Formal Methods in Software Development

Ivano Salvo
and Igor Melatti

Computer Science Department

SAPIENZA
UNIVERSITÀ DI ROMA

Lesson **12**, December 17th, 2019

Lesson 12a:

Probabilistic CTL

Probabilistic CTL

Probabilistic CTL (PCTL) enrich the syntax of CTL with a **probabilistic operator** $\mathbf{P}_{[a,b]}(\varphi)$ whose intended semantics is that the probability of φ is in the interval $[a, b]$ (obviusly $a \geq 0$ and $a \leq b \leq 1$).

In PCTL we can define **quantitative properties** to be checked in a Markov chain. The **interpretation** of formula is **boolean**.

$\mathbf{P}_{[a,b]}(\varphi)$ is the probabilistic counterpart of the path quantifiers **E** and **A**.

Probabilistic CTL: Syntax

State formulae:

$$\psi \ ::= \ \text{true} \ | \ a \ | \ \psi_1 \wedge \psi_2 \ | \ \neg\psi \ | \ \mathbf{P}_J(\varphi)$$

where $a \in AP$, φ is a path formula, and $J \subseteq [0,1]$ is an interval with rational bounds.

Path formulae:

$$\varphi \ ::= \ \mathbf{X}\,\psi \ | \ \psi_1 \,\mathbf{U}\, \psi_2 \ | \ \psi_1 \,\mathbf{U}^{\leq n}\, \psi_2$$

where ψ_1, ψ_2 are state formulae and n is a natural number.

As in CTL, temporal operators **X** and **U** are required to be preceded by **P**. Intervals can be abbreviated: $\mathbf{P}_{\leq 0.5}$ means $\mathbf{P}_{[0,0.5]}$, $\mathbf{P}_{=1}$ means $\mathbf{P}_{[1,1]}$, and $\mathbf{P}_{>0}$ means $\mathbf{P}_{]0,1]}$ etc.

Semantic is similar to CTL. Step bounded until $\psi_1 \,\mathbf{U}^{\leq n}\, \psi_2$ requires that ψ_2 holds after at most n steps.

Probabilistic CTL: examples

The simulation of a six-sided die by a fair coin. The PCTL formula $\bigwedge_{i=1..6} \mathbf{P}_{=1/6}(\mathbf{F}\ i)$ asserts that each 6 possible outcomes have equal probability.

In the communication protocol, the PCTL formula:

$\mathbf{P}_{=1}(\mathbf{F}\ \text{delivered}) \wedge \mathbf{P}_{=1}(\mathbf{G}\ (\text{try} \rightarrow \mathbf{P}_{\geq 0.99}\ (\mathbf{F}^{\leq 3}\ \text{delivered})))$

Asserts that almost surely some message will be delivered and that almost surely, for any attempt to send a message, with probability at least 99% the message will be sent within 3 steps.

Probabilistic CTL: semantics

The semantics is the same as that of CTL, except for $\mathcal{P}_J(\varphi)$ and bounded until. We have:

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

$$\pi \models \psi_1 \, \mathbf{U}^{\leq n} \, \psi_2 \text{ iff } \exists\, 0 \leq j \leq n.\ \pi_j \models \psi_2 \wedge (\forall\, 0 \leq k < j.\ \pi_k \models \psi_1)$$

Formally, we need to check whether events specified by PCTL path formulae are measurable.

Theorem: For each PCTL path formula φ and state s of a Markov chain, $\text{Path}(s, \varphi) = \{\pi \in \text{Path}(s) \mid \pi \models \varphi\}$ is measurable.

Proof: Induction on φ. If $\varphi \equiv \mathbf{X}\, \varphi'$, then Path(s, φ) is the union of Path(t, φ'), such that $t \models \varphi'$. If $\varphi \equiv \psi_1 \, \mathbf{U}^{\leq n} \, \psi_2$, then Path(s, φ) is the union of all cylinder sets Cyl($s_0 s_1 \ldots s_k$), where $k \leq n$, $s_k \models \psi_2$ and $s_i \models \psi_1$ for $0 \leq i < k$. If $\varphi \equiv \psi_1 \, \mathbf{U} \, \psi_2$, then Path(s, φ) can be written as $\cup_{n \geq 0} \{\pi \in \text{Path}(s) \mid \pi \models \psi_1 \, \mathbf{U}^{\leq n} \, \psi_2\}$. □

Probabilistic CTL: equivalences

As usual, other operators, such as **F** and **R** as well as other boolean connectives can be derived using duality.
E.g. $\mathbf{F}^{\leq n}\,\psi \equiv \mathsf{true}\ \mathbf{U}^{\leq n}\,\psi$.

