Probabilistic Model Checking

A few hints on probabilities in verification

Absolute guarantee of correctness?

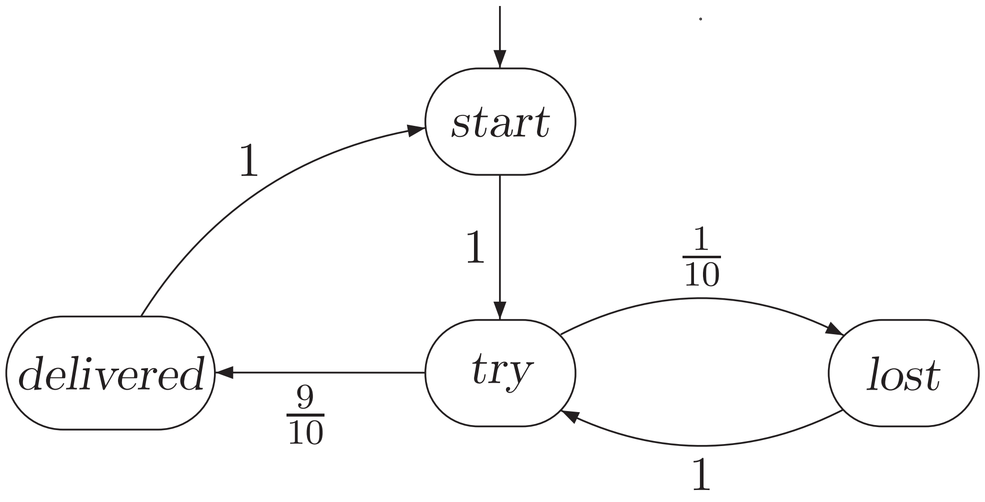
- Often system are subject to stochastic phenomena
  - Message loss, hardware failure, …
- They can initially be tackled with nondeterminism, but we'd like to have some bounds on probability
  - "If Messages can be lost 1% of the times then correct answers are delivered 90% of the times"
- For this, we need to extend our models to allow probabilities
  - Markov Chains, Markov Decision Processes
- And to verify their properties
  - Qualitative: "good eventually happens with probability 1", "bad happens with probability 0"
  - Quantitative: "Good eventually happens with at least 95% probability", "Bad happens with less than 5% probability"

Discrete-time Markov Chains

- Markov chains are the most popular operational model for the evaluation of *performance* and *dependability* of information-processing systems.
- Roughly speaking, Markov chains are transition systems with probability distributions for the successors of each state.
  - That is, instead of a nondeterministic choice, the next state is chosen probabilistically.
- They lack nondetermism, hence they cannot model interleaving behavior of concurrent processes

An example: lost messages

In state try, a message is delivered, but it is lost with probability 1/10 , in which case the message will be sent off again, until it is eventually delivered.



Formal Definition

A *(discrete-time) Markov chain* is a tuple $\mathcal{M} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ where

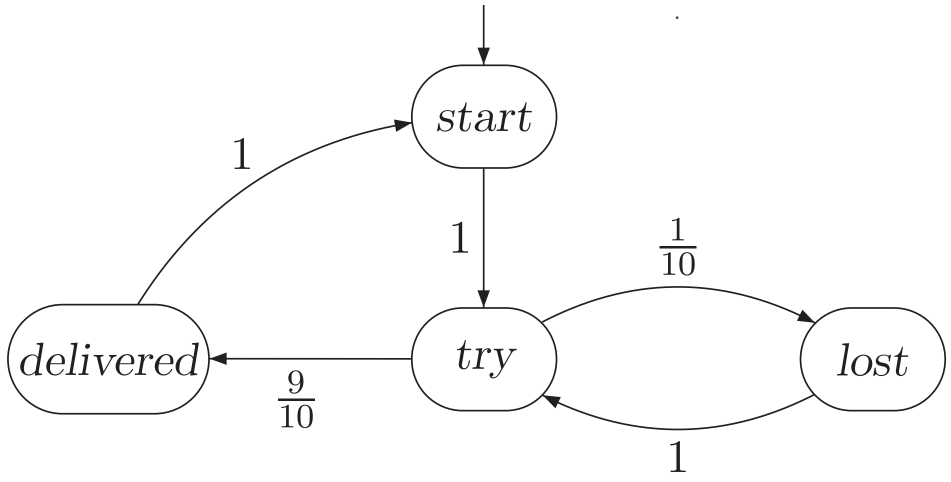
- S is a countable, nonempty set of states,
- $\mathbf{P} : S \times S \to [0, 1]$ is the *transition probability function* such that for all states s:

$$\sum_{s' \in S} \mathbf{P}(s, s') = 1,$$

- $\iota_{\text{init}} : S \to [0, 1]$ is the *initial distribution*, such that $\sum_{s \in S} \iota_{\text{init}}(s) = 1$, and
- AP is a set of atomic propositions and $L : S \to 2^{AP}$ a labeling function.

