

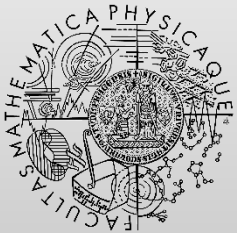
Behavior models and verification

Lecture 13

<http://d3s.mff.cuni.cz>



Jan Kofroň, František Plášil



FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Motivation

- In some cases **absolute** absence of errors is infeasible
 - failures of particular parts of system
 - non-deterministic behavior of users
 - ...
- It might be useful to determine level of reliability in terms of probability
 - frequency of errors
 - time to recovery
 - throughput
 - mean waiting time
 - ...

The approach is:

Probabilistic (stochastic) model checking

Lecture based on M. Kwiatkowska et al.: Stochastic Model Checking
[<http://www.prismmodelchecker.org/papers/sfm07.pdf>]

Stochastic model checking

- Not only validity of certain properties
 - but also probability of reaching states/paths
- → Need for special language
 - PCTL = Probabilistic Computational Tree Logic
 - CSL = Continuous Stochastic Logic
- **Discrete-time Markov Chains (DTMC)** are used as models for discrete time analysis
- **Continuous-time Markov Chains (CTMC)** are used for continuous time analysis

Discrete-time Markov chains



Discrete-time Markov chains

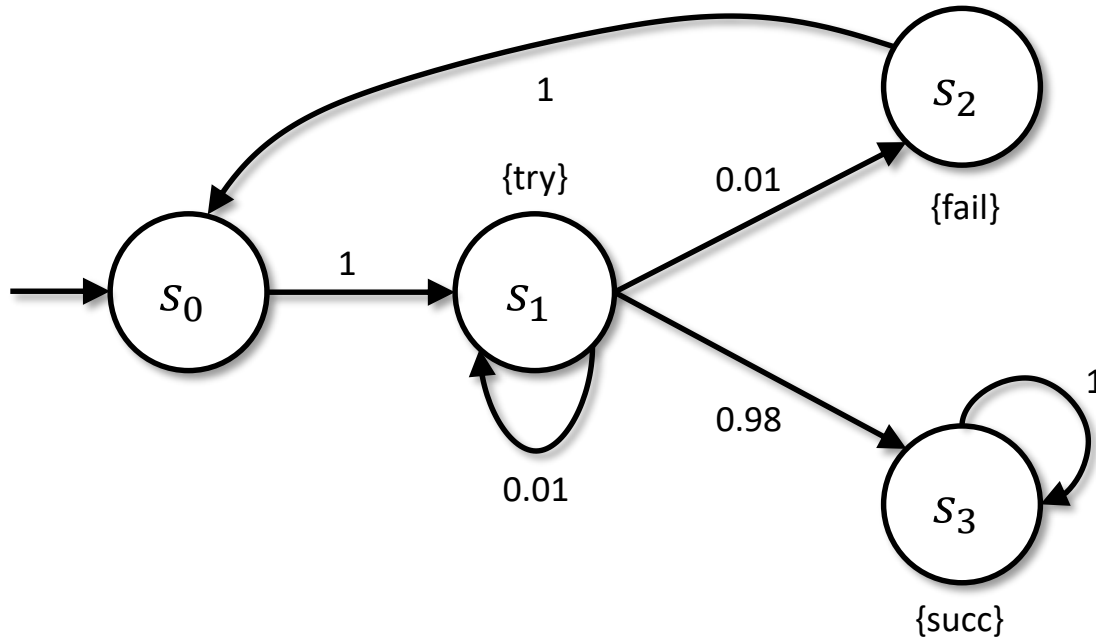
Definition: A labeled DTMC D is a tuple $(S, \bar{s}, \mathbf{P}, L)$ where:

- S is finite set of states
- $\bar{s} \in S$ is initial state
- $\mathbf{P}: S \times S \rightarrow [0,1]$ is **transition probability matrix** where $\sum_{s' \in S} \mathbf{P}(s, s') = 1$ for all $s \in S$
- $L: S \rightarrow 2^{AP}$ is labeling function assigning to each state set $L(s)$ of atomic propositions

Discrete-time Markov chains

- Sum of probabilities of transitions originating in each state must be 1!
- Terminating states can be modeled by self-loop with probability 1

Example



$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Notions

- **Path** is non-empty sequence $s_0s_1s_2 \dots$ where $s_i \in S$ and $\forall i \geq 0: P(s_i, s_{i+1}) > 0$
- Path can be finite or infinite
- $Path^D(s)$ – set of **infinite** paths in D starting at s
 - this is default meaning of paths
- $Path_{fin}^D(s)$ – set of **finite** paths in D starting at s

Path probability

Probability for finite path $\omega_{fin} \in P_{fin}^D(s)$:

$$P_s(\omega_{fin}) = \begin{cases} 1 & \text{if } n = 0 \\ \prod_{i=0}^{n-1} P(\omega(i), \omega(i+1)) & \text{otherwise} \end{cases}$$

where n is length of ω_{fin}

Cylinder set $C(\omega_{fin}) \subseteq Path^D(s)$:

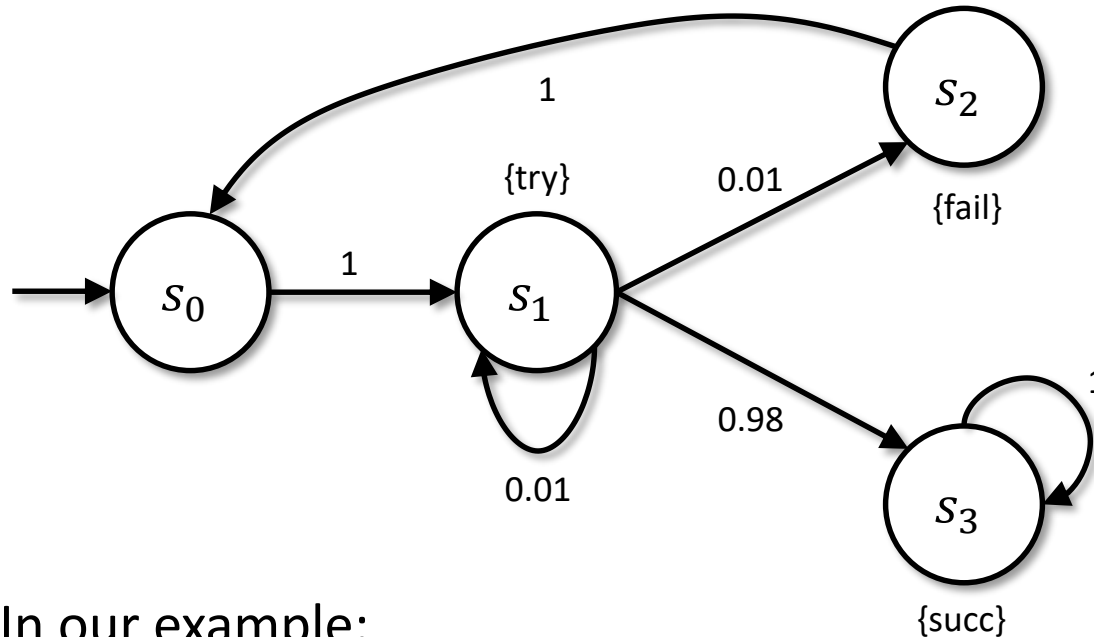
$$C(\omega_{fin}) \stackrel{\text{def}}{=} \{\omega \in Path^D(s) \mid \omega_{fin} \text{ is a prefix of } \omega\}$$