In general, we have that $\mathbf{P}_{<p}(\varphi) \equiv \mathbf{P}_{>p}(\neg\varphi)$ and $\mathbf{P}_{]a,b]}(\varphi) \equiv \neg\mathbf{P}_{\leq a}(\varphi) \wedge \mathbf{P}_{\leq b}(\varphi)$. Be careful to the duality between lower and upper bounds! Therefore we could limit to consider only upper-bounds and one between $\mathbf{P}_{=1}$ and $\mathbf{P}_{=0}$ for qualitative properties.

If an event *E* holds with probability **at most** *p*, then the complementary event *E* holds with probability **at least** 1-*p*. For example: $\mathbf{P}_{\leq p}(\mathbf{G}\ \varphi) \equiv \mathbf{P}_{\geq 1-p}(\mathbf{F}\ \neg\varphi)$ and $\mathbf{P}_{]p,q]}(\mathbf{G}^{\leq n}\ \varphi) \equiv \mathbf{P}_{[1-q,\ 1-p[}(\mathbf{G}^{\leq n}\ \varphi)$.

PCTL: proving equivalences

Let us consider the equivalence $\mathbf{P}_{\geq 0}(\mathbf{X}\ \mathbf{P}_{>0}(\mathbf{F}\ \psi)) \equiv \mathbf{P}_{>0}(\mathbf{F}\ \mathbf{P}_{>0}(\mathbf{X}\ \psi))$

($\Rightarrow$) Let s be such that $s \models \mathbf{P}_{\geq 0}(\mathbf{X}\ \mathbf{P}_{>0}(\mathbf{F}\ \psi))$, then there exists t, such that $\mathcal{P}(s, t)>0$ and $t \models \mathbf{P}_{>0}(\mathbf{F}\ \psi)$ and therefore there exists a finite path $t_0 t_1 \ldots t_k$ where $t=t_0$ and $t_k \models \varphi$. Therefore $t_{k-1} \models \mathbf{X}\ \psi$. Since $s\ t_0 t_1 \ldots t_{k-1}$ is a path fragment starting in s with positive probability, we have $s \models \mathbf{P}_{>0}(\mathbf{F}\ \mathbf{P}_{>0}(\mathbf{X}\ \psi))$.

($\Leftarrow$) Conversely, if $s \models \mathbf{P}_{>0}(\mathbf{F}\ \mathbf{P}_{>0}(\mathbf{X}\ \psi))$ then there exists a path fragment $s_0 s_1 \ldots s_k$ with $s=s_0$ and $s_k \models \mathbf{P}_{>0}(\mathbf{X}\ \psi)$, but this means that s_k has a successor t such that $t \models \psi$. This means that the path fragment $s_1 \ldots s_k t$ is a witness for $s_1 \models \mathbf{P}_{>0}(\mathbf{F}\ \psi)$ and hence $s \models \mathbf{P}_{\geq 0}(\mathbf{X}\ \mathbf{P}_{>0}(\mathbf{F}\ \psi))$.

PCTL model checking

The problem is to verify in a Markov chain if $s \models \varphi$, where φ is a PCTL formula. As for CTL, the idea is to compute set of states Sat(ψ) for all subformulae ψ of φ. For propositional sub-formulae, the problem is essentially the same as in CTL, so the interesting case is to determine $\text{Sat}(\mathbf{P}_J\ \psi) = \{s \in S \mid Pr(s \models \psi) \in J\}$.

For the operator X, it suffices to multiply the matrix $\mathcal{P}$ by the characteristic vector of Sat(ψ):

$$Pr(s \models \mathbf{X}\ \psi) = \sum_{s' \in \text{Sat}(\psi)} \mathcal{P}(s, s')$$

If we have formulae of the form $\psi_1\ \mathbf{U}^{\leq n}\ \psi_2$ or $\psi_1\ \mathbf{U}\ \psi_2$, we can just use technique we have seen for constrained reachability in the last lesson, where C=Sat(ψ_1) and B=Sat(ψ_2).