Example: transition probability matrix

Using the enumeration start, try, lost, delivered for the states, the transition probability function P can be viewed as a 4×4 matrix and the initial distribution as a column vector



$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{10} & \frac{9}{10} \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \qquad \iota_{\text{init}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

An example of a path is $\pi = (\mathit{start}\ \mathit{try}\ \mathit{lost}\ \mathit{try}\ \mathit{lost}\ \mathit{try}\ \mathit{delivered})^{\omega}$

Along this path each message has to be retransmitted two times before delivery. It follows that $\inf(\pi) = S$. For $T = \{\, \mathit{lost}, \mathit{delivered} \,\}$, we have $\mathbf{P}(\mathit{try}, T) = 1$. ■

LTL? CTL?

- By abstracting away probabilities, a DTMC is a transition system TS, obtained by considering non-zero probability initial states and transitions. Can we use LTL/CTL model checking?
- Yes, but in the example the following CTL formula holds:

$$start \models \exists \Box \neg delivered$$

- Because we can build an infinite path where a message is never delivered! But it is clear that this path has "0 probability"
- So we need a different logic!

LTL-style formulae and reachability probabilities

- We can use LTL to talk abou events, e.g. GF B describes the event that the subset B of states is visited infinitely often.
- Given a LTL formula φ the probability that φ holds in state s is:

$$Pr(s \models \varphi) \;=\; Pr_s\{\,\pi \in Paths(s) \mid \pi \models \varphi\,\}.$$

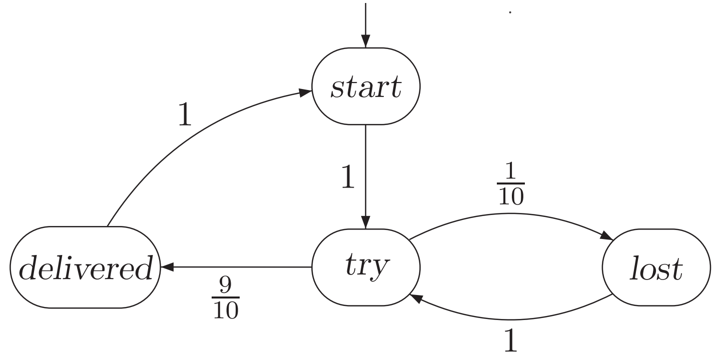
- Pr_s is the total probability of the sets of paths with initial state s, for which the path formula φ holds.

Example: the lossy channel M

- Can we reach the state «delivered»?
- We are interested in path fragments $s_0 \ldots s_n$ with s_n = *delivered* and

$$s_i \neq \mathit{delivered} \text{ for } 0 \leqslant i < n$$

$$\widehat{\pi}_n = \mathit{start}\,\mathit{try}\,(\mathit{lost}\,\mathit{try})^n\,\mathit{delivered}$$



- Probability of $\widehat{\pi_n}$ is $\left(\frac{1}{10}\right)^n \cdot \frac{9}{10}$

$$Pr^{\mathcal{M}}(\lozenge \mathit{delivered}) = \sum_{n=0}^{\infty} \left(\frac{1}{10}\right)^n \cdot \frac{9}{10} = \frac{\frac{9}{10}}{1 - \frac{1}{10}} = \frac{\frac{9}{10}}{\frac{9}{10}} = 1.$$