Probability measure

Probability measure Pr_s is function defined as:

$$Pr_s \left(C(\omega_{fin}) \right) = P_s(\omega_{fin}) \text{ for all } \omega_{fin} \in Path_{fin}^D(s)$$

Probability measure – Example



In our example:

$$Pr_{s_0}(C(s_0s_1s_1s_1)) = 1.00 \cdot 0.01 \cdot 0.01 = 0.0001$$

$$Pr_{s_0}(C(s_0s_1s_1s_2)) = 1.00 \cdot 0.01 \cdot 0.01 = 0.0001$$

$$Pr_{s_0}(C(s_0s_1s_1s_3)) = 1.00 \cdot 0.01 \cdot 0.98 = 0.0098$$

$$Pr_{s_0}(C(s_0s_1s_2s_0)) = 1.00 \cdot 0.01 \cdot 1.00 = 0.01$$

$$Pr_{s_0}(C(s_0s_1s_3s_3)) = 1.00 \cdot 0.98 \cdot 1.00 = 0.98$$

Probabilistic Computational Tree Logic (PCTL)



Probabilistic Computational Tree Logic (PCTL)

- Extension of CTL
- Syntax:

$$\begin{aligned}\Phi &::= true \mid a \mid \neg\Phi \mid \Phi \wedge \Phi \mid P_{\sim p}[\phi] && \text{(state formula)} \\ \phi &::= X\Phi \mid \Phi U^{\leq k} \Phi, \text{ where} && \text{(path formula)}\end{aligned}$$

a is atomic proposition

$\sim \in \{<, \leq, \geq, >\}$

$p \in [0,1]$

$k \in \mathbb{N} \cup \infty$

- ... plus common (derived) facts:
 - $false \equiv \neg true$
 - $\Phi \vee \Psi \equiv \neg(\neg\Phi \wedge \neg\Psi)$

Semantics of PCTL

$s \models \text{true}$ for all $s \in S$

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

$s \models \neg\Phi \Leftrightarrow s \not\models \Phi$

$s \models \Phi \wedge \Psi \Leftrightarrow s \models \Phi \wedge s \models \Psi$

$s \models P_{\sim p}[\phi] \Leftrightarrow \text{Prob}^D(s, \phi) \sim p$

$\omega \models X\Phi \Leftrightarrow \omega(1) \models \Phi$

$\omega \models \phi U^{\leq k} \psi \Leftrightarrow \exists i \in \mathbb{N}: (i \leq k \wedge \omega(i) \models \psi \wedge \forall j < i: (\omega(j) \models \phi))$

where $\text{Prob}^D(s, \phi) \stackrel{\text{def}}{=} Pr_s\{\omega \in \text{Path}^D(s) \mid \omega \models \phi\}$

Common CTL operators

- CTL F and G operators:

$$P_{\sim p}[F \Phi] \equiv P_{\sim p}[true U^{\leq \infty} \Phi]$$

$$P_{\sim p}[F^{\leq k} \Phi] \equiv P_{\sim p}[true U^{\leq k} \Phi]$$

$$G \Phi \equiv \neg F \neg \Phi$$

$$G^{\leq k} \Phi \equiv \neg F^{\leq k} \neg \Phi$$

Negation

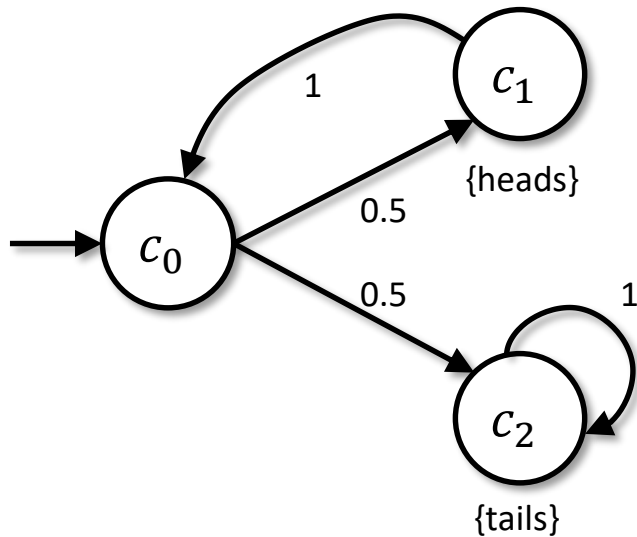
- Syntax does not allow for negation of path formulae
- However, it holds:

$$P_{\sim p}[G \Phi] \equiv P_{\bar{\sim}1-p}[F \neg\Phi]$$
$$P_{\sim p}[G^{\leq k} \Phi] \equiv P_{\bar{\sim}1-p}[F^{\leq k} \neg\Phi]$$

where $\bar{<} \equiv >$, $\bar{\leq} \equiv \geq$, $\bar{\geq} \equiv \leq$, $\bar{>} \equiv <$

Quantifiers

- $P_{\sim p}[\cdot]$ is probabilistic analogue to path quantifiers:
 - $EF\Phi \equiv P_{>0}[F \Phi]$
 - But: $AF\Phi$ is **NOT** the same as $P_{\geq 1}[F \Phi]$



c_0 satisfies $P_{\geq 1}[F \text{ tails}]$
 c_0 does **NOT** satisfy $AF \text{ tails}$

Examples of PCTL properties

- $P_{\geq 0.4}[X \textit{ delivered}]$
 - probability that message gets delivered in next step is at least 0.4
- $init \rightarrow P_{\leq 0}[F \textit{ error}]$
 - error state is not reachable from any init state
- $P_{\geq 0.9}[\neg \textit{down } U \textit{ served}]$
 - probability that server does not go down before request gets served is at least 0.9
- $P_{< 0.1}[\neg \textit{done } U^{\leq 10} \textit{ fault}]$
 - probability that error occurs before protocol is done and within 10 steps is less than 0.1

Model checking PCTL

- Based on CTL model checking algorithm
 1. decomposing formula into sub-formulae
 2. in bottom-up manner finding set of states satisfying particular sub-formulae
 3. the set of states for the input formula at root
- Special handling of the P formulae