For the bounded operator $\mathbf{U}^{\leq n}$ we have to stop after n iterations.

PCTL model checking

Theorem: Let $\mathcal{M}$ be a finite MC and φ be a PCTL formula. The model checking problem $\mathcal{M} \models \varphi$ can be solved in time $\mathcal{O}(poly(size(\mathcal{M})) \cdot n_{\max} \cdot |\varphi|)$ where $n_{\max}$ is the maximum step bound that appears in formulae of the form $\psi_1 \, \mathbf{U}^{\leq n} \, \psi_2$.

For efficiency reasons, **qualitative properties** such as $\mathbf{P}_{=1}(\psi_1 \, \mathbf{U} \, \psi_2)$ or $\mathbf{P}_{>0}(\psi_1 \, \mathbf{U} \, \psi_2)$ are solved by using graph-based algorithms [this avoids solving systems of linear equations].

A **counterexample** or **witness** in PCTL is a set of path fragments that show the refutation or satisfaction of a formula.

Counterexamples and witnesses

Example: If $s \models \mathbf{P}_{\leq p}(\mathbf{F}\,\psi)$, then $Pr(s \models \mathbf{F}\,\psi) > p$. A proof is a set Π of finite path fragments such that all $\pi \in \Pi$, $\pi = s_0 s_1 \ldots s_k$, $s_k \models \psi$ and for $i<k$, $s_i \not\models \psi$ and $\sum_{\pi \in \Pi} Pr(\pi) > p$.

If $s \not\models \mathbf{P}_{\geq p}(\mathbf{F}\,\psi)$, is obtained by a set Π of path that refute $\mathbf{F}\,\psi$. These paths have the shape $\pi = s_0 s_1 \ldots s_k$, for $i \leq k$, $s_i \not\models \psi$ s_i, and s_k belongs to a BSCC C of $\mathcal{M}$ such that $C \cap \mathrm{Sat}(\psi) = \varnothing$. Moreover we must have that $\sum_{\pi \in \Pi} Pr(\pi) > 1-p$. The cylinder sets $Cyl(\pi)$ satisfies $\mathbf{G}\neg\psi$ paths.

To compute $Pr(s \models \mathbf{G}\neg\psi)$ it is necessary to consider paths that reach a BSCC T of $\mathcal{M}$ such that $C \cap \mathrm{Sat}(\psi) = \varnothing$ through $\neg\psi$ states: we can collect all such paths (increasing k) until the probability is greater than $1-p$.

PCTL model checking: Example

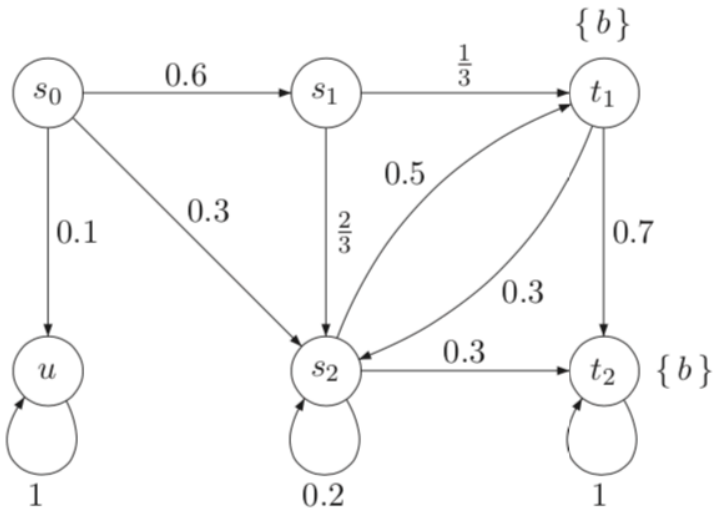
Let us consider the MC below. Let us assume that we are checking the property $\mathbf{P}_{\leq 1/2}(\mathbf{F}\, b)$ and that s_0 is the initial state.

$\mathcal{M} \models \mathbf{P}_{\leq 1/2}(\mathbf{F}\, b)$ is witnessed by three paths:

$$\{s_0 s_1 t_1,\ s_0 s_1 s_2 t_1\ ,\ s_0 s_2 t_1\}$$

whose probability is 0.2+0.2+0.15=0.55>0.5=1/2.

Observe that the counterexample is not unique. There are other paths such as $s_0 s_1 s_2 t_2$ and $s_0 s_2 t_2$.



