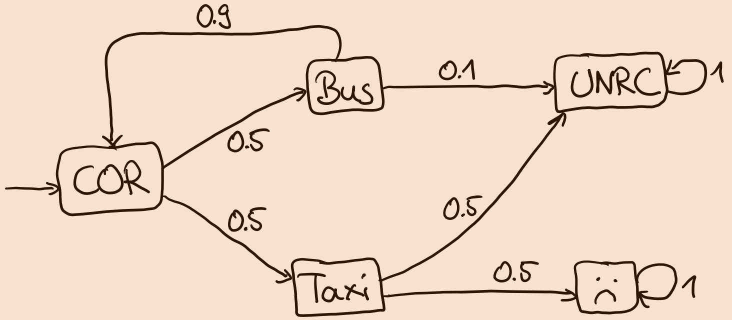
Formal Approaches to Decision-Making under Uncertainty

Lecture 3-1: Algorithms for DTMCs

Arnd Hartmanns
Formal Methods and Tools
UNIVERSITY OF TWENTE

Reachability Probabilities

G = {UNRC}



The **probability to reach** a goal state in $G \subseteq S$

$$\mathrm{P}(\diamond\, G) \quad \text{or} \quad \mathrm{P}(\diamond^{\leq b}\, G)$$

is now easily defined as

$$\mathrm{P}(\diamond\, G) = P_M(\{s_0\, s_1 \ldots \in \mathrm{Paths}(M) \mid \exists i{:}\, s_i \in G\})$$

$P(\diamond \{UNRC\}) = 0.5 \cdot 0.5 + 0.5 \cdot 0.1 + 0.5 \cdot 0.9 \cdot 0.5 \cdot 0.5 + \ldots$ $\quad P(\diamond \{Bus\}) = 0.5$

and $\mathrm{P}\left(\diamond^{\leq b}\, G\right) = P_M(\{s_0\, s_1 \ldots \in \mathrm{Paths}(M) \mid \exists i \leq b{:}\, s_i \in G\})$

DTMC Reachability

DTMC $\langle S, T, s_I \rangle$

– want to compute $P(\Diamond^{\leq b} G)$ and $P(\Diamond G)$

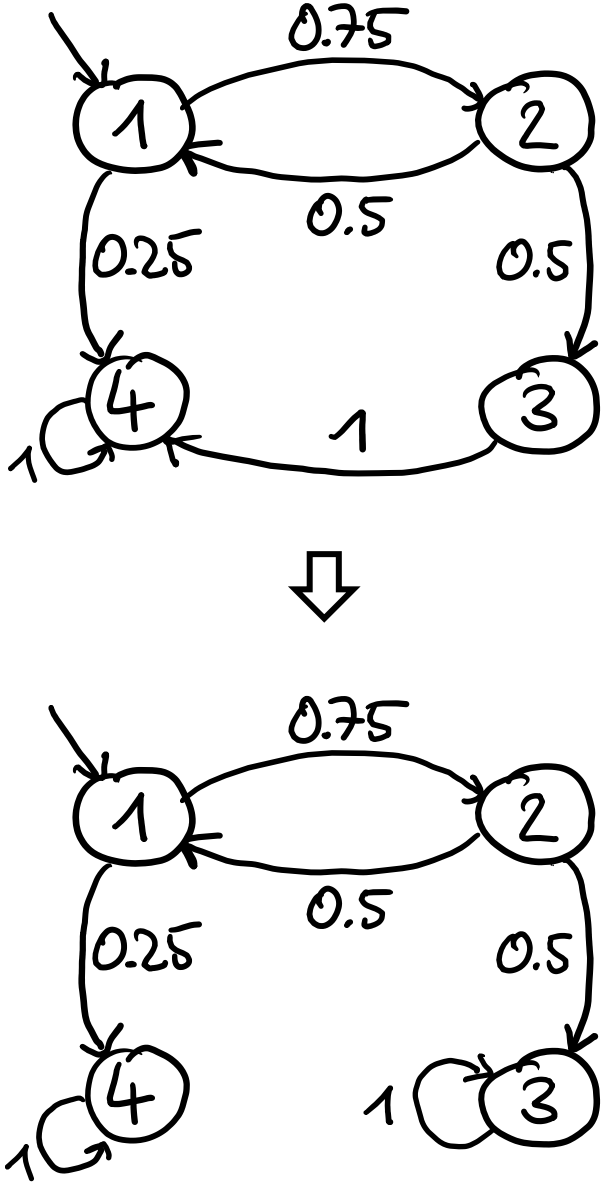
Preprocessing:

1. Make all states in G absorbing.
2. Compute state set S_0 where $P(\Diamond G)$ from $s = 0$

   → graph reachability of G

Example: $G = \{ 3 \}$

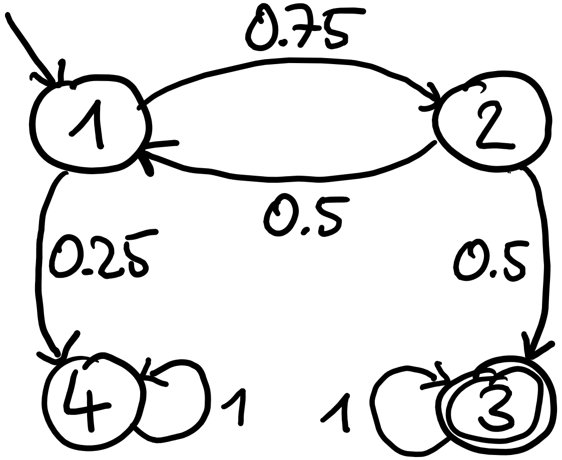
$S_0 = \{ 4 \}$



DTMC Reachability

DTMC $\langle S, T, s_I \rangle$ with probability matrix $\boldsymbol{P}$

– want to compute $\mathrm{P}(\lozenge^{\leq b}\, G)$



Let $x_i[s] \overset{\text{def}}{=} \mathrm{P}\left(\lozenge^{\leq i}\, G\right)$ from s.

