# Real-time and Probabilistic Systems Verification

Luca Tesei

MSc in Computer Science, University of Camerino

## Topics

- Probabilities in Markov Decision Processes
- Positional and Finite-Memory Policies
- Reachability Properties

More:

The slides in the following pages are taken from the material of the course “Modelling and Verification of Probabilistic Systems” held by Prof. Dr. Ir. Joost-Pieter Katoen at Aachen University.

# Modeling and Verification of Probabilistic Systems

## Lecture 11: Reachability Probabilities in MDPs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://www-i2.informatik.rwth-aachen.de/i2/mvps11/

May 30, 2011

## Overview

1. Markov Decision Processes
2. Probabilities in MDPs
3. Policies
   - Positional policies
   - Finite-memory policies
4. Reachability probabilities
   - Mathematical characterisation
   - Value iteration
   - Linear programming
5. Summary

## Overview

1. Markov Decision Processes
2. Probabilities in MDPs
3. Policies
   - Positional policies
   - Finite-memory policies
4. Reachability probabilities
   - Mathematical characterisation
   - Value iteration
   - Linear programming
5. Summary

## Markov decision process (MDP)

### Markov decision processes

- In MDPs, both nondeterministic and probabilistic choices coexist.
- MDPs are transition systems in which in any state a nondeterministic choice between probability distributions exists.
- Once a probability distribution has been chosen nondeterministically, the next state is selected probabilistically—as in DTMCs.
- Any MC is thus an MDP in which in any state the probability distribution is uniquely determined.

Randomized distributed algorithms are typically appropriately modeled by MDPs, as probabilities affect just a small part of the algorithm and nondeterminism is used to model concurrency between processes by means of interleaving.

# Markov decision process (MDP)

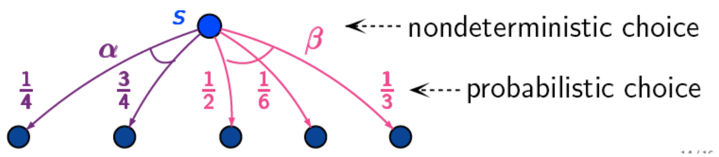
## Markov decision process

An MDP $\mathcal{M}$ is a tuple $(S, Act, \mathbf{P}, \iota_{\text{init}}, AP, L)$ where

- $S$ is a countable set of states with initial distribution $\iota_{\text{init}} : S \to [0, 1]$
- $Act$ is a finite set of actions
- $\mathbf{P} : S \times Act \times S \to [0, 1]$, transition probability function such that:

$$\text{for all } s \in S \text{ and } \alpha \in Act : \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$$

- $AP$ is a set of atomic propositions and labeling $L : S \to 2^{AP}$.



nondeterministic choice

probabilistic choice

# Markov decision process (MDP)

## Markov decision process

An MDP $\mathcal{M}$ is a tuple $(S, Act, \mathbf{P}, \iota_{\text{init}}, AP, L)$ where

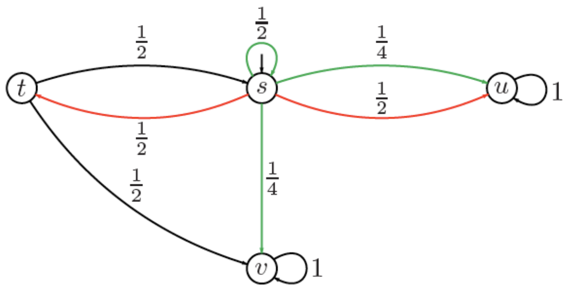
- $S$, $\iota_{\text{init}} : S \to [0, 1]$, $AP$ and $L$ are as before, i.e., as for DTMCs, and
- $Act$ is a finite set of actions
- $\mathbf{P} : S \times Act \times S \to [0, 1]$, transition probability function such that:

$$\text{for all } s \in S \text{ and } \alpha \in Act : \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$$

## Enabled actions

Let $Act(s) = \{ \alpha \in Act \mid \exists s' \in S. \mathbf{P}(s, \alpha, s') > 0 \}$ be the set of enabled actions in state $s$. We require $Act(s) \neq \varnothing$ for any state $s$.