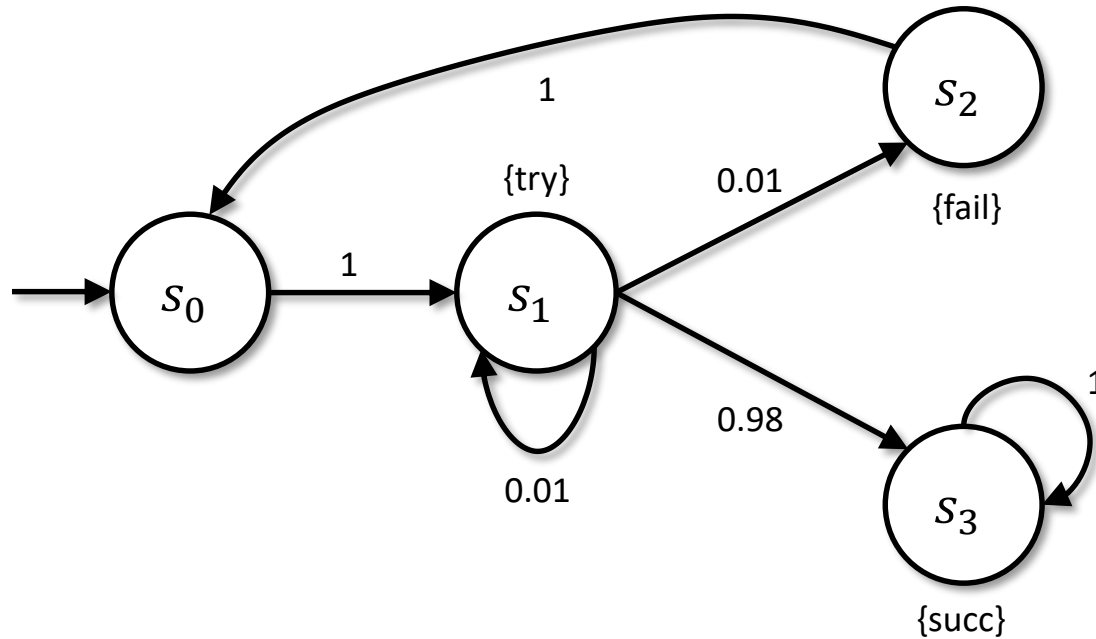
- For $P_{\sim p}[X\Phi]$ we need to compute $Prob^D(s, X\Phi)$ for each state s :

$$Prob^D(s, X\Phi) = \sum_{s' \in Sat(\Phi)} \mathbf{P}(s, s')$$

where $Sat(\Phi)$ is set of states satisfying Φ

- Let $\underline{\Phi}(s) = \begin{cases} 1 & \text{if } s \in Sat(\Phi) \\ 0 & \text{otherwise} \end{cases}$
- $\underline{Prob^D}(X\Phi) = \mathbf{P} \cdot \underline{\Phi}$
 - Vector with probabilities for particular states

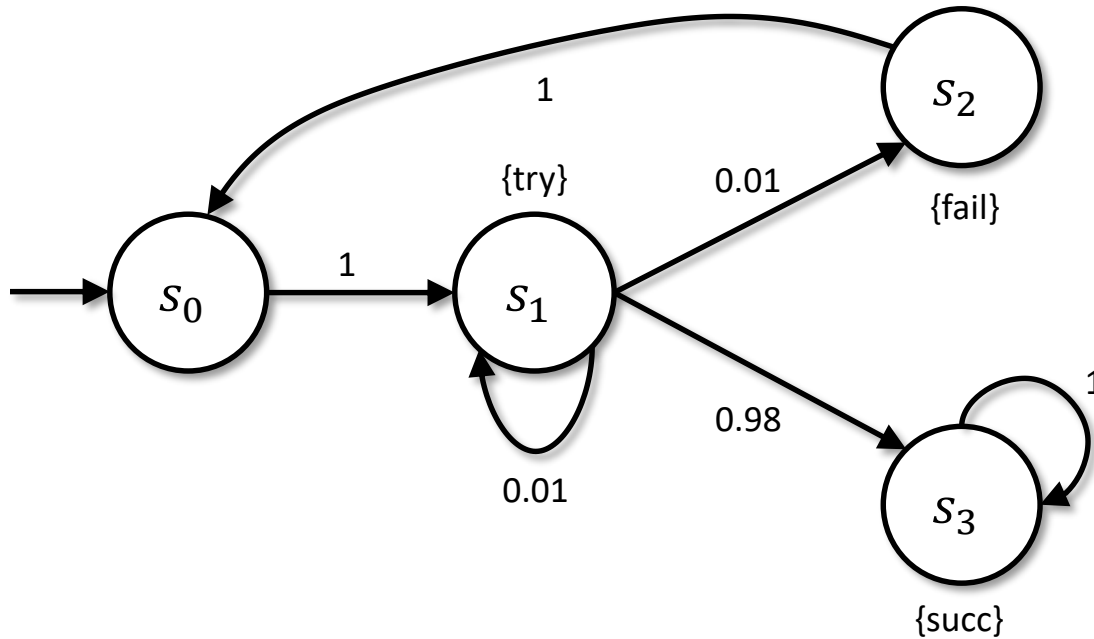
XΦ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P_{\geq 0.9}[X(\neg try \vee succ)]$$

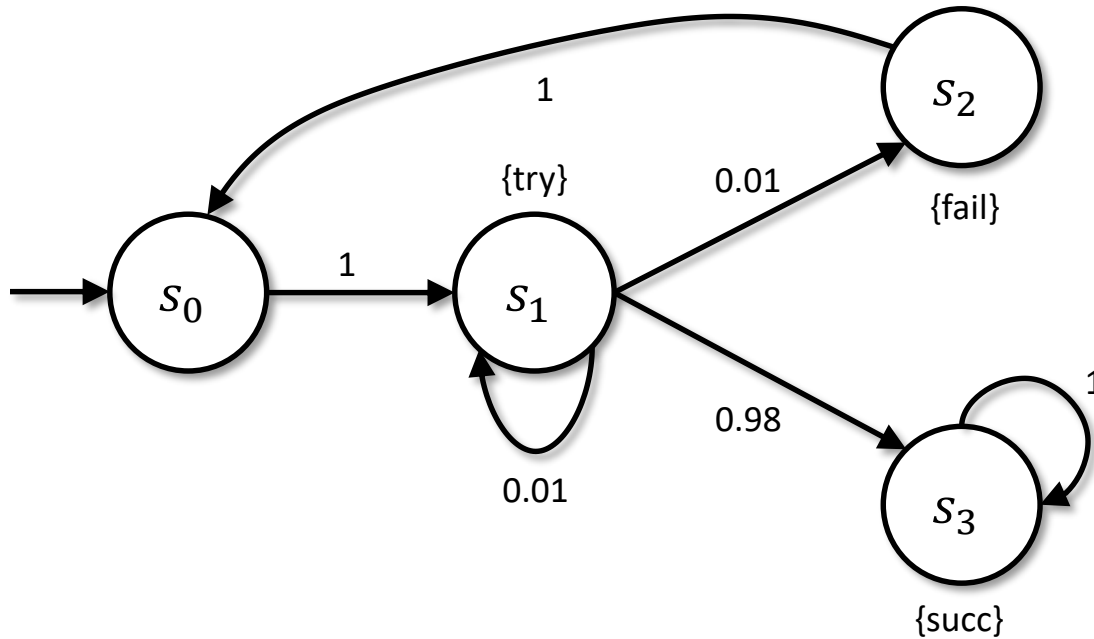
XΦ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Sat(\neg try \vee succ) = \{S_0, S_2, S_3\} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