Then:

$x_0[s] = 1$ if $s \in G$ else 0

$x_i[s] = \sum_{s'} T(s, s') \cdot x_{i-1}[s']$

$$\boldsymbol{P} = \begin{pmatrix} 0 & 0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

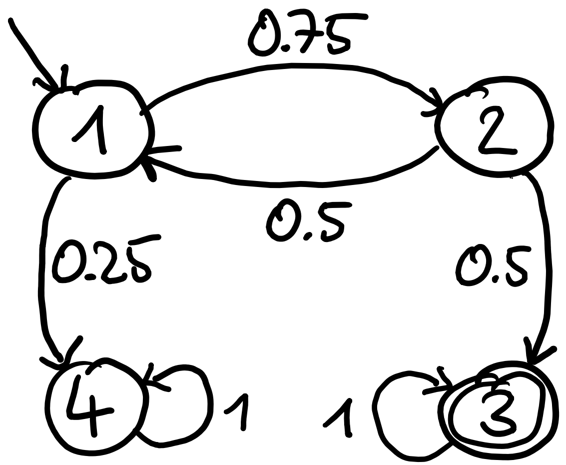
→ can formulate as matrix-vector multiplications:

$$\boldsymbol{x}_i = \boldsymbol{P}^i \boldsymbol{x}_0$$

DTMC Reachability

DTMC $\langle S, T, s_I \rangle$ with probability matrix $\boldsymbol{P}$

– want to compute $\mathrm{P}(\diamond\, G)$



Let $x[s] \stackrel{\text{def}}{=} \mathrm{P}(\diamond\, G)$ from s and $S_? \stackrel{\text{def}}{=} S \setminus (S_0 \cup G)$.

$$\boldsymbol{P} = \begin{pmatrix} 0 & 0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Then:

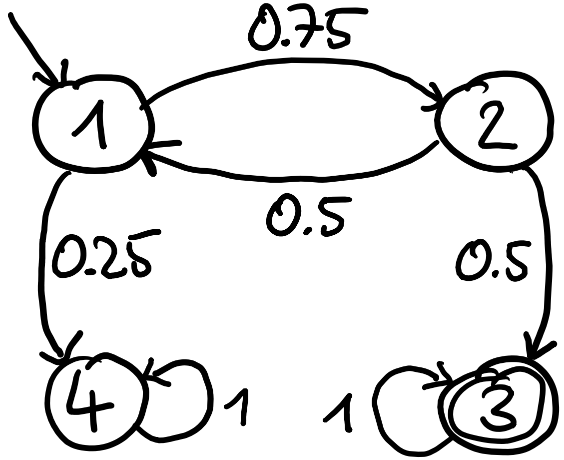
$x[s] = 1$ if $s \in G$

$x[s] = 0$ if $s \in S_0$

$x[s] = \sum_{s'} T(s, s') \cdot x[s]$ if $s \in S_?$

$\rightarrow$ solve linear equation system with $|S|$ variables

DTMC Reachability



DTMC $\langle S, T, s_I \rangle$ with probability matrix $\boldsymbol{P}$

– want to compute $\mathrm{P}(\diamond\, G)$

Let $x[s] \overset{\text{def}}{=} \mathrm{P}(\diamond\, G)$ from s and $S_? \overset{\text{def}}{=} S \setminus (S_0 \cup G)$.

Then, if $s_I \in S_?$:

$$x[s] = \textstyle\sum_{s' \in S_?} T(s, s') \cdot x[s] + \sum_{s_g \in G} T(s, s_g)$$

$$\boldsymbol{P} = \begin{pmatrix} 0 & 0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

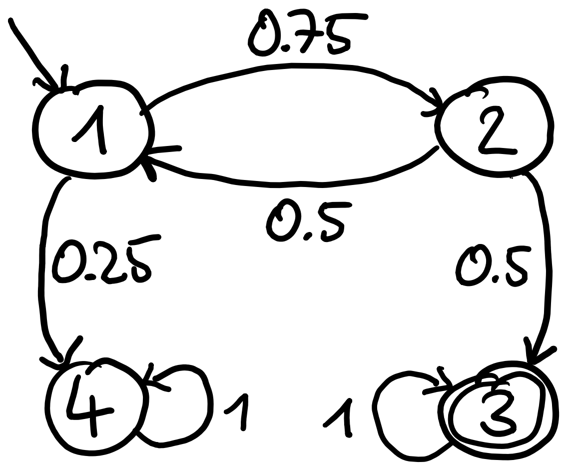
→ solve linear equation system $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$

where $A = \big(\boldsymbol{P}(s, s')\big)_{s, s' \in S_?}$ and $\boldsymbol{b}$ are the one-step pr. to G

DTMC Reachability

DTMC $\langle S, T, s_I \rangle$ with probability matrix $\boldsymbol{P}$

– want to compute $\mathrm{P}(\diamond\, G)$



Solve linear equation system $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$

$$\boldsymbol{P} = \begin{pmatrix} 0 & 0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1. Direct methods:

   Gaussian elimination,
   L/U decomposition, …

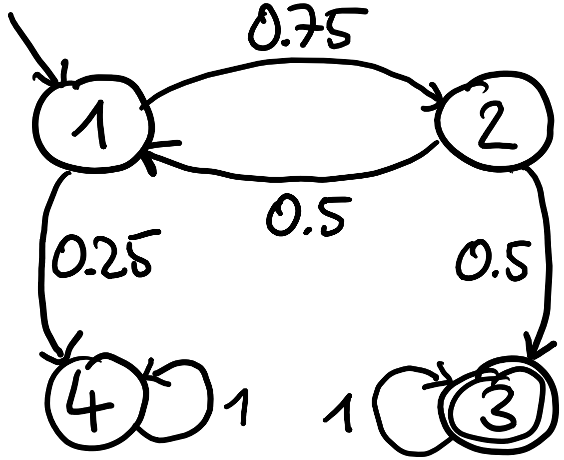
2. Iterative methods:

   Power, Jacobi, Gauss-Seidel, …

→ iterative methods preferred due to scalability – *but: inexact results*

Exercise I

Compute P(◊ { 3 }) for this DTMC by setting up and solving the linear equation system.