PCTL: Probabilistic Computation Tree Logic

- A branching time temporal logic based on CTL
- Syntax

$$\Phi \ ::= \ \text{true} \ \Big| \ a \ \Big| \ \Phi_1 \wedge \Phi_2 \ \Big| \ \neg\Phi \ \Big| \ \mathbb{P}_J(\varphi)$$

where $a \in AP$, φ is a path formula and $J \subseteq [0,1]$ is an interval with rational bounds. PCTL *path formulae* are formed according to the following grammar:

$$\varphi \ ::= \ \bigcirc\Phi \ \Big| \ \Phi_1 \, \mathsf{U} \, \Phi_2 \ \Big| \ \Phi_1 \, \mathsf{U}^{\leqslant n} \, \Phi_2$$

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

abbreviations are used; e.g., $\mathbb{P}_{\leqslant 0.5}(\varphi)$ denotes $\mathbb{P}_{[0,0.5]}(\varphi)$, $\mathbb{P}_{=1}(\varphi)$ stands for $\mathbb{P}_{[1,1]}(\varphi)$, and $\mathbb{P}_{>0}(\varphi)$ denotes $\mathbb{P}_{]0,1]}(\varphi)$.

Some intuition on non-probabilistic operators

The propositional logic fragment of PCTL, as well as the path formulae $\bigcirc \Phi$ and $\Phi_1 \, \mathsf{U} \, \Phi_2$ has the same meaning as in CTL. Path formula $\Phi_1 \, \mathsf{U}^{\leqslant n} \Phi_2$ is the *step-bounded* variant of $\Phi_1 \, \mathsf{U} \, \Phi_2$. It asserts that the event specified by Φ_2 will hold within at most n steps, while Φ_1 holds in all states that are visited before a Φ_2-state has been reached. Other Boolean connectives are derived in the usual way, e.g., $\Phi_1 \vee \Phi_2$ is obtained by $\neg(\neg\Phi_1 \wedge \neg\Phi_2)$. The eventually operator ($\Diamond$) can be derived as usual: $\Diamond\Phi = \text{true} \, \mathsf{U} \, \Phi$. Similarly, for step-bounded eventually we have:

$$\Diamond^{\leqslant n}\Phi = \text{true} \, \mathsf{U}^{\leqslant n} \Phi.$$

A path satisfies $\Diamond^{\leqslant n}\Phi$ if it reaches a Φ-state within n steps.

The always operator can be derived using the duality of eventually and always and the duality of lower and upper bounds.

$$\mathbb{P}_{\leqslant p}(\Box\Phi) = \mathbb{P}_{\geqslant 1-p}(\Diamond\neg\Phi) \quad \text{and} \quad \mathbb{P}_{]p,q]}(\Box^{\leqslant n}\Phi) = \mathbb{P}_{[1-q,1-p[}(\Diamond^{\leqslant n}\neg\Phi).$$

Usage of probabilistic operator

- Example: a 6-face die: 1 ,2,3,4 5 and 6 are the possible outcome
- If the die is fair, then: $$\bigwedge_{1 \leqslant i \leqslant 6} \mathbb{P}_{=\frac{1}{6}}(\Diamond i)$$
- A communication protocol with a lossy channel

$$\mathbb{P}_{=1}(\Diamond delivered) \;\wedge\; \mathbb{P}_{=1}\Big(\Box\,(\,try\_to\_send \;\rightarrow\; \mathbb{P}_{\geqslant 0.99}(\Diamond^{\leqslant 3} delivered)\,)\Big)$$

Almost surely some message will be delivered (first conjunct) and that almost surely for any attempt to send a message, with probability at least 0.99, the message will be delivered within three steps.

PCTL semantics

- Semantics for PCTL over a Markov chain has same definition as CTL over a TS for the "non-probabilistic fragment"
- The semantics of the probability operator for a state s is:

$$s \models \mathbb{P}_J(\varphi) \quad \text{iff} \quad Pr(s \models \varphi) \in J.$$

$$\text{Here, } Pr(s \models \varphi) \;=\; Pr_s\{\, \pi \in Paths(s) \mid \pi \models \varphi \,\}.$$

Pr_s is the total probability of the sets of paths with initial state s, for which the path formula φ holds.