Qualitative fragment of PCTL

The goal here is to compare the expressive power of PCTL wrt CTL. It is evident that quantitative properties cannot be expressed in CTL. But what about **qualitative properties**?

State formulae:

$$\psi \ ::= \ \text{true} \mid a \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \mathbf{P}_{>0}(\varphi) \mid \mathbf{P}_{=1}(\varphi)$$

where $a \in AP$, φ is a path formula.

Path formulae:

$$\varphi \ ::= \ \mathbf{X}\,\psi \mid \psi_1 \,\mathbf{U}\, \psi_2$$

where ψ_1, ψ_2 are state formulae.

Observations: $\mathbf{P}_{=0}(\varphi)=\neg\mathbf{P}_{>0}(\varphi)$ and $\mathbf{P}_{<1}(\varphi)=\neg\mathbf{P}_{=1}(\varphi)$.

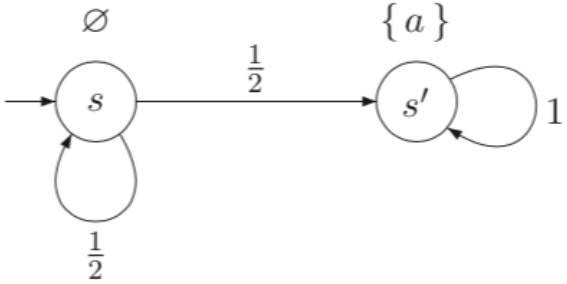
Definition: The PCTL formula φ is **equivalent** to the CTL formula ψ, notation $\varphi \equiv \psi$ iff $\text{Sat}(\varphi)=\text{Sat}(\psi)$ for all MC $\mathcal{M}$.

"Trivial" Equivalences

It is well-known that almost surely differs from all, because of some path with zero probability. In the MC below, we have $s \models \mathbf{P}_{=1}(\mathbf{F}\ a)$ but $s \not\models \mathbf{A}\ \mathbf{F}\ a$. The converse always holds.

For certain formulae, $\mathbf{P}_{=1}$ corresponds to $\mathbf{A}$ and $\mathbf{P}_{>0}$ corresponds to $\mathbf{E}$. For example: $s \models \mathbf{P}_{=1}(\mathbf{X}\ \varphi) \Leftrightarrow s \models \mathbf{A}\ \mathbf{X}\ \varphi$ and $s \models \mathbf{P}_{>0}(\mathbf{X}\ \varphi) \Leftrightarrow s \models \mathbf{E}\ \mathbf{X}\ \varphi$.

We have: $s \models \mathbf{P}_{>0}(\mathbf{F}\ \varphi) \Leftrightarrow s \models \mathbf{E}\ \mathbf{F}\ \varphi$ and $s \models \mathbf{P}_{=1}(\mathbf{G}\ \varphi) \Leftrightarrow s \models \mathbf{A}\ \mathbf{G}\ \varphi$



We show how to prove this statements:

Assuming $s \models \mathbf{P}_{>0}(\mathbf{F}\ \varphi)$, we have $Pr(s \models \mathbf{F}\ \varphi)>0$ that implies that there exists a finite path fragment whose last state satisfies φ. But this path fragment is a witness of $s \models \mathbf{E}\ \mathbf{F}\ \varphi$ in CTL.

Conversely, assuming $s \models \mathbf{E}\ \mathbf{F}\ \varphi$ we have that there exist a finite path fragment and its cylinder satisfies $s \models \mathbf{P}_{>0}(\mathbf{F}\ \varphi)$.

The other statement follows by duality.

"Trivial" Equivalences

Theorem: There is no CTL formula equivalent to the following PCTL formulae: $\mathbf{P}_{=1}(\mathbf{F}\ a)$ and $\mathbf{P}_{>0}(\mathbf{G}\ a)$.

Proof (idea): If we consider the infinite random walk, the validity of the above PCTL formulae depend on probabilities on transitions, whereas CTL formulae depend only on the underlying graph. □

This theorem is **no longer true** on **finite Markov chains**: $\mathbf{P}_{=1}(\mathbf{F}\ a)$ is equivalent to the CTL formula $\mathbf{A}\ ((\mathbf{E}\ \mathbf{F}\ a)\ \mathbf{W}\ a)$, where **W** is the "weak until operator": $\varphi\ \mathbf{W}\ \psi \equiv (\varphi\ \mathbf{U}\ \psi) \vee \mathbf{G}\ \varphi$.