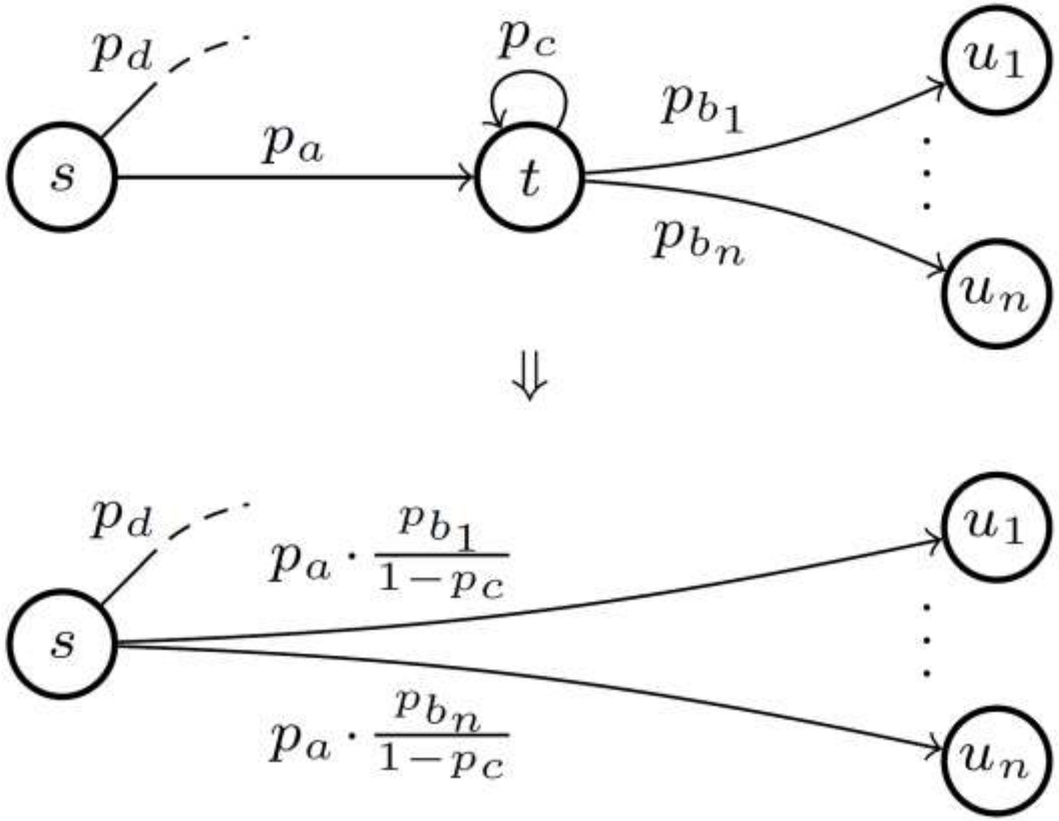


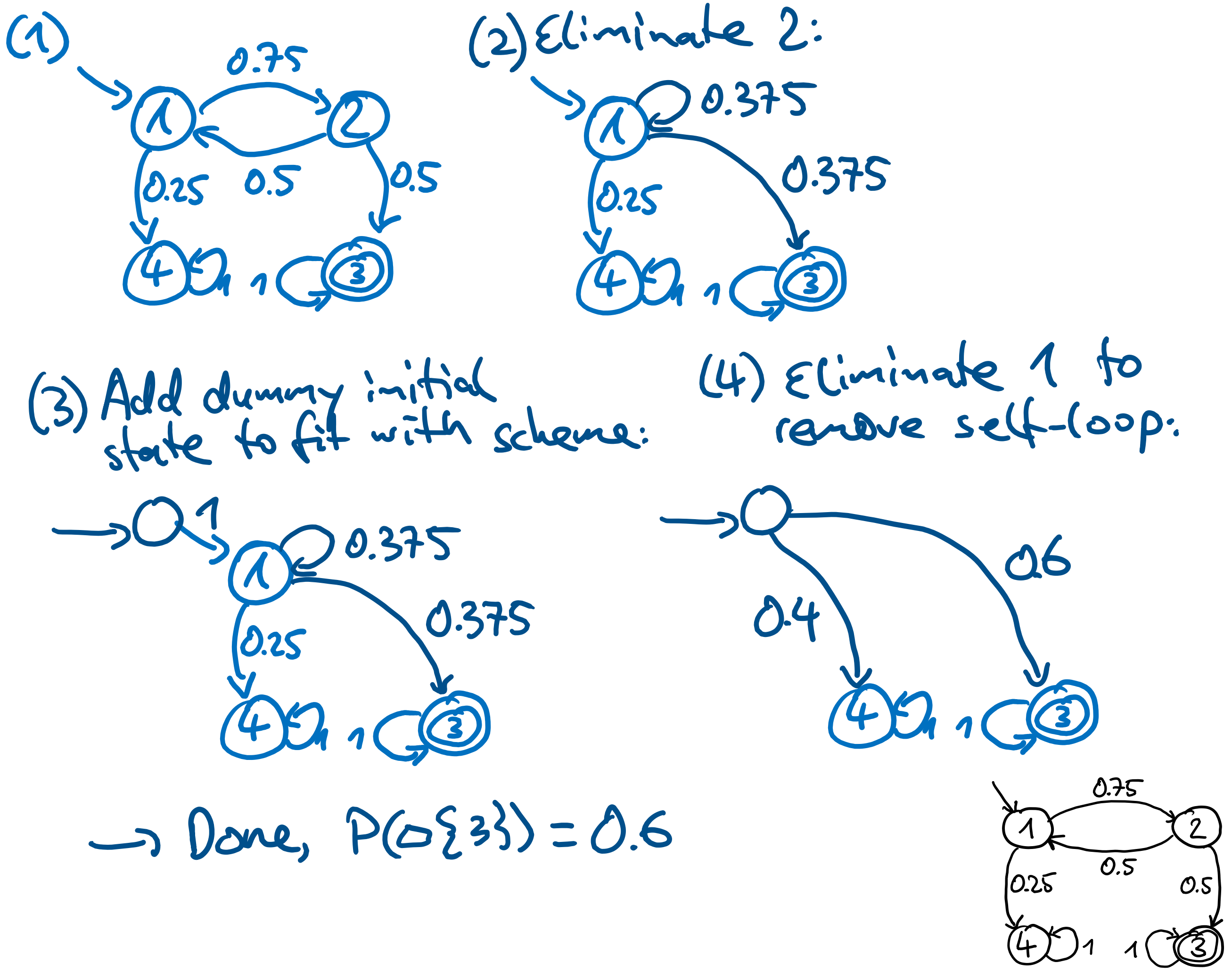
$$P = \begin{pmatrix} 0 & 0.75 & 0 & 0.25 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

