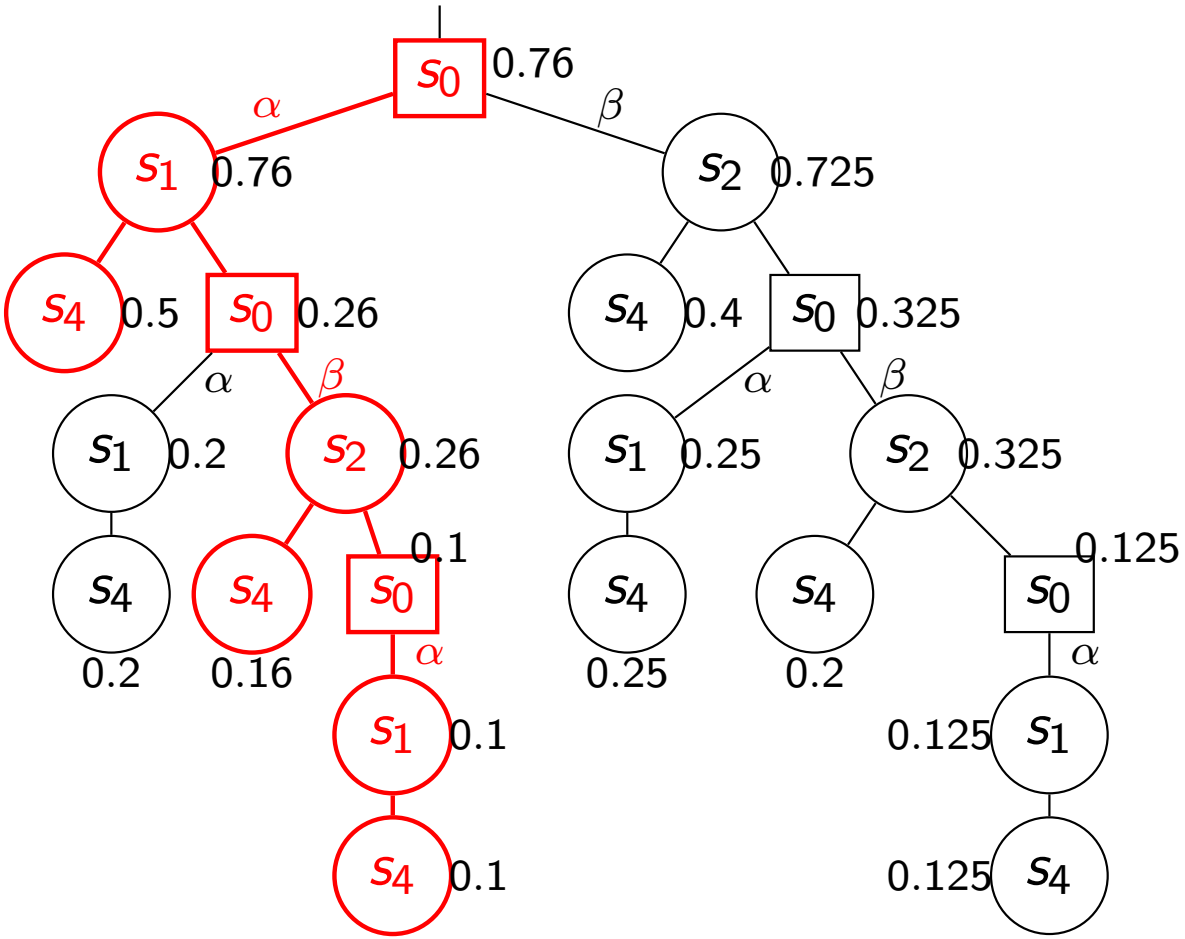
Path-based counterexamples for MDPs [Aljazzar, Leue]

Path-based counterexamples for MDPs [Aljazzar, Leue]



Path-based counterexamples for MDPs [Aljazzar, Leue]





Minimal counterexamples

Another approach

Another approach

Goal

Compute a critical subsystem with a minimal number of states.