PCTL semantics of the other operators

$$
\begin{array}{lll}
s \models a & \text{iff} & a \in L(s), \\
s \models \neg\Phi & \text{iff} & s \not\models \Phi, \\
s \models \Phi \wedge \Psi & \text{iff} & s \models \Phi \text{ and } s \models \Psi, \\
\pi \models \bigcirc \Phi & \text{iff} & \pi[1] \models \Phi, \\
\pi \models \Phi \,\mathsf{U}\, \Psi & \text{iff} & \exists\, j \geqslant 0.\ (\pi[j] \models \Psi \ \wedge\ (\forall\, 0 \leqslant k < j.\, \pi[k] \models \Phi))\,, \\
\pi \models \Phi \,\mathsf{U}^{\leqslant n}\, \Psi & \text{iff} & \exists\, 0 \leqslant j \leqslant n.\ (\pi[j] \models \Psi \ \wedge\ (\forall\, 0 \leqslant k < j.\, \pi[k] \models \Phi))
\end{array}
$$

where for path $\pi = s_0\, s_1\, s_2 \ldots$ and integer $i \geqslant 0$, $\pi[i]$ denotes the $(i{+}1)$-st state of π, i.e., $\pi[i] = s_i$. ■

PCTL Model Checking

The PCTL model-checking problem is the following decision problem. Given a finite Markov chain $\mathcal{M}$, state s in $\mathcal{M}$, and PCTL state formula Φ, determine whether $s \models \Phi$. As for CTL model checking, the basic procedure is to compute the satisfaction set $Sat(\Phi)$. This is done recursively using a bottom-up traversal of the parse tree of Φ; see Algorithm

Theorem 10.40. Time Complexity of PCTL Model Checking for MCs

For finite Markov chain $\mathcal{M}$ and PCTL formula Φ, the PCTL model-checking problem $\mathcal{M} \models \Phi$ can be solved in time

$$\mathcal{O}(\,\mathrm{poly}(\mathit{size}(\mathcal{M})) \cdot n_{\max} \cdot |\Phi|\,)$$

where $n_{\max}$ is the maximal step bound that appears in a subpath formula $\Psi_1 \, \mathsf{U}^{\leqslant n} \Psi_2$ of Φ (and $n_{\max} = 1$ if Φ does not contain a step-bounded until operator).

Qualitative fragment of PCTL

- Allowing only ">0" "=1", "=0", "<1" as probability bound
  - Useful when we do not need to have explicit probability
  - Model Checking becomes more efficient
- This fragment is NOT equivalent to CTL, since, e.g, "with probability 0" can be different from "never"

$$\mathbb{P}_{=1}(a \,\mathsf{U}\, b)$$

There is no CTL formula that is equivalent to $\mathbb{P}_{=1}(\Diamond a)$.

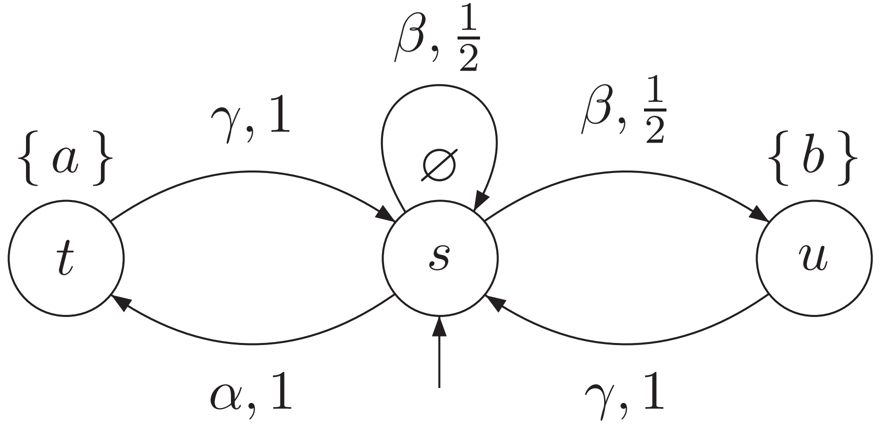
There is no CTL formula that is equivalent to $\mathbb{P}_{>0}(\Box a)$.

There is no qualitative PCTL formula that is equivalent to $\forall \Diamond a$.