State Elimination

Graphical variant of solving the linear equation system:

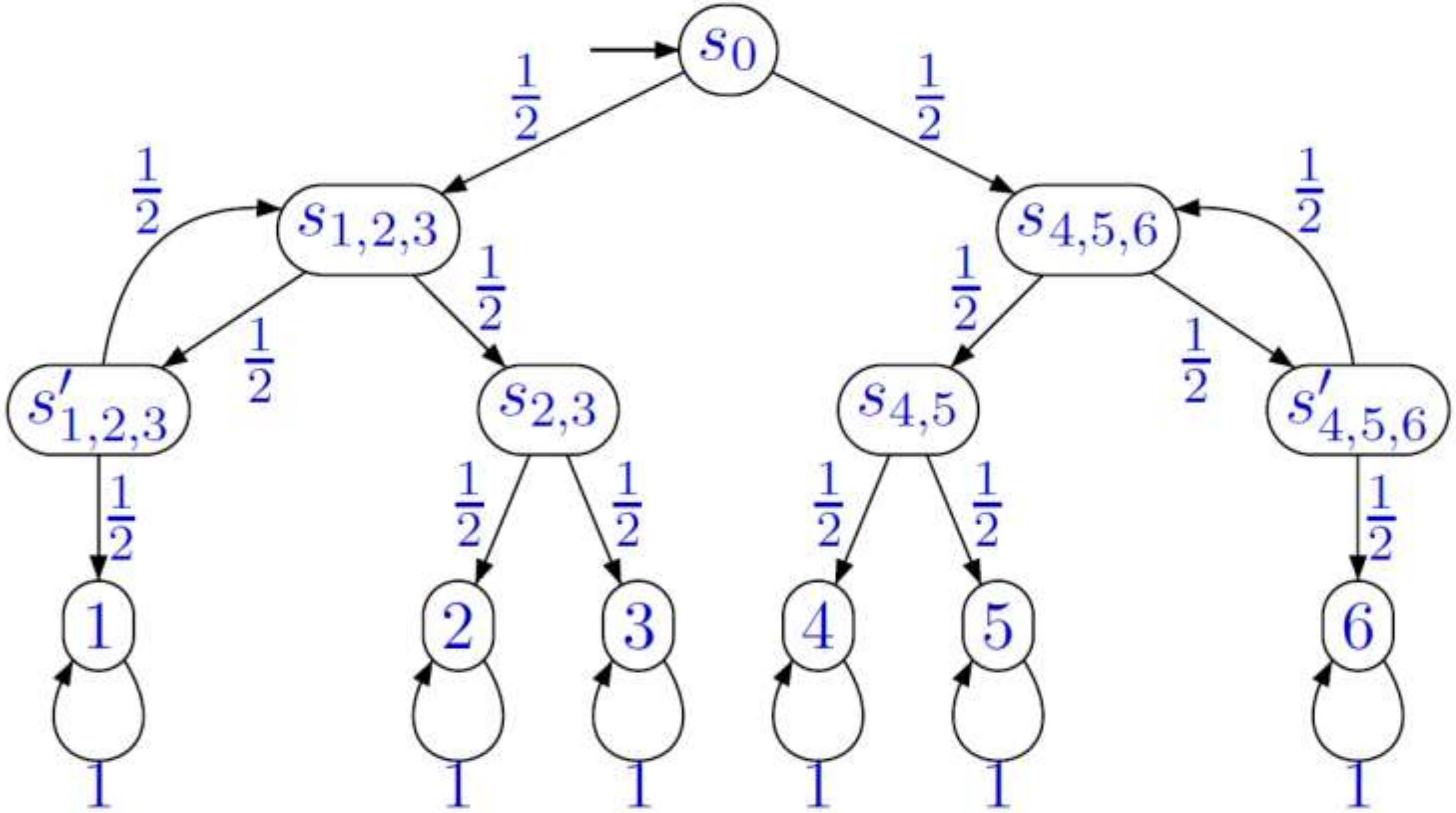
eliminate states one-by-one → the *state elimination* method