Possible approaches:

- SAT-modulo-theories solving
- Mixed integer linear programming

MILP formulation for DTMCs

MILP formulation for DTMCs

- $1 - \lambda$: probability bound
- $x_s \in \{0, 1\} \subseteq \mathbb{Z}$ with $x_s = 1$ iff s belongs to the subsystem
- $p_s \in [0, 1] \subseteq \mathbb{R}$: probability of state s within the subsystem

MILP formulation for DTMCs

- $1 - \lambda$: probability bound
- $x_s \in \{0, 1\} \subseteq \mathbb{Z}$ with $x_s = 1$ iff s belongs to the subsystem
- $p_s \in [0, 1] \subseteq \mathbb{R}$: probability of state s within the subsystem

$$\min\left(-\tfrac{1}{2} p_{s_{\text{init}}} + \textstyle\sum_{s \in S} x_s\right) \text{ such that}$$

$$p_{s_{\text{init}}} > 1 - \lambda$$

$$\forall s \in T : x_s = p_s$$

$$\forall s \in S \setminus T : p_s \leq x_s \qquad p_s \leq \textstyle\sum_{s' \in S} P(s, s') \cdot p_{s'}$$

MILP formulation for MDPs

MILP formulation for MDPs

- $\sigma_{s,a} \in [0,1] \subseteq \mathbb{Z}$: encoding of the scheduler

MILP formulation for MDPs

- $\sigma_{s,a} \in [0,1] \subseteq \mathbb{Z}$: encoding of the scheduler

$$\min\left(-\tfrac{1}{2} p_{s_{\text{init}}} + \sum_{s \in S} x_s\right) \text{ such that}$$

$$p_{s_{\text{init}}} > 1 - \lambda$$

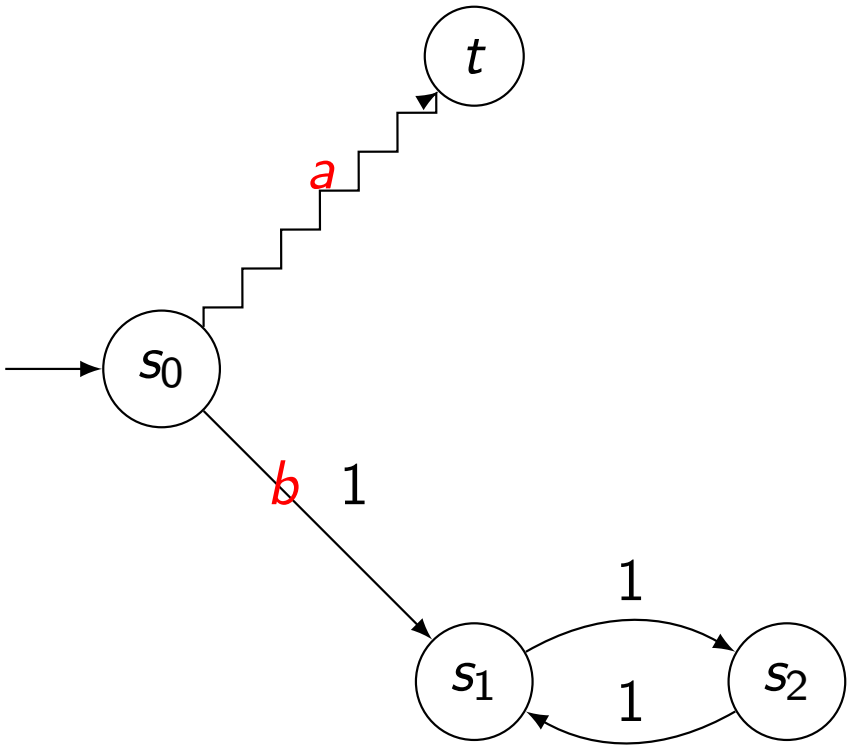
$$\textit{targets} : x_s = p_s$$

$$\textit{non-target } s : p_s \leq x_s \qquad x_s = \sum_{a \in A} \sigma_{s,a}$$

$$\textit{non-target } s,\ \textit{action } a : p_s \leq (1 - \sigma_{s,a}) + \sum_{s' \in S} P(s,a,s') \cdot p_{s'}$$