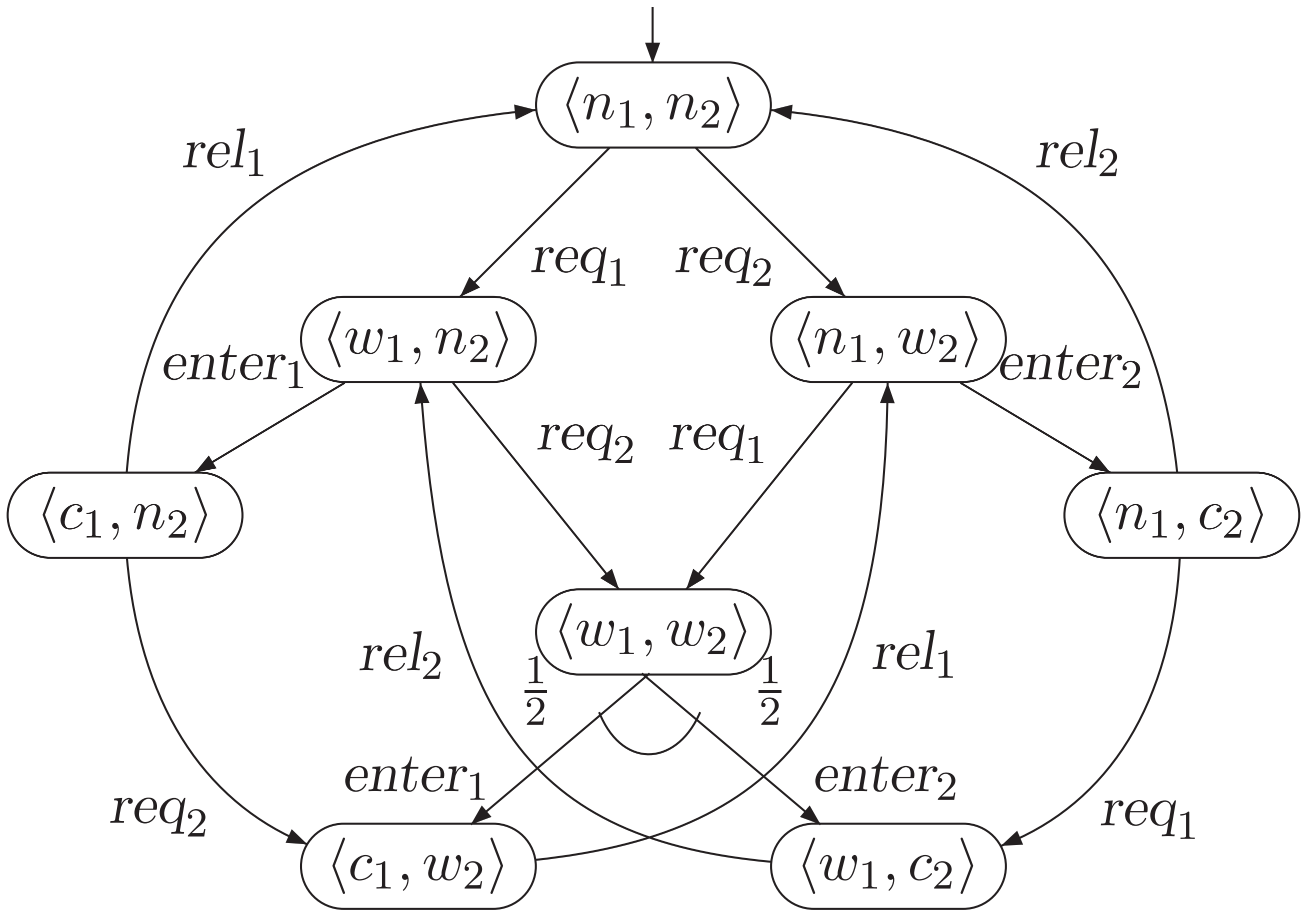
There is no qualitative PCTL formula that is equivalent to $\exists \Box a$.

Markov Decision Processes

- A variant of MC allowing nondeterminism
- Useful for modeling concurrent systems



A randomized mutual exclusion protocol



PCTL model checking over MDP

- Similar to the case for MC

Theorem 10.115. Time Complexity of PCTL Model Checking for MDPs

For finite MDP $\mathcal{M}$ and PCTL formula Φ, the PCTL model-checking problem $\mathcal{M} \models \Phi$ can be determined in time

$$\mathcal{O}(\,\mathrm{poly}(\mathit{size}(\mathcal{M})) \cdot n_{\max} \cdot |\Phi|\,)$$

where $n_{\max}$ is the maximal step bound that appears in a sub-path formula $\Psi_1 \, \mathsf{U}^{\leqslant n} \Psi_2$ of Φ (and $n_{\max} = 1$ if Φ does not contain a step-bounded until operator).