Exercise IIa

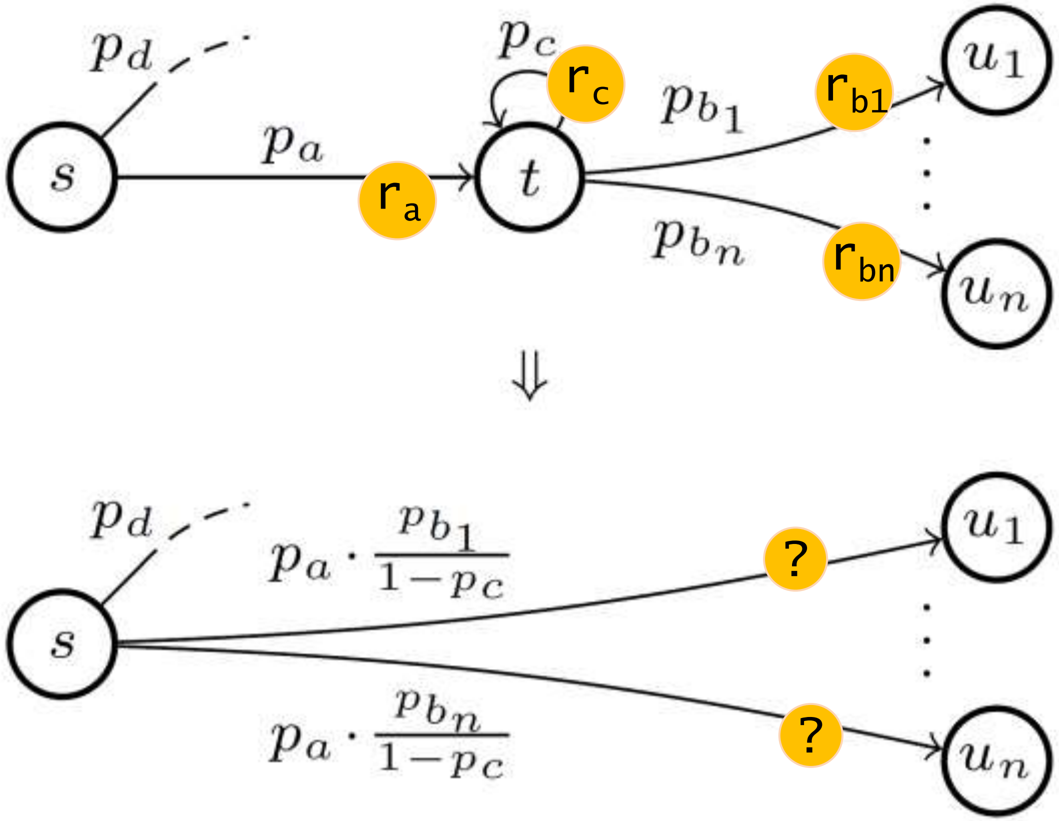
Use state elimination to check the correctness of the Knuth-Yao die:



Exercise IIb (BONUS)

Extend the state elimination scheme with rewards:

maintain the *expected* acc. reward of passing through state t



(hint: it's all about geometric series)

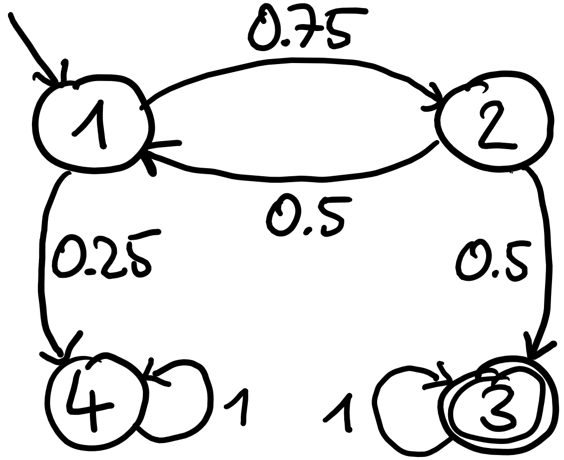
Value Iteration

Our $\mathrm{P}(\Diamond^{\leq b}\, G)$ algorithm –

$$x_0[s] = 1 \text{ if } s \in G \text{ else } 0$$

$$x_i[s] = \sum_{s'} T(s,s') \cdot x_{i-1}[s]$$

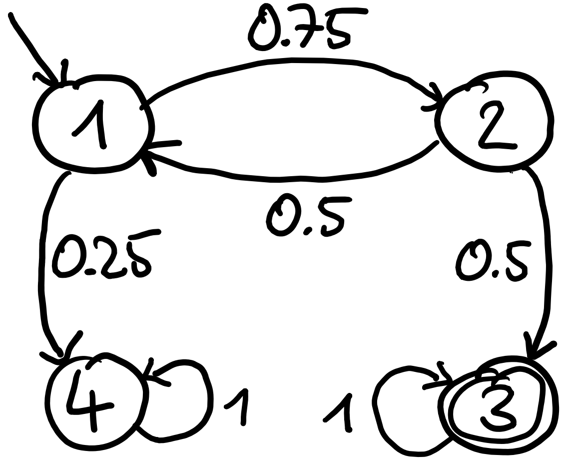
– is also known as **value iteration**:



i =	0	1	2	3	4
$x_i[1]$	0	0	0.375	0.375	
$x_i[2]$	0	0.5	0.5	0.6875	…
$x_i[3]$	1	1	1	1	
$x_i[4]$	0	0	0	0	

Value Iteration

$x_i[s] = \sum_{s'} T(s,s') \cdot x_{i-1}[s]$



We have

$$\lim_{i\to\infty} x_i[s] = x[s]$$

→ iterative method to approximate $\mathrm{P}(\diamond\, G)$

Q: When to stop?