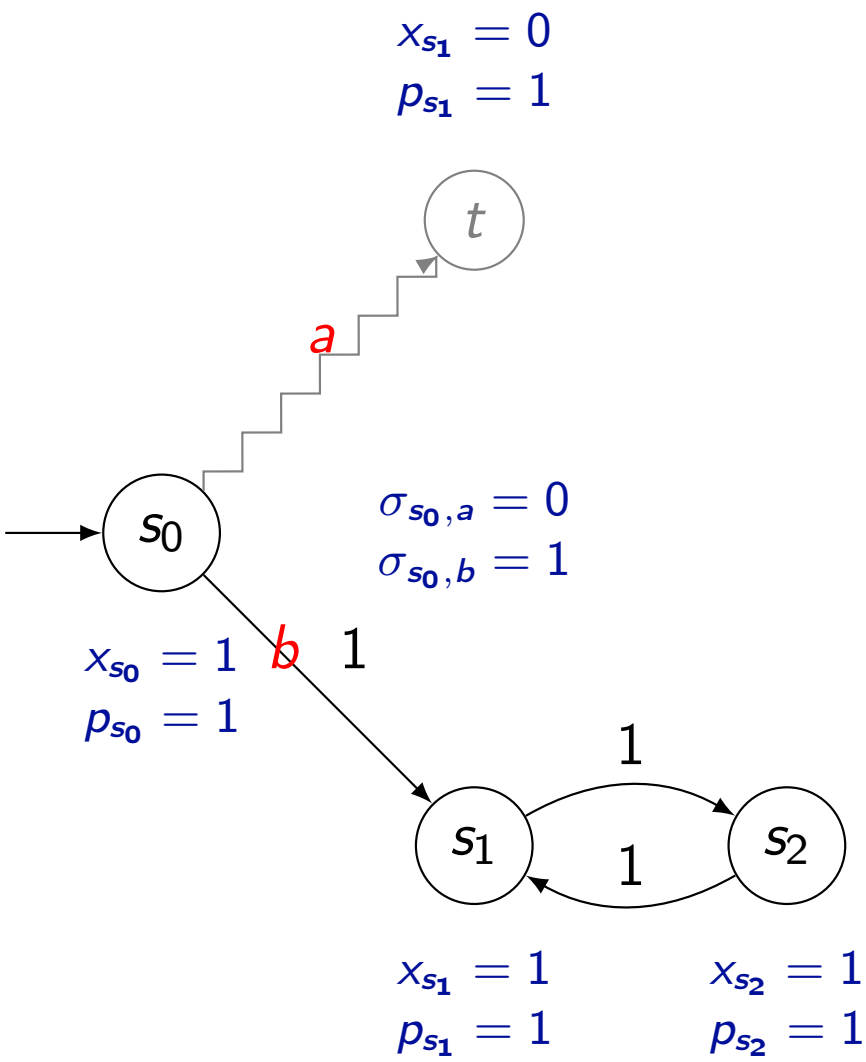
MILP formulation for MDPs: Problematic states

- $\sigma_{s,a} \in [0,1] \subseteq \mathbb{Z}$: encoding of the scheduler



MILP formulation for MDPs: Problematic states

- $\sigma_{s,a} \in [0,1] \subseteq \mathbb{Z}$: encoding of the scheduler



MILP formulation for MDPs

- $\sigma_{s,a} \in [0,1] \subseteq \mathbb{Z}$: encoding of the scheduler

$$\min\left(-\tfrac{1}{2} p_{s_{\text{init}}} + \sum_{s \in S} x_s\right) \text{ such that}$$

$$p_{s_{\text{init}}} > 1 - \lambda$$

$$\textit{targets} : x_s = p_s$$

$$\textit{non-target } s : p_s \leq x_s \qquad x_s = \sum_{a \in A} \sigma_{s,a}$$

$$\textit{non-target } s, \ \textit{action } a : p_s \leq (1 - \sigma_{s,a}) + \sum_{s' \in S} P(s,a,s') \cdot p_{s'}$$

$$\textit{probl. } s,\ s' \in \textit{succ}(s,a) : 2t_{s,s'} \leq x_s + x_{s'}$$

$$r_s < r_{s'} + (1 - t_{s,s'})$$

$$(1 - x_s) + (1 - \sigma_{s,a}) + \sum_{s' \in \textit{succ}(s,a)} t_{s,s'} \geq 1$$