XΦ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.99 \\ 1 \\ 1 \end{pmatrix}$$

$\Phi U^{\leq k} \Psi$ – for $k \neq \infty$

- For $P_{\sim p}[\Phi U^{\leq k} \Psi]$ we need to compute $Prob^D(s, \Phi U^{\leq k} \Psi)$ for each state s :

$$\begin{aligned} Prob^D(s, \Phi U^{\leq k} \Psi) &= \\ &= \begin{cases} 1 & \text{if } s \in Sat(\Psi) \\ 0 & \text{if } k = 0 \text{ or } s \in Sat(\neg\Phi \wedge \neg\Psi) \\ \sum_{s' \in S} \mathbf{P}(s, s') \cdot Prob^D(s', \Phi U^{\leq k-1} \Psi) & \text{otherwise} \end{cases} \end{aligned}$$

where $Sat(\Phi)$ is set of states satisfying Φ

$\Phi U^{\leq k} \Psi$ – for $k \neq \infty$

Definition: For any *DTMC* $D = (S, \bar{s}, \mathbf{P}, L)$ and PCTL formula Φ , let $D[\Phi] = (S, \bar{s}, \mathbf{P}[\Phi], L)$ where, if $s \not\models \Phi$, then $\mathbf{P}[\Phi](s, s') = \mathbf{P}(s, s')$ for all $s' \in S$, and if $s \models \Phi$, then $\mathbf{P}[\Phi](s, s) = 1$ and $\mathbf{P}[\Phi](s, s') = 0$ for all $s' \neq s$.

Then it holds:

$$Prob^D(s, \Phi U^{\leq k} \Psi) = \sum_{s' \models \Psi} \pi_{s,k}^{D[\neg\Phi \vee \Psi]}(s')$$

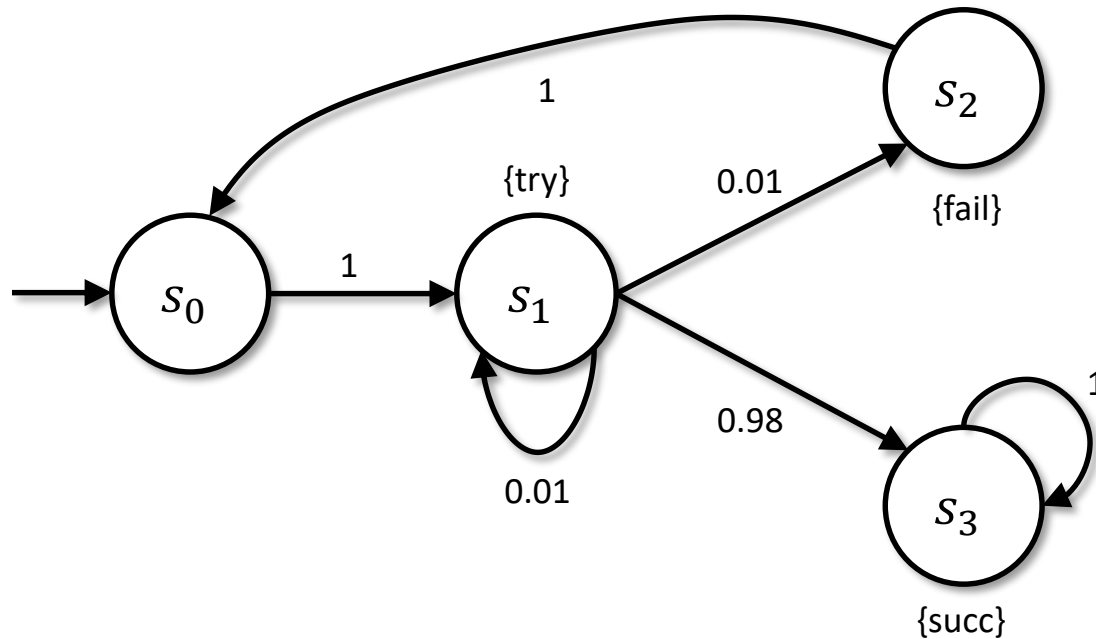
$\Phi \ U^{\leq k} \ \Psi$ – for $k \neq \infty$

- Vector of probabilities $\underline{Prob}^D(\Phi \ U^{\leq k} \ \Psi)$ can be computed as:

$$\underline{Prob}^D(\Phi \ U^{\leq k} \ \Psi) = (\mathbf{P}[\neg\Phi \vee \Psi])^k \cdot \underline{\Psi}$$

- Usually computed in iterative way
 - but can be pre-computed for particular k

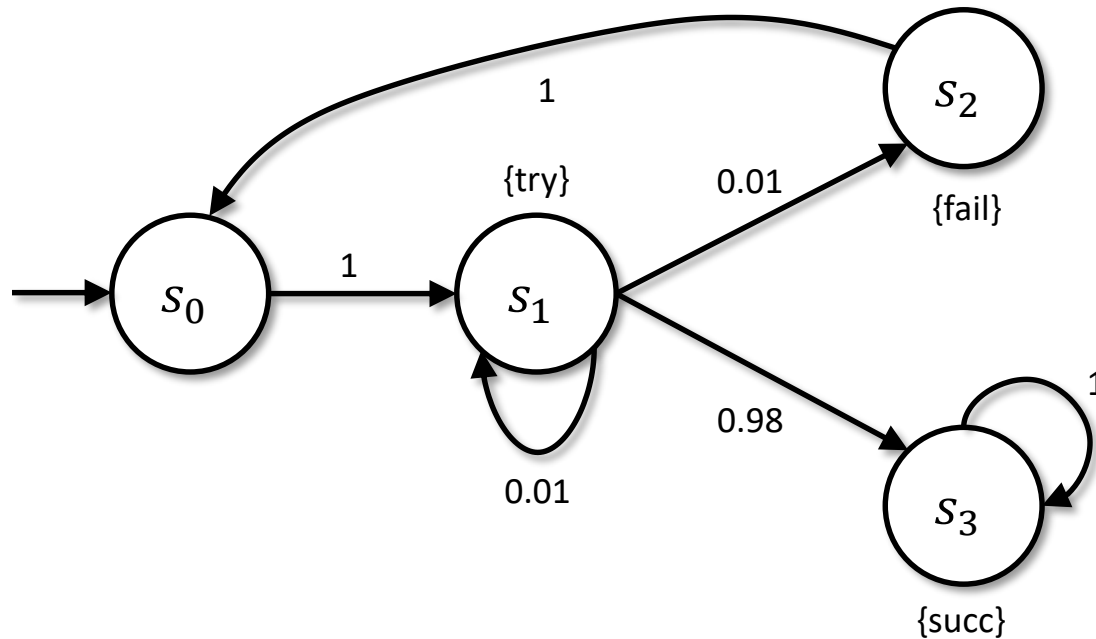
$\Phi \ U^{\leq k} \ \Psi$ – for $k \neq \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P_{>0.98}[F^{\leq 2} succ] = P_{>0.98}[true \ U^{\leq 2} succ]$$