Surprisingly, we can prove the following:

Theorem: There is no qualitative PCTL formula equivalent to the following CTL formulae: $\mathbf{A}\ \mathbf{F}\ a$ and $\mathbf{E}\ \mathbf{G}\ a$.

PCTL and fairness

As we have seen, often a 0-probability loop makes the difference between a PCTL property $\mathbf{P}_{=1}(\boldsymbol{\varphi})$ and a CTL $\mathbf{A}\ \varphi$.

Let us define the following strong fairness constraints:

$$\text{sfair} = \bigwedge_{s \in S} \bigwedge_{t \in \text{post}(s)} \mathbf{GF}\ s \rightarrow \mathbf{GF}\ t$$

Then we have the following equivalences:

$s \models \mathbf{P}_{=1}(\varphi\ \mathbf{U}\ \psi) \Leftrightarrow s \models_{\text{sfair}} \mathbf{A}\ (\varphi\ \mathbf{U}\ \psi)$ and $s \models \mathbf{P}_{>0}(\mathbf{G}\ \varphi) \Leftrightarrow s \models_{\text{sfair}} \mathbf{E}\ \mathbf{G}\ \varphi$

Therefore, qualitative PCTL is a sort of CTL plus strong fairness.

Lesson 12b:

Linear Time Properties

Linear Time Properties

Linear Time Properties are a set of traces.

The model checking problem for them is:

"Given a finite MC $\mathcal{M}$, and a linear time property P, compute the probability of the set of path of $\mathcal{M}$ for which P holds."

Here, we address this problem for Safety Regular Propertis, and hint to the solution for ω-Regular Properties.

Definition: Let $\mathcal{M}$ be a MC, and P a ω-Regular Properties (both over the same set AP of atomic propositions). The **probability for $\mathcal{M}$ to exhibit a trace in P**, denote $Pr_{\mathcal{M}}(P)$, is:

$$Pr_{\mathcal{M}}(P) = Pr_{\mathcal{M}}\{ \pi \in \text{Paths}(\mathcal{M}) \mid \text{trace}(\pi) \in P \}$$

For a state s of $\mathcal{M}$, we write $Pr_{\mathcal{M}}(s \models P) = Pr_{\mathcal{M}}\{ \pi \in \text{Paths}(s) \mid \text{trace}(\pi) \in P \}$. For an LTL formula φ, $Pr_{\mathcal{M}}(\varphi) = Pr_{\mathcal{M}}(\text{words}(\varphi))$.

Linear Time Properties: idea

The basic idea **is exactly the same as for Linear Time Properties in classical Model Checking**: reduce the problem of computing $Pr_{\mathcal{M}}(P)$ to a reachability problem in a Markov chain $\mathcal{M}\otimes\mathcal{A}$ where $\mathcal{A}$ represent (the complement) of P.

The main difference is that to guarantee that $\mathcal{M}\otimes\mathcal{A}$ is Markov chain, we need that $\mathcal{A}$ is a deterministic automata.This is not a problem for **Regular Safety properties**, that are characterized as the complement of bad prefixes generated by a Deterministic Finite Automata (DFA): DFAs have the same expressive power of Non-deterministic Finite Automata (NFAs).

For **ω-Regular Properties**, we know that Deterministic Büchi Automata (DBAs) are strictly less expressive of their non-deterministic counterpart (NBAs).

For this reason, another family of automata, Deterministic Rabin Automata (DRA) is considered.

Regular Safety Properties

A safety property is regular whenever the set of bad prefixes is a regular language. Let $\mathcal{A}=(Q, 2^{AP}, \delta, q_0, F)$ be a DFA. We have: $P_{\text{safe}}=\{A_0A_1A_2\ldots \in (2^{AP})^{\omega} \mid A_0A_1A_2\ldots A_n \notin \mathcal{L}(\mathcal{A}) \}$. Our problem is to compute the probability:

$$Pr_{\mathcal{M}}(P_{\text{safe}}) = 1 - \sum_{s \in S} \iota(s) \cdot Pr(s \models \mathcal{A})$$