A: Stop when $\forall s\colon |x_i[s] - x_{i-1}[s]| \leq \epsilon$ (absolute error)

or when $\forall s\colon \frac{|x_i[s] - x_{i-1}[s]|}{x_i[s]} \leq \epsilon$ (relative error)

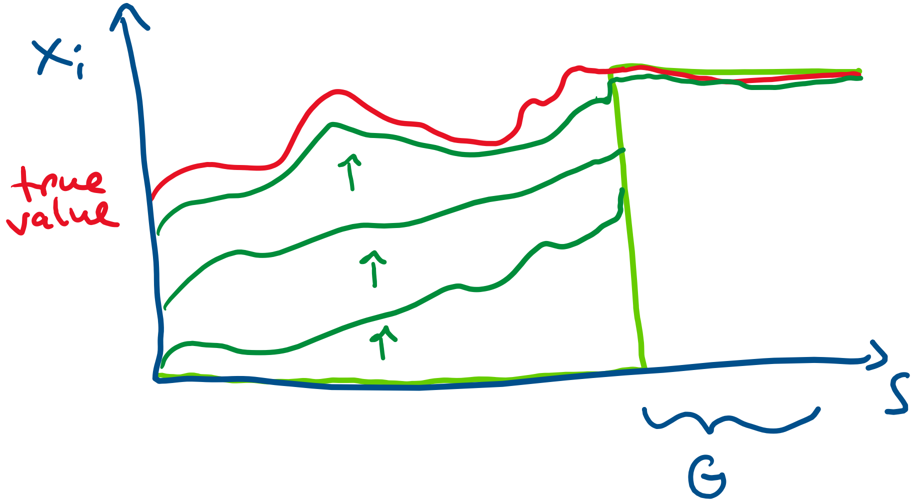
Problem: absolute/relative error of $\leq \epsilon$ at step i does not guarantee $|x[s] - x_i[s]| \leq \epsilon$ (or relative variant)

→ value iteration for $\mathrm{P}(\diamond\, G)$ is **unsound**!

Interval iteration

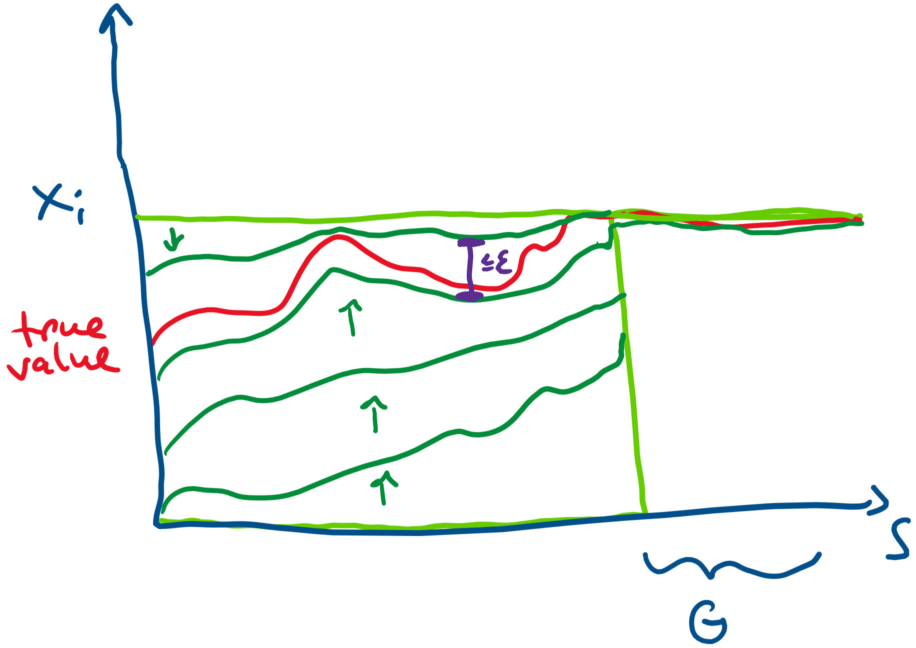
Value iteration:

approximate from below



Interval iteration:

approximate from below *and above*



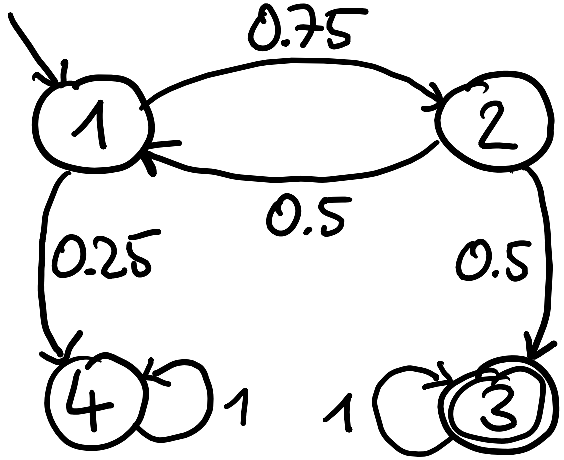
Value Iteration

Adaptation to $\mathrm{R}(\Diamond\, G)$ is easy:

1. Precompute S_1 states where $\mathrm{P}(\Diamond\, G) = 1$
2. If $s_I \notin S_1$: return ∞; otherwise, iterate:

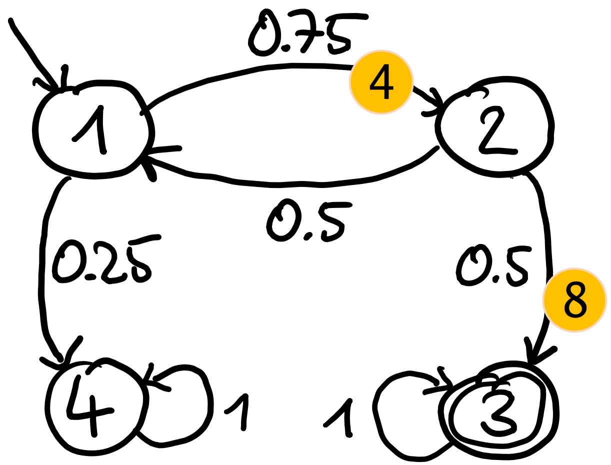
$$x_0[s] = 0 \text{ if } s \in S_1 \text{ else } \infty$$

$$x_i[s] = 0 \text{ if } s \in G \text{ else } \sum_{s'} T(s,s') \cdot (R(s,s') + x_{i-1}[s])$$