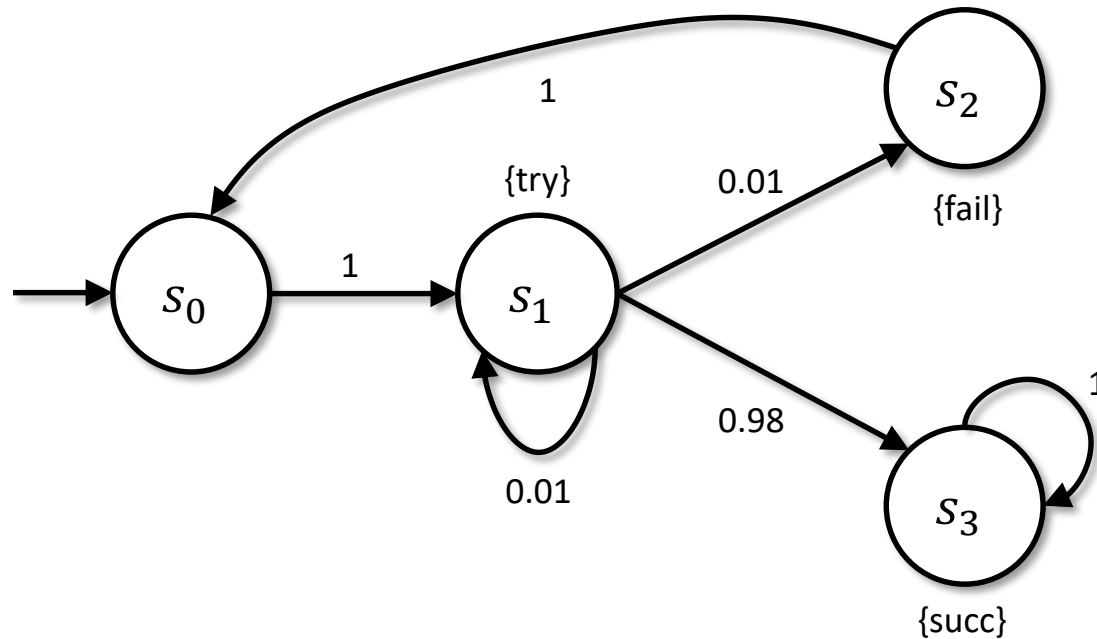
$\Phi \cup^{\leq k} \Psi$ – for $k \neq \infty$ – Example



$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Sat}(true) = \{s_0, s_1, s_2, s_3\}, \quad \text{Sat}(succ) = \{s_3\}$$
$$\mathbf{P}[\neg true \vee succ] = \mathbf{P}$$

$\Phi U^{\leq k} \Psi$ – for $k \neq \infty$ – Example



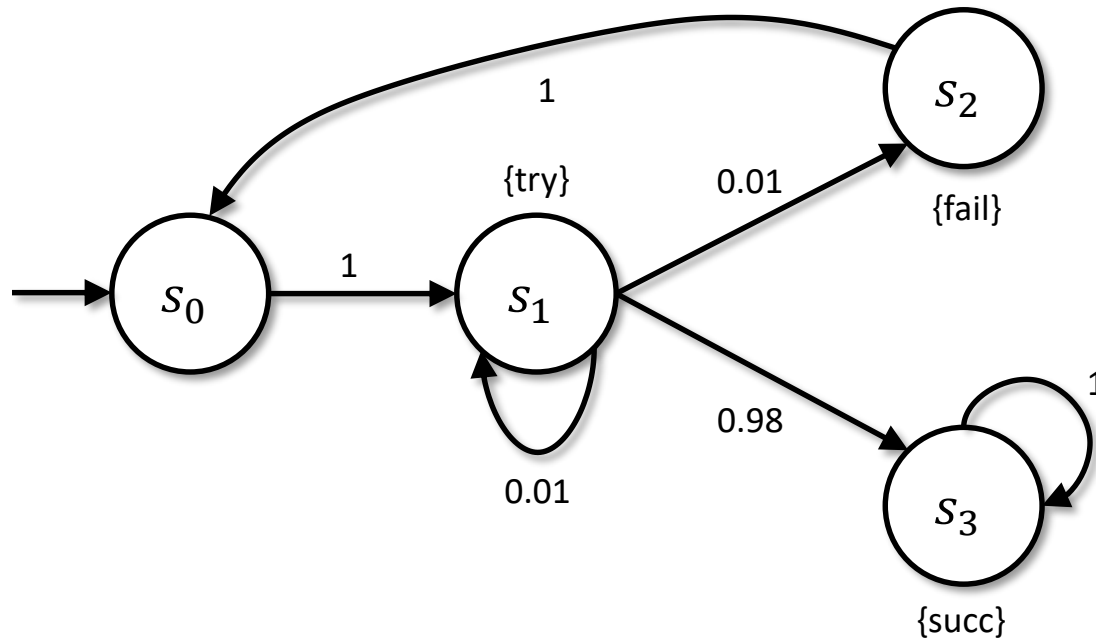
$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\underline{Prob}^D(\Phi U^{\leq 0} \Psi) = succ = [0, 0, 0, 1]$$

$$\underline{Prob}^D(\Phi U^{\leq 1} \Psi) = \mathbf{P}[\neg true \vee succ] \cdot \underline{Prob}^D(\Phi U^{\leq 0} \Psi) = [0, 0.98, 0, 1]$$

$$\underline{Prob}^D(\Phi U^{\leq 2} \Psi) = \mathbf{P}[\neg true \vee succ] \cdot \underline{Prob}^D(\Phi U^{\leq 1} \Psi) = [0.98, 0.9898, 0, 1]$$

$\Phi U^{\leq k} \Psi$ – for $k \neq \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\underline{Prob}^D(\Phi U^{\leq 2} \Psi) = [0.98, 0.9898, 0, 1]$$

$$\text{Hence } Sat(P_{>0.98}[F^{\leq 2} succ]) = \{s_1, s_3\}$$

$\Phi U^{\leq k} \Psi$ – for $k = \infty$

- For brevity, instead of $U^{\leq \infty}$ we just write U
- We need to compute $Prob^D(s, \Phi U \Psi)$ for each state s :

$$\begin{aligned} Prob^D(s, \Phi U \Psi) &= \\ &= \begin{cases} 1 & \text{if } s \in Sat(\Psi) \\ 0 & \text{if } s \in Sat(\neg\Phi \wedge \neg\Psi) \\ \sum_{s' \in S} \mathbf{P}(s, s') \cdot Prob^D(s', \Phi U \Psi) & \text{otherwise} \end{cases} \end{aligned}$$