# An example MDP



- Initial distribution: $\iota_{\text{init}}(s) = 1$ and $\iota_{\text{init}}(t) = \iota_{\text{init}}(u) = \iota_{\text{init}}(u) = 0$
- Set of enabled actions in state $s$ is $Act(s) = \{ \alpha, \beta \}$ where
  - $\mathbf{P}(s, \alpha, s) = \frac{1}{2}$, $\mathbf{P}(s, \alpha, t) = 0$ and $\mathbf{P}(s, \alpha, u) = \mathbf{P}(s, \alpha, v) = \frac{1}{4}$
  - $\mathbf{P}(s, \beta, s) = \mathbf{P}(s, \beta, v) = 0$, and $\mathbf{P}(s, \beta, t) = \mathbf{P}(s, \beta, u) = \frac{1}{2}$
- $Act(t) = \{ \alpha \}$ with $\mathbf{P}(t, \alpha, s) = \mathbf{P}(t, \alpha, u) = \frac{1}{2}$ and 0 otherwise

# Overview

1 Markov Decision Processes

2 Probabilities in MDPs

3 Policies
- Positional policies
- Finite-memory policies

4 Reachability probabilities
- Mathematical characterisation
- Value iteration
- Linear programming

5 Summary

# Paths in an MDP

## State graph

The *state graph* of MDP $\mathcal{M}$ is a digraph $G = (V, E)$ with $V$ are the states of $M$, and $(s, s') \in E$ iff $\mathbf{P}(s, \alpha, s') > 0$ for some $\alpha \in Act$.

## Paths

An infinite *path* in an MDP $\mathcal{M} = (S, Act, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ is an infinite sequence $s_0\, \alpha_1\, s_1\, \alpha_2\, s_2\, \alpha_3 \ldots \in (S \times Act)^\omega$, written as

$$\pi \;=\; s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots,$$

such that $\mathbf{P}(s_i, \alpha_{i+1}, s_{i+1}) > 0$ for all $i \geqslant 0$. Any finite prefix of $\pi$ that ends in a state is a *finite path*.

Let $Paths(\mathcal{M})$ denote the set of paths in $\mathcal{M}$, and $Paths^*(\mathcal{M})$ the set of finite prefixes thereof.

# Pre- and post

## Direct successors and predecessors of a state

For $s \in S$, $\alpha \in Act$ and $T \subseteq S$, let $\mathbf{P}(s, \alpha, T)$ denote the probability of moving to a state in $T$ via $\alpha$, i.e.,

$$\mathbf{P}(s, \alpha, T) \;=\; \sum_{t \in T} \mathbf{P}(s, \alpha, t).$$

$Post(s, \alpha)$ denotes the set of $\alpha$-successors of $s$:

$$Post(s, \alpha) \;=\; \{\, t \in S \mid \mathbf{P}(s, \alpha, t) > 0 \,\}.$$

Note: $Post(s, \alpha) = \varnothing$ if and only if $\alpha \notin Act(s)$.

$Pre(t)$ denotes the set of pairs $(s, \alpha)$ with $s \in S$ and $\alpha \in Act(s)$ such that $t \in Post(s, \alpha)$, i.e.,

$$Pre(t) \;=\; \{\, (s, \alpha) \in S \times Act \mid \mathbf{P}(s, \alpha, t) > 0 \,\}.$$

$Pre(G) = \bigcup_{s \in G} Pre(s)$ and $Pre^*(G)$ is the reflexive and transitive closure of $Pre(G)$.