Where $Pr(s \models \mathcal{A})$ is given by:

$$Pr(s \models \mathcal{A}) = Pr_{\mathcal{M}}(\pi \in \text{Paths}(s) \mid \text{pref}(\text{trace}(\pi)) \cap \mathcal{L}(\mathcal{A}) \neq \varnothing)$$

$$= Pr_{\mathcal{M}}(\pi \in \text{Paths}(s) \mid \text{trace}(\pi) \notin P_{\text{safe}})$$

Therefore:

$$Pr(s \models \mathcal{A}) = \sum_{\pi \in \text{BP}(s)} Pr_{\mathcal{M}}(\pi)$$

where BP(s) is the set of minimal bad prefixes defined by $\mathcal{L}(\mathcal{A})$ starting in s. Computing these sums could be hard. We will adapt the automaton-based techniques for classical MC.

Product Markov Chain

Definition 10.50. Product Markov Chain

Let $\mathcal{M} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a Markov chain and $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be a DFA. The product $\mathcal{M} \otimes \mathcal{A}$ is the Markov chain:

$$\mathcal{M} \otimes \mathcal{A} = (S \times Q, \mathbf{P}', \iota'_{\text{init}}, \{ accept \}, L')$$

where $L'(\langle s, q \rangle) = \{ accept \}$ if $q \in F$ and $L'(\langle s, q \rangle) = \varnothing$ otherwise, and

$$\iota'_{\text{init}}(\langle s, q \rangle) = \begin{cases} \iota_{\text{init}}(s) & \text{if } q = \delta(q_0, L(s)) \\ 0 & \text{otherwise.} \end{cases}$$

The transition probabilities in $\mathcal{M} \otimes \mathcal{A}$ are given by

$$\mathbf{P}'(\langle s, q \rangle, \langle s', q' \rangle) = \begin{cases} \mathbf{P}(s, s') & \text{if } q' = \delta(q, L(s')) \\ 0 & \text{otherwise.} \end{cases}$$

■

Note that: **1**) Each state $\langle s, q \rangle$ in $\mathcal{M} \otimes \mathcal{A}$ records the state in $\mathcal{A}$ for the path fragment taken so far in $\mathcal{M}$. **2**) The deterministic automata $\mathcal{A}$ does not affect probabilities.

Quant. analysis for regular safety

Theorem 10.51. Quantitative Analysis for Safety Properties

Let P_{safe} be a regular safety property, $\mathcal{A}$ a DFA for the set of bad prefixes of P_{safe}, $\mathcal{M}$ a Markov chain, and s a state in $\mathcal{M}$. Then:

$$\begin{aligned} Pr^{\mathcal{M}}(s \models P_{safe}) &= Pr^{\mathcal{M}\otimes\mathcal{A}}(\langle s, q_s\rangle \not\models \Diamond accept) \\ &= 1 - Pr^{\mathcal{M}\otimes\mathcal{A}}(\langle s, q_s\rangle \models \Diamond accept) \end{aligned}$$

where $q_s = \delta(q_0, L(s))$.

In this way, we have reduced the problem of quantitative analysis of Regular Safety problem to a quantitative reachability (see lesson **11**).

Qualitative analysis is similar, following qualitative reachability.

Unfortunately, **this does not extend** to the more general ω-Regular Properties because, in general, they require Non-deterministic Büchi automata and **non-determinism affects probabilities**!

Deterministic Rabin Automata

Definition 10.53. Deterministic Rabin Automaton (DRA)

A *deterministic Rabin automaton* (DRA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, Acc)$ where Q is a finite set of states, Σ an alphabet, $\delta : Q \times \Sigma \to Q$ the transition function, $q_0 \in Q$ the starting state, and

$$Acc \subseteq 2^Q \times 2^Q.$$

A run for $\sigma = A_0 A_1 A_2 \ldots \in \Sigma^\omega$ denotes an infinite sequence $q_0\, q_1\, q_2 \ldots$ of states in $\mathcal{A}$ such that $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geqslant 0$. The run $q_0\, q_1\, q_2 \ldots$ is *accepting* if there exists a pair $(L, K) \in Acc$ such that

$$(\exists n \geqslant 0. \forall m \geqslant n. q_m \notin L) \ \wedge \ \left(\overset{\infty}{\exists}\, n \geqslant 0. q_n \in K\right).$$

The *accepted language* of $\mathcal{A}$ is

$$\mathcal{L}_\omega(\mathcal{A}) \ = \ \{\, \sigma \in \Sigma^\omega \mid \text{the run for } \sigma \text{ in } \mathcal{A} \text{ is accepting} \,\}.$$