$\Phi U^{\leq k} \Psi$ – for $k = \infty$

- This system of equations can have many solutions – we convert it to one with just one solution
- The following sets are computed using fixpoint algorithm (similar to CTL case, using complement on sets):

$$\text{Sat}(P_{\leq 0}[\Phi U \Psi]) = \{s \in S \mid \text{Prob}^D(s, \Phi U \Psi) = 0\}$$

$$\text{Sat}(P_{\geq 1}[\Phi U \Psi]) = \{s \in S \mid \text{Prob}^D(s, \Phi U \Psi) = 1\}$$

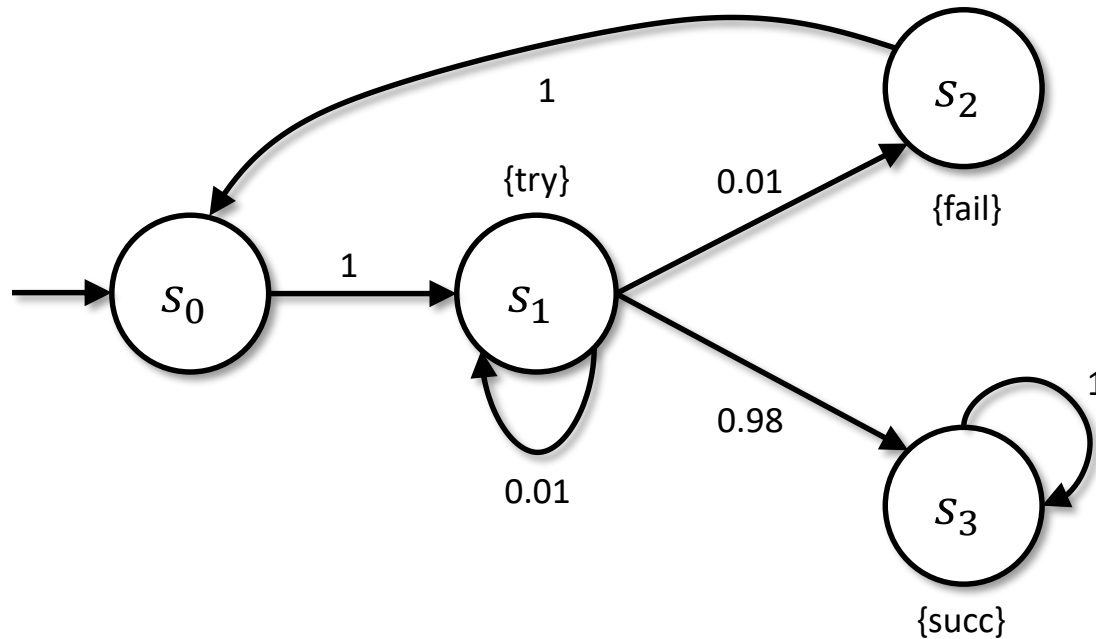
$\Phi U^{\leq k} \Psi$ – for $k = \infty$

- Resulting system of equation then reads:

$$\begin{aligned} Prob^D(s, \Phi U \Psi) &= \\ &= \begin{cases} 1 & \text{if } s \in Sat(P_{\geq 1}[\Phi U \Psi]) \\ 0 & \text{if } s \in Sat(P_{\leq 0}[\Phi U \Psi]) \\ \sum_{s' \in S} \mathbf{P}(s, s') \cdot Prob^D(s', \Phi U \Psi) & \text{otherwise} \end{cases} \end{aligned}$$

- Having computed sets for probabilities 0 and 1, we can restrict computation to rest of states
 - Optimization

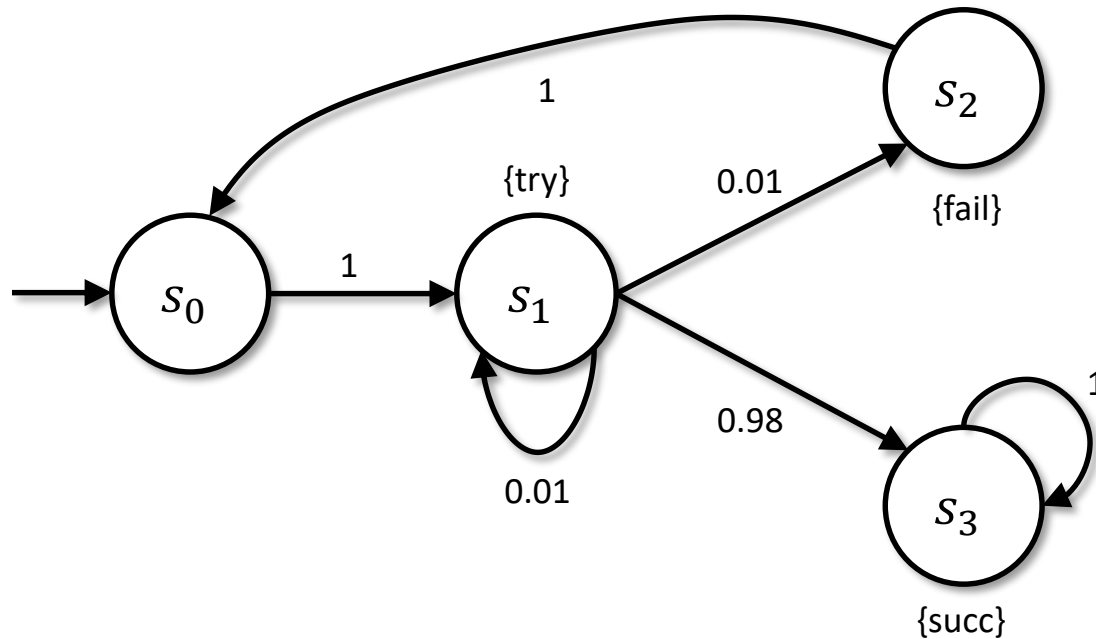
$\Phi \ U^{\leq k} \ \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P_{>0.99}[try \ U \ succ]$$