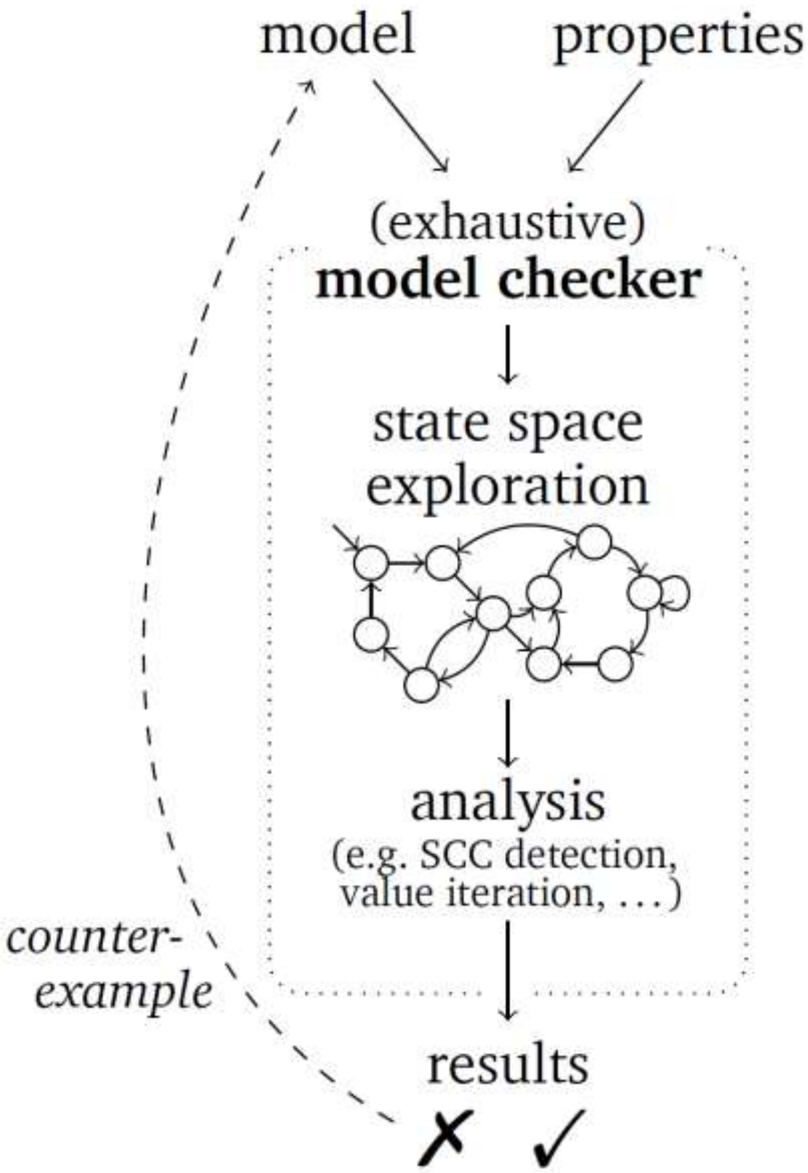


Exercise III

Run value iteration for $\mathrm{R}(\diamond\{3,4\})$ on this DTMC with an absolute-error $\epsilon = 1.2$ stopping criterion