# Overview

1 Markov Decision Processes

2 Probabilities in MDPs

3 Policies
- Positional policies
- Finite-memory policies

4 Reachability probabilities
- Mathematical characterisation
- Value iteration
- Linear programming

5 Summary

# Policies

## Policy

Let $\mathcal{M} = (S, Act, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be an MDP. A *policy* for $\mathcal{M}$ is a function $\mathfrak{S} : S^+ \to Act$ such that $\mathfrak{S}(s_0\, s_1 \ldots s_n) \in Act(s_n)$ for all $s_0\, s_1 \ldots s_n \in S^+$.

The path

$$\pi \;=\; s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$$

is called a $\mathfrak{S}$-path if $\alpha_i = \mathfrak{S}(s_0 \ldots s_{i-1})$ for all $i > 0$.

For any policy, the actions are omitted from the *history* $s_0\, s_1 \ldots s_n$. This is not a restriction as for any sequence $s_0\, s_1 \ldots s_n$ the relevant actions $\alpha_i$ are given by $\alpha_{i+1} = \mathfrak{S}(s_0\, s_1 \ldots s_i)$. Hence, the scheduled action sequence can be constructed from prefixes of the path at hand.

# Induced DTMC of an MDP by a policy

## DTMC of an MDP induced by a policy

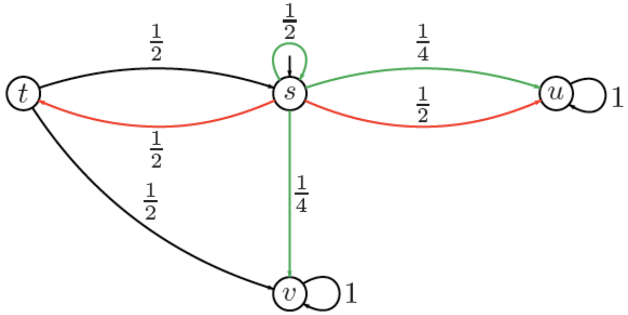
Let $\mathcal{M} = (S, Act, \mathbf{P}, \iota_{\text{init}}, AP, L)$ be an MDP and $\mathfrak{S}$ a policy on $\mathcal{M}$. The DTMC induced by $\mathfrak{S}$, denoted $\mathcal{M}_{\mathfrak{S}}$, is given by

$$\mathcal{M}_{\mathfrak{S}} \;=\; (S^+, \mathbf{P}_{\mathfrak{S}}, \iota_{\text{init}}, AP, L')$$

where for $\sigma = s_0 s_1 \ldots s_n$: $\mathbf{P}_{\mathfrak{S}}(\,\sigma,\;\sigma\, s_{n+1}\,) \;=\; \mathbf{P}(\,s_n,\,\mathfrak{S}(\sigma),\,s_{n+1}\,)$ and $L'(\sigma) \;=\; L(s_n)$.

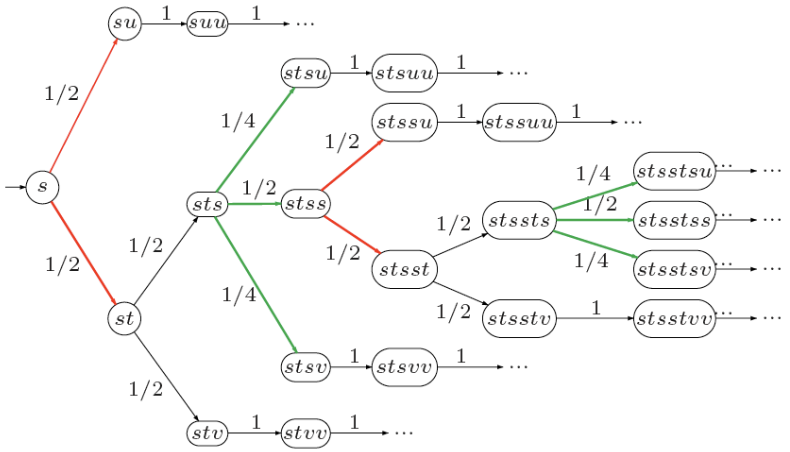
$\mathcal{M}_{\mathfrak{S}}$ is infinite, even if the MDP $\mathcal{M}$ is finite. Intuitively, state $s_0\, s_1 \ldots s_n$ of DTMC $\mathcal{M}_{\mathfrak{S}}$ represents the configuration where the MDP $\mathcal{M}$ is in state $s_n$ and $s_0\, s_1 \ldots s_{n-1}$ stands for the history. Since policy $\mathfrak{S}$ might select different actions for finite paths that end in the same state $s$, a policy as defined above is also referred to as *history-dependent*.