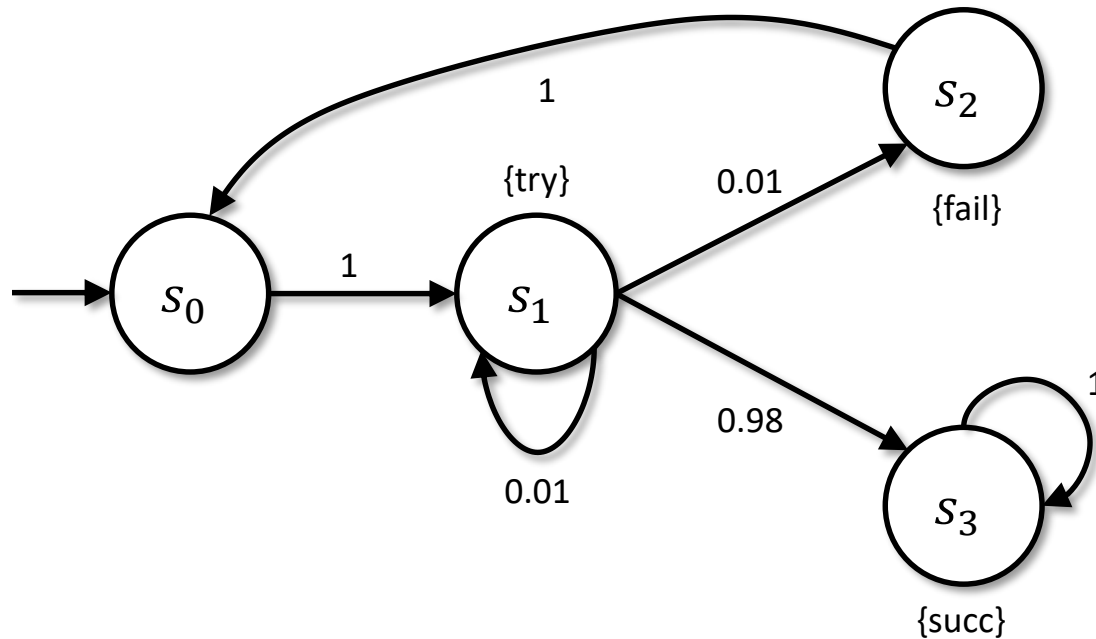
$\Phi \cup^{\leq k} \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$Sat(try) = \{s_1\}, \quad Sat(succ) = \{s_3\}$$

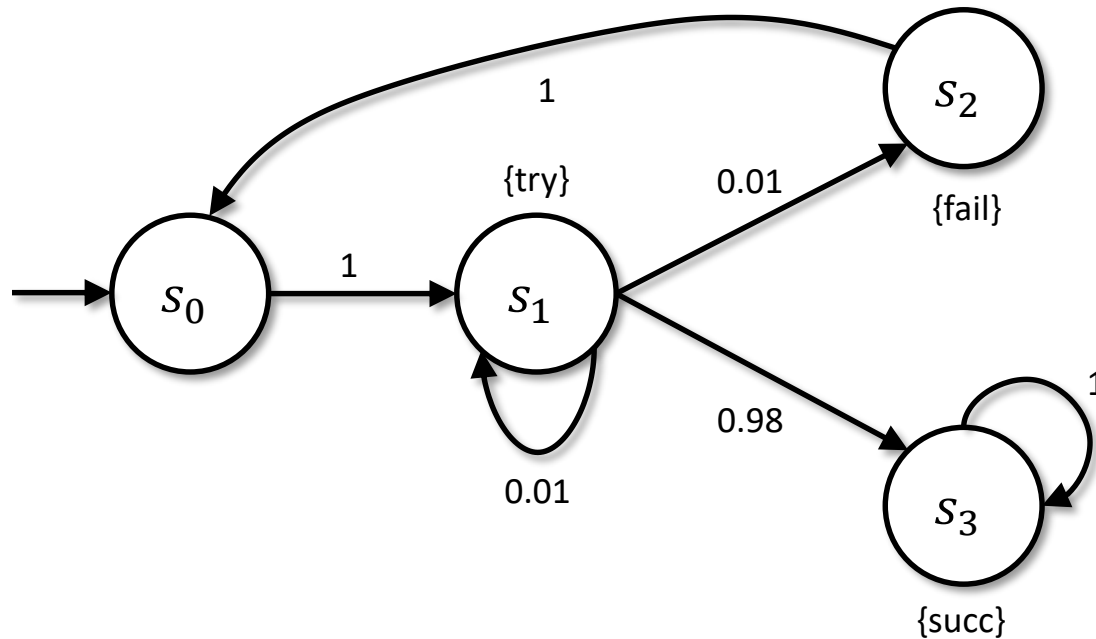
$\Phi U^{\leq k} \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Sat}(P_{\leq 0}[\text{try } U \text{ succ}]) = \{s_0, s_2\}, \quad \text{Sat}(P_{\geq 1}[\text{try } U \text{ succ}]) = \{s_3\}$$

$\Phi U^{\leq k} \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

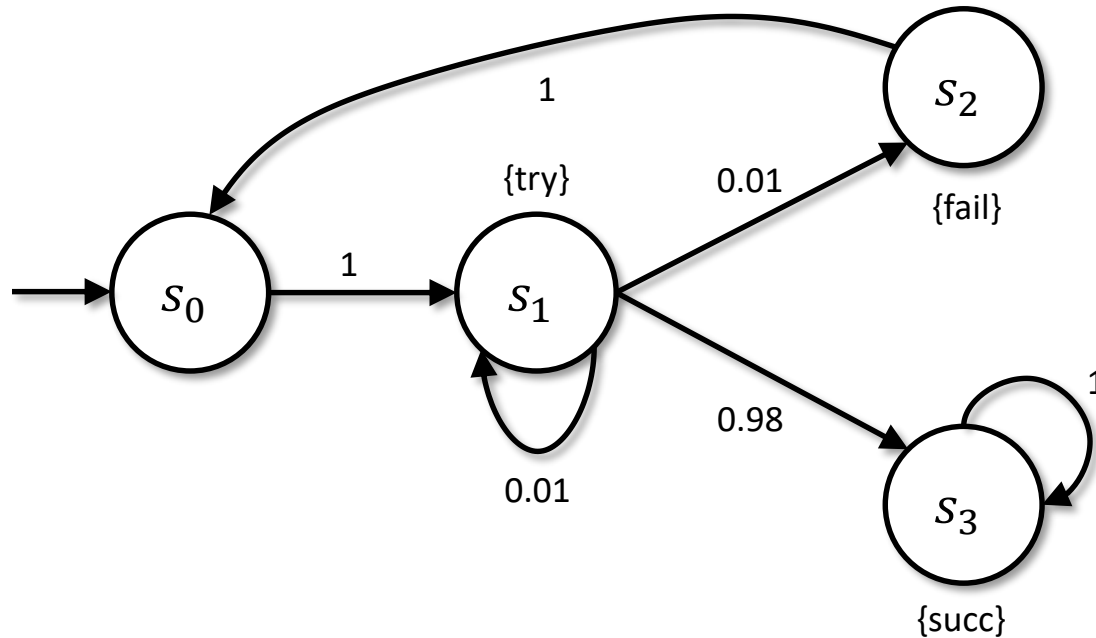
$$Prob^D(s_0, \text{try } U \text{ succ}) = 0$$