MILP approach for minimal critical command sets

MILP approach for minimal critical command sets

- $x_c \in \{0, 1\} \subseteq \mathbb{Z}$: selecting commands

MILP approach for minimal critical command sets

- $x_c \in \{0,1\} \subseteq \mathbb{Z}$: selecting commands

$$\min\left(-\tfrac{1}{2} p_{s_{\text{init}}} + \sum_{c \in C} x_c\right) \text{ such that}$$

$$p_{s_{\text{init}}} > 1 - \lambda$$

$$\textit{targets} : p_s = 1$$

$$\textit{non-target } s : p_s \leq \sum_{a \in A} \sigma_{s,a}$$

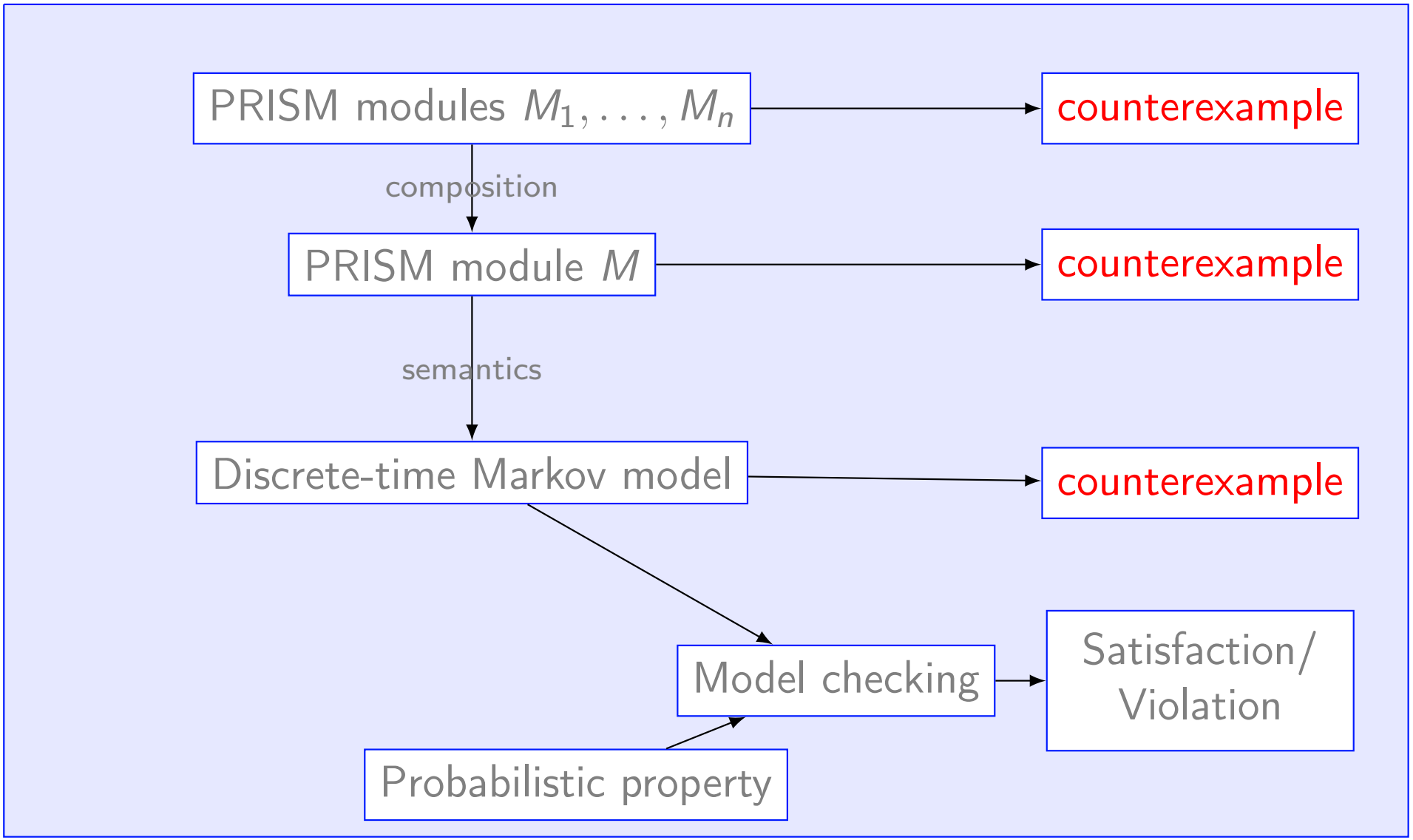
$$\textit{non-target } s,\ \textit{action } a : p_s \leq (1 - \sigma_{s,a}) + \sum_{s' \in S} P(s,a,s') \cdot p_{s'}$$