■

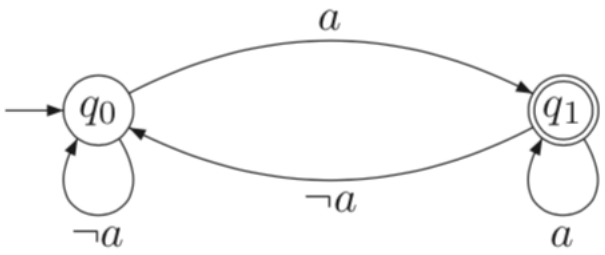
A computation is accepting in a DRA if there exists a pair of set of states $(L_i, K_i) \in Acc$ such that states in L_i are visited only finitely many times and states of K_i are visited infinitely often. Accepting paths satisfies the LTL formula $\bigvee_i (\mathbf{FG}\neg L_i \wedge \mathbf{GF}\, K_i)$. Büchi automata are a particular case where Acc=$\{(\varnothing, F)\}$

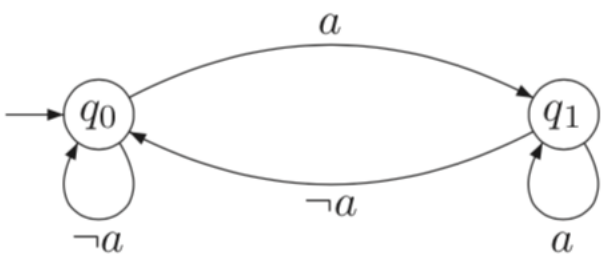
DRAs: Example

Let us consider the LTL property: **GF** a. This property can be modeled by a DBA (left). The corresponding DRA is on the right, where $Acc = \{(\varnothing, \{q_1\})\}$.

Properties such as **FG** a are not expressible by DBA. On the other hand the same DRA with $Acc = \{(\{q_0\}, \{q_1\})\}$ accepts the infinite words whose ends with an infinite suffix that never visit q_0 and these are exactly the words in words(**FG** a).

Theorem: The class of languages accepted by DRAs agrees with the class of ω-regular languages.





Linear Time Prop.: conclusions

Theorem 10.56. DRA-Based Analysis of Markov Chains

Let $\mathcal{M}$ be a finite Markov chain, s a state in $\mathcal{M}$, $\mathcal{A}$ a DRA, and let U be the union of all accepting BSCCs in $\mathcal{M} \otimes \mathcal{A}$. Then:

$$Pr^{\mathcal{M}}(s \models \mathcal{A}) \;=\; Pr^{\mathcal{M}\otimes\mathcal{A}}(\,\langle s, q_s\rangle \models \Diamond U\,)$$

where $q_s = \delta(q_0, L(s))$.

The overall procedure is polynomial in $|\mathcal{M}\otimes\mathcal{A}|$.

On the other hand, it can be shown that trasforming a LTL properties φ into the corresponding DRA automata **is double exponential in $|\varphi|$, that is $2^{2^{|\varphi|}}$**. There are some more sophisticated techniques that avoid this complexity.

However, in any case, we have the following:

Theorem 10.58.

The qualitative model-checking problem for finite Markov chains is PSPACE*-complete.*

Lesson 12c:

Probabilistic bisimulation

Probabilistic CTL*: Syntax

State formulae:

$$\psi \ ::= \ \text{true} \mid \ a \ \mid \psi_1 \wedge \psi_2 \ \mid \ \neg\psi \ \mid \ \mathbf{P}_J(\varphi)$$

where $a \in AP$, φ is a path formula, and $J \subseteq [0,1]$ is an interval with rational bounds.

PCTL* path formulae:

$$\varphi \ ::= \psi \mid \mathbf{X}\,\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2 \ \mid \varphi_1 \wedge \varphi_2 \mid \neg\, \varphi$$

where ψ_1, ψ_2 are PCTL* state formulae.

Exercise: Define the bounded until operator.

According to the non-probabilistic case, model checking problem for PCTL* can be solved by alternating the PCTL procedure and LTL, thus obtaing an algorithm exponential in $|\varphi|$ and the problem has been shown to be PSPACE-complete.