# Example MDP



Consider a policy that alternates between selecting red and green, starting with red.

# Example induced DTMC



Induced DTMC for a policy that alternates between selecting red and green.

# Probability measure on MDP

## Probability measure on MDP

Let $Pr^{\mathcal{M}}_{\mathfrak{S}}$, or simply $Pr^{\mathfrak{S}}$, denote the probability measure $Pr^{\mathcal{M}_{\mathfrak{S}}}$ associated with the DTMC $\mathcal{M}_{\mathfrak{S}}$.

This measure is the basis for associating probabilities with events in the MDP $\mathcal{M}$. Let, e.g., $P \subseteq (2^{AP})^{\omega}$ be an $\omega$-regular property. Then $Pr^{\mathfrak{S}}(P)$ is defined as:

$$Pr^{\mathfrak{S}}(P) \;=\; Pr^{\mathcal{M}_{\mathfrak{S}}}(P) \;=\; Pr_{\mathcal{M}_{\mathfrak{S}}}\{\,\pi \in Paths(\mathcal{M}_{\mathfrak{S}}) \mid trace(\pi) \in P\,\}.$$

Similarly, for fixed state $s$ of $\mathcal{M}$, which is considered as the unique starting state,

$$Pr^{\mathfrak{S}}(s \models P) \;=\; Pr_s^{\mathcal{M}_{\mathfrak{S}}}\{\,\pi \in Paths(s) \mid trace(\pi) \in P\,\}$$

where we identify the paths in $\mathcal{M}_{\mathfrak{S}}$ with the corresponding $\mathfrak{S}$-paths in $\mathcal{M}$.

# Positional policy

## Positional policy

Let $\mathcal{M}$ be an MDP with state space $S$. Policy $\mathfrak{S}$ on $\mathcal{M}$ is *positional* (or: *memoryless*) iff for each sequence $s_0\, s_1 \ldots s_n$ and $t_0\, t_1 \ldots t_m \in S^+$ with $s_n = t_m$:

$$\mathfrak{S}(s_0\, s_1 \ldots s_n) \;=\; \mathfrak{S}(t_0\, t_1 \ldots t_m).$$

In this case, $\mathfrak{S}$ can be viewed as a function $\mathfrak{S} : S \to Act$.

Policy $\mathfrak{S}$ is positional if it always selects the same action in a given state. This choice is independent of what has happened in the history, i.e., which path led to the current state.

# Finite-memory policy

## Finite-memory policy

Let $\mathcal{M}$ be an MDP with state space $S$ and action set $Act$. A *finite-memory policy* $\mathfrak{S}$ for $\mathcal{M}$ is a tuple $\mathfrak{S} = (Q, act, \Delta, start)$ where

- $Q$ is a finite set of modes,
- $\Delta : Q \times S \to Q$ is the transition function,
- $act : Q \times S \to Act$ is a function that selects an action $act(q, s) \in Act(s)$ for any mode $q \in Q$ and state $sinS$ of $\mathcal{M}$,
- $start : S \to Q$ is a function that selects a *starting mode* for state $s$ of $\mathcal{M}$.