$$\textit{non-target } s,\ \textit{action } a : \sigma_{s,a} \leq x_c$$

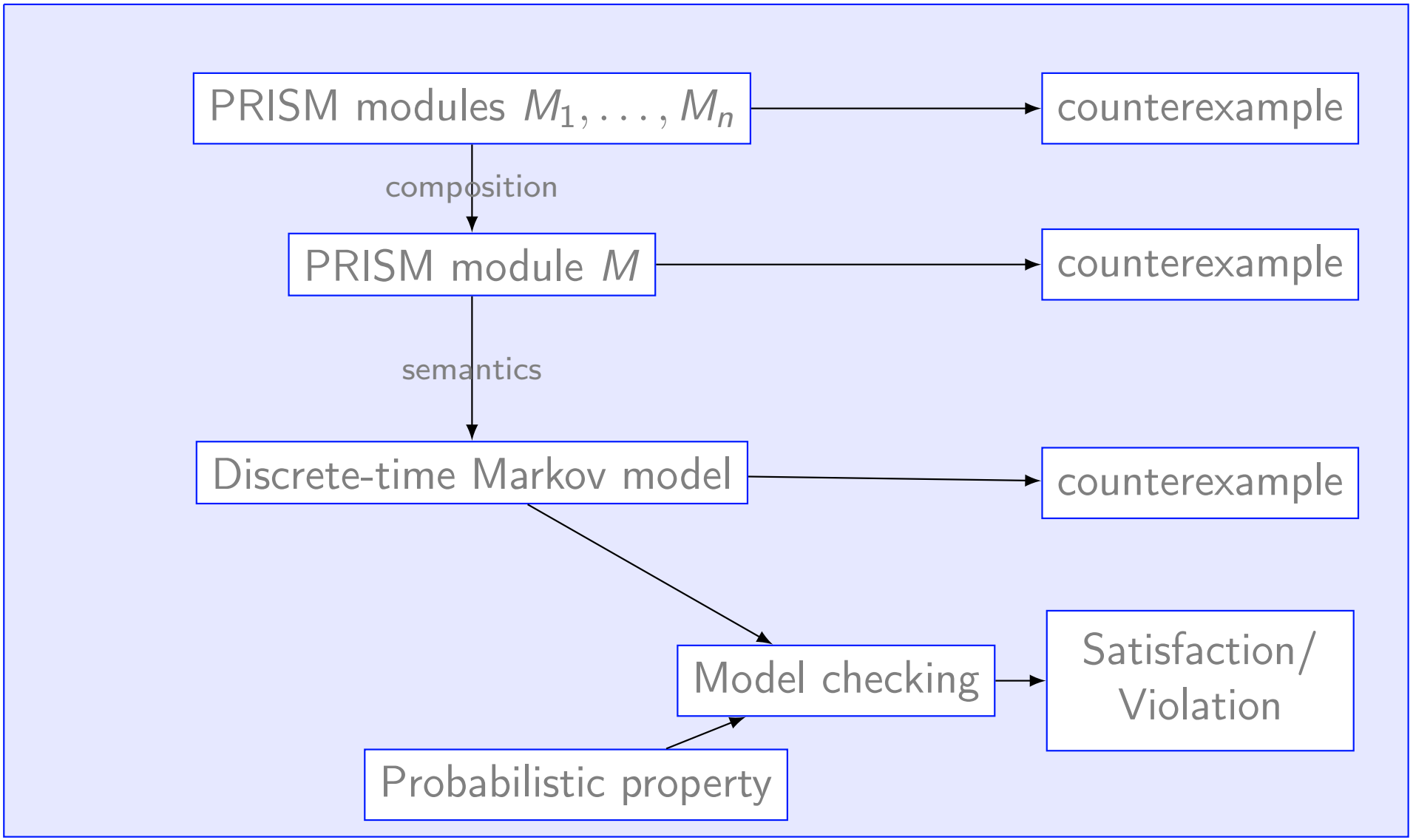
$$\textit{probl. } s,\ \textit{action } a : \sigma_{s,a} \leq \sum_{s' \in succ(s,a)} t_{s,s'}$$

$$\textit{probl. } s,\ s' \in succ(s,a) : r_s < r_{s'} + (1 - t_{s,s'})$$

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property

Contents


PRISM modules $M_1, \ldots, M_n$
counterexample
composition
PRISM module M
counterexample
semantics
Discrete-time Markov model
counterexample
Model checking
Satisfaction/
Violation
Probabilistic property