Probabilistic bisimulation

Definition 10.60. Bisimulation for Markov Chains

Let $\mathcal{M} = (S, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be a Markov chain. A *probabilistic bisimulation* on $\mathcal{M}$ is an equivalence relation $\mathcal{R}$ on S such that for all states $(s_1, s_2) \in \mathcal{R}$:

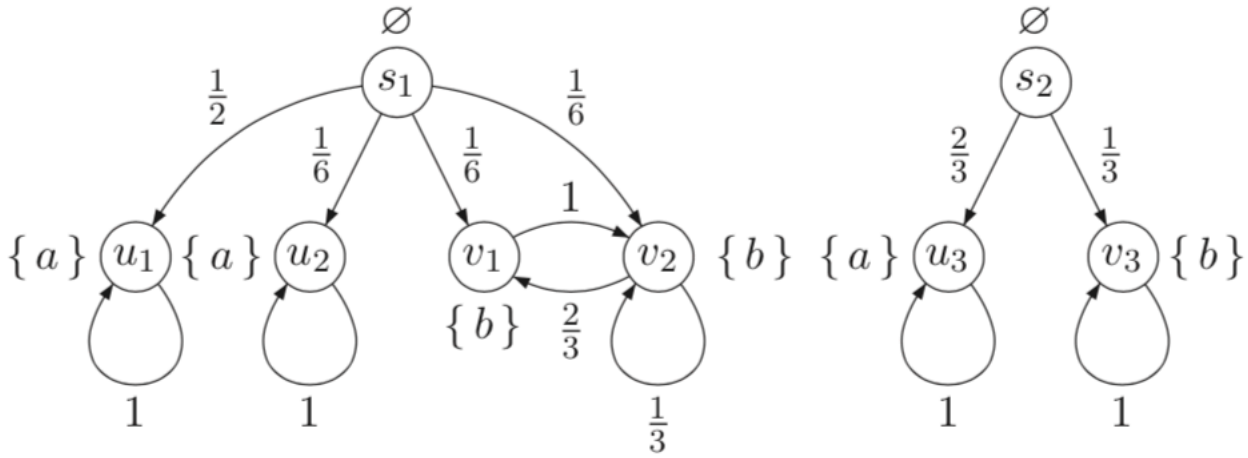
1. $L(s_1) = L(s_2)$.
2. $\mathbf{P}(s_1, T) = \mathbf{P}(s_2, T)$ for each equivalence class $T \in S/\mathcal{R}$.

States s_1 and s_2 are *bisimulation-equivalent* (or bisimilar), denoted $s_1 \sim_{\mathcal{M}} s_2$, if there exists a bisimulation $\mathcal{R}$ on $\mathcal{M}$ such that $(s_1, s_2) \in \mathcal{R}$. ■

The main difference wrt bisimulation for Transition Systems is that here **we require that each bisimulation is a equivalence relation** (otherwise this definition does not make sense).

Remembere that, in any case, the **greatest bisimulation is always an equivalence relation**.

Probabil. bisimulation: example



The reflexive, symmetric and transitive closure of the relation: $\mathcal{R} = \{(s_1, s_2), (u_1, u_3), (u_2, u_3), (v_1, v_3), (v_2, v_3)\}$ is a probabilistic bisimulation. To see this, we have first to devise equivalence classes, that are: $T_1=\{s_1, s_2\}$, $T_2=\{u_1, u_2 , u_3\}$, and $T_3=\{v_1, v_2 , v_3\}$.

We finally check that $\mathcal{P}(s_1, T_1)=\mathcal{P}(s_2, T_1)=0$, $\mathcal{P}(s_1, T_2)=\mathcal{P}(s_2, T_2)=2/3$, and $\mathcal{P}(s_1, T_3)=\mathcal{P}(s_2, T_3)=1/3$ [and so on].

Bisimulation char. of PCTL(*)

Probabilistic bisimulation preserves all PCTL*-definable quantitative properties.

Probabilistic bisimulation is the coarsest equivalence enjoying this property.

Non-equivalent states can be distinguished by using PCTL*.

This is formally stated by the following:

Theorem: Let $\mathcal{M}$ be a MC and s_1, s_2 states of $\mathcal{M}$. Then the following statements are equivalent:

- s_1 and s_2 are probabilistic bisimilar
- s_1 and s_2 fulfill the same set of PCTL/PCTL* formulae

Lesson 12

Merry Crhistmas and Happy New Year…

…Questions?