$i =$	0	1
$x_i[1]$		
$x_i[2]$		
$x_i[3]$		
$x_i[4]$		

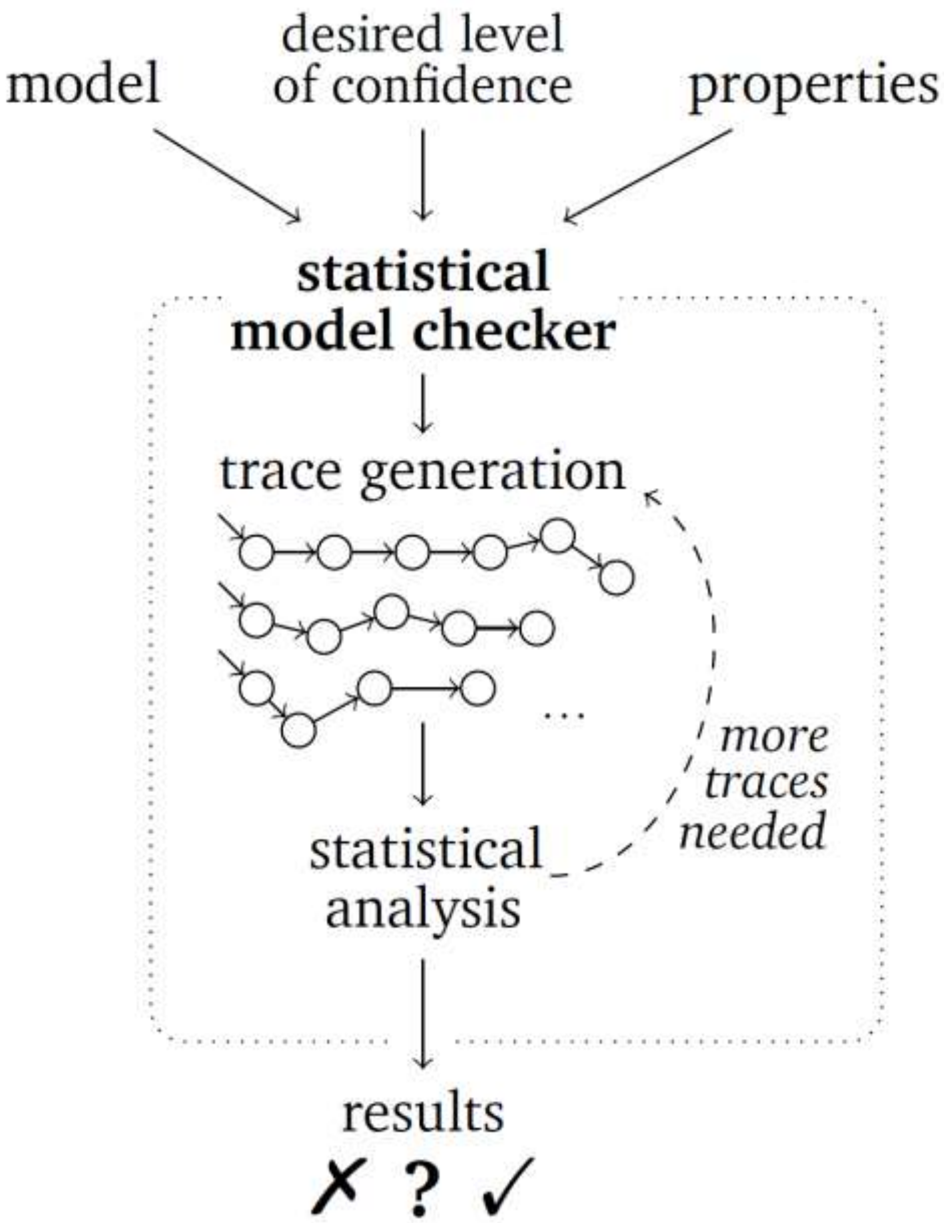


Probabilistic model checking (PMC)

requires an in-memory representation of the entire DTMC – states and probabilistic transition relation

to solve the linear equation system, perform value iteration, ...

Statistical Model Checking



Statistical model checking (SMC)

Monte Carlo simulation

Two steps:

1. Random path generation:

   outcome in $\{0,1\}$ of every path $=$ binomial random variable X_i

   sample mean $\bar{X} = \sum_{i=1}^{k} X_i / k$ is unbiased estimator for $\mathrm{P}(\diamond G)$

2. Statistical evaluation of outcomes

   to ensure correct results up to error $\pm\,\epsilon$ with confidence $1 - \delta$

SMC: Statistical Evaluation

For $\mathrm{P}(\diamond G)$: can use the Okamoto bound, which states that

$$\mathbb{P}(|\overline{X} - p| > \epsilon) < \delta$$

sample mean (estimate) — actual (unknown) value — error — $1-\delta$ "confidence"

$$\text{if } k \geq \frac{\ln\left(\frac{2}{\delta}\right)}{2 \cdot \epsilon^2}$$

Given two out of k, ϵ, δ, can compute the missing value a priori

e.g. 95% confidence ($\delta = 0.05$) with $\epsilon = 0.01$ needs $k = 18445$ runs

SMC: Sequential Testing

For $\mathrm{P}(\diamond G) \sim x$: use e.g. Wald's sequential probability ratio test

SPRT

indifference region

Input: DTMC $M = \langle S, T, s_I \rangle$, $G \subseteq S$, $x \in [0,1]$, ~~$d \in \mathbb{N}$~~, $\epsilon, \alpha, \beta \in (0,1)$
Output: requirement satisfaction: *true*, *false*, or *unknown*

type I/II errors

$p_0 := \min\{x + \epsilon, 1\}$, $p_1 := \max\{x - \epsilon, 0\}$
$a := \log((1-\beta)/\alpha)$, $b := \log(\beta/(1-\alpha))$
$r := 0$
repeat
    $v :=$ `simulate`(M, G, d)
    if $v = \textit{unknown}$ **then return** *unknown*
    else if $v = \textit{true}$ **then** $r := r + \log p_1 - \log p_0$
    else $r := r + \log(1 - p_1) - \log(1 - p_0)$
    if $r \leq b$ **then return** $\sim = \geq$ // *likely* $\mathrm{P}(\diamond G) \geq p_0 \geq x$
    else if $r \geq a$ **then return** $\sim = \leq$ // *likely* $\mathrm{P}(\diamond G) \leq p_1 \leq x$

SMC for Rewards

Expected rewards:

random variables X_i are no longer binomial, but follow some unknown probability distribution

~~Okamoto~~ ~~SPRT~~

→ confidence intervals still possible

SMC vs. PMC

Statistical model checking

= random path generation
plus statistical evaluation

Memory usage:

$O(1)$ (in size of DTMC)

Runtime:

slow if high precision
required (e.g. rare events)
or bound in SPRT close
to ε but not inside indifference

Probabilistic model checking

= full state space exploration
plus numeric computation

Memory usage:

$O(\text{size of DTMC})$

Runtime:

depends on model
size & structure
(e.g. probabilities
on loops)

Exercise IV

1. Let us use the Okamoto bound for SMC.
   a) With $k = 10000$ simulation runs and desired confidence level $1 - \delta = 0.9$, what is the error ϵ that we can guarantee?
   b) Let δ and ϵ be fixed. How does k change if we halve the error, i.e. we use $\epsilon' = \epsilon/2$ instead of the original ϵ?

2. Let us study a single coin flip with outcomes *heads* and *tails*.

   Use a real coin to estimate $\mathrm{P}(\diamond \{heads\})$ via many "simulation runs" (= coin flips). Use the SPRT to perform only as many flips as necessary to determine whether $\mathrm{P}(\diamond \{heads\}) > \frac{1}{3}$ with indifference $\epsilon = 0.05$ and $\alpha = \beta = 0.1$.

   Document the steps of the SPRT algorithm as you run it.

DTMC Algorithms in Modest

Probabilistic model checking: *mcsta*

Statistical model checking: *modes*

(Demo)