$$Prob^D(s_1, \text{try } U \text{ succ}) = 0.01 \cdot Prob^D(s_1, \text{try } U \text{ succ}) + 0.01 \cdot Prob^D(s_2, \text{try } U \text{ succ}) + 0.98 \cdot Prob^D(s_3, \text{try } U \text{ succ})$$

$$Prob^D(s_2, \text{try } U \text{ succ}) = 0$$

$$Prob^D(s_3, \text{try } U \text{ succ}) = 1$$

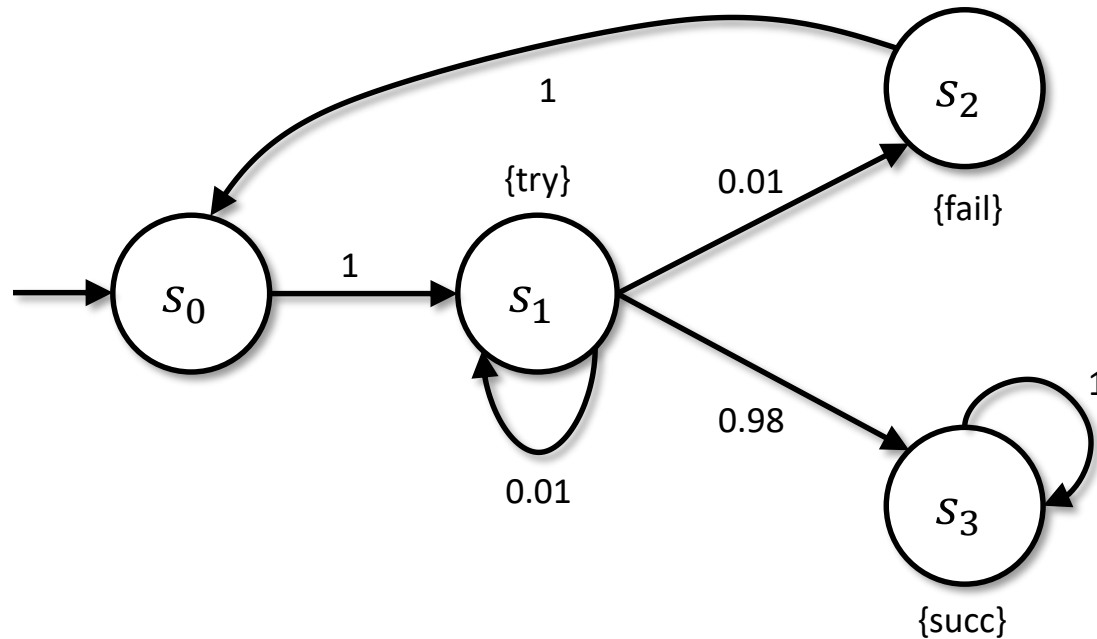
$\Phi U^{\leq k} \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\underline{Prob}^D(\text{try } U \text{ succ}) = (0, \frac{98}{99}, 0, 1)$$

$\Phi U^{\leq k} \Psi$ – for $k = \infty$ – Example



$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0.01 & 0.01 & 0.98 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\underline{Prob}^D(\text{try } U \text{ succ}) = (0, \frac{98}{99}, 0, 1)$$

$P_{>0.99}[\text{try } U \text{ succ}]$ is satisfied in s_3

Extending DTMC and PCTL with rewards

- DTMC and PCTL can be extended by rewards (or costs)
 - specification of cost for transition
 - reasoning about cost of particular computation, e.g., satisfying PCTL property, restricting to computations with cost less than k , ...

Continuous-time Markov Chains



Continuous-time Markov Chains

- Transitions are supposed to occur at real time
 - contrary to DTMC where they occur at discrete time steps
- CTMC allow to reason about different properties
 - Continuous Stochastic Logic (CSL) is used instead of PCTL
 - very close to PCTL including time specifications
 - support for specification of time intervals

Continuous-time Markov Chains

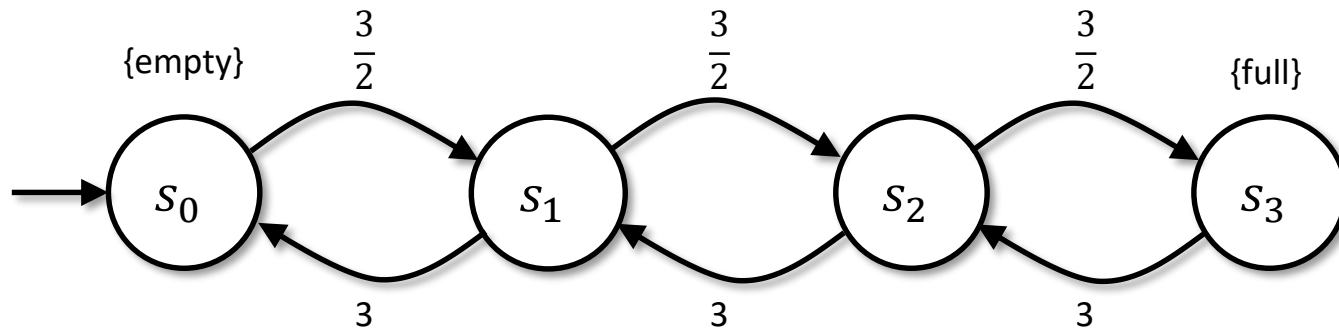
- Instead of probability matrix of DTMC, we have **transition rate matrix** (of real numbers)
 - assigns rates to each pair of states
 - rates determine the probability of the transition
 - exponential distribution – probability of transition (s, s') within t time units, if $\mathbf{R}(s, s') > 0$ equals

$$1 - e^{-\mathbf{R}(s, s') \cdot t}$$

- **Exit rate** $E(s)$ of state s is given by

$$E(s) \stackrel{\text{def}}{=} \sum_{s' \in S} \mathbf{R}(s, s')$$

CTMC – Example



$$\mathbf{R} = \begin{pmatrix} 0 & \frac{3}{2} & 0 & 0 \\ 3 & 0 & \frac{3}{2} & 0 \\ 0 & 3 & 0 & \frac{3}{2} \\ 0 & 0 & 3 & 0 \end{pmatrix}$$

$$\mathbf{p}_{emb} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{2}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{2}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

PRISM – Probabilistic Model Checker

- Allows for checking DTMC, CTMC and other types of models
- Uses simple dedicated input language
- <http://www.prismmodelchecker.org/>

```
// Two process mutual exclusion
mdp
module M1
    x : [0..2] init 0;
    [] x=0 -> 0.8:(x'=0) + 0.2:(x'=1);
    [] x=1 & y!=2 -> (x'=2);
    [] x=2 -> 0.5:(x'=2) + 0.5:(x'=0);
endmodule
module M2
    y : [0..2] init 0;
    [] y=0 -> 0.8:(y'=0) + 0.2:(y'=1);
    [] y=1 & x!=2 -> (y'=2);
    [] y=2 -> 0.5:(y'=2) + 0.5:(y'=0);
endmodule
```