# An MDP under a finite-memory policy

The behavior of an MDP $\mathcal{M}$ under a fm-policy $\mathfrak{S} = (Q, act, \Delta, start)$ is:

- Initially, a starting state $s_0$ is randomly determined according to the initial distribution $\iota_{\text{init}}$, i.e., $\iota_{\text{init}}(s_0) > 0$.
- The fm-policy $\mathfrak{S}$ initializes its DFA to the mode $q_0 = start(s_0) \in Q$.
- If $\mathcal{M}$ is in state $s$ and the current mode of $\mathfrak{S}$ is $q$, then the decision of $\mathfrak{S}$, i.e., the selected action, is $\alpha = act(q, s) \in Act(s)$.
- The policy changes to mode $\Delta(q, s)$, while $\mathcal{M}$ performs the selected action $\alpha$ and randomly moves to the next state according to the distribution $\mathbf{P}(s, \alpha, \cdot)$.

# Finite-memory policies

## Relation fm-policy to definition policy

An fm-policy $\mathfrak{S} = (Q, act, \Delta, start)$ is identified with policy, $\mathfrak{S}' : Paths^* \to Act$ which is defined as follows.

1. For the starting state $s_0$, let $\mathfrak{S}'(s_0) = act(start(s_0), s_0)$.
2. For path fragment $\widehat{\pi} \;=\; s_0\, s_1 \ldots s_n$ let

$$\mathfrak{S}'(\widehat{\pi}) \;=\; act(q_n, s_n)$$

where $q_0 = start(s_0)$ and $q_{i+1} = \Delta(q_i, s_i)$ for $0 \leqslant i \leqslant n$.

Positional policies can be considered as fm-policies with just a single mode.

# The DTMC under an fm-policy

## Remark

For fm-policy $\mathfrak{S}$, the DTMC $\mathcal{M}_{\mathfrak{S}}$ can be identified with a DTMC where the states are just pairs $\langle s, q \rangle$ where $s$ is a state in the MDP $\mathcal{M}$ and $q$ a mode of $\mathfrak{S}$.

Formally, $\mathcal{M}'_{\mathfrak{S}}$ is the DTMC with state space $S \times Q$, labeling $L'(\langle s, q \rangle) = L(s)$, the starting distribution $\iota_{\text{init}}$, and the transition probabilities:

$$\mathbf{P}'_{\mathfrak{S}}(\langle s, q \rangle, \langle t, p \rangle) \; = \; \mathbf{P}(s, act(q, s), t).$$

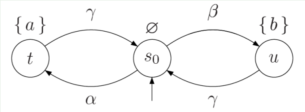
For any MDP $\mathcal{M}$ anf fm-policy $\mathfrak{S}$: $\mathcal{M}_{\mathfrak{S}} \;\sim_p\; \mathcal{M}'_{\mathfrak{S}}$.

Hence, if $\mathcal{M}$ is a finite MDP, then we consider $\mathcal{M}_{\mathfrak{S}}$ as a finite MC.

# Positional versus fm-policies

## Positional policies are insufficient for $\omega$-regular properties

Consider the MDP:



Positional policy $\mathfrak{S}_\alpha$ always chooses $\alpha$ in state $s_0$
Positional policy $\mathfrak{S}_\beta$ always chooses $\beta$ in state $s_0$. Then:

$$Pr_{\mathfrak{S}_\alpha}(s_0 \models \Diamond a \wedge \Diamond b) \; = \; Pr_{\mathfrak{S}_\beta}(s_0 \models \Diamond a \wedge \Diamond b) \; = \; 0.$$

Now consider fm-policy $\mathfrak{S}_{\alpha\beta}$ which alternates between selecting $\alpha$ and $\beta$. Then: $Pr_{\mathfrak{S}_{\alpha\beta}}(s_0 \models \Diamond a \wedge \Diamond b) \; = \; 1$.

Thus, the class of positional policies is insufficiently powerful to characterise minimal (or maximal) probabilities for $\omega$-regular properties.

# Overview

1. Markov Decision Processes
2. Probabilities in MDPs
3. Policies
   - Positional policies
   - Finite-memory policies
4. Reachability probabilities
   - Mathematical characterisation
   - Value iteration
   - Linear programming
5. Summary

# Reachability probabilities

## Reachability probabilities

Let $\mathcal{M}$ be an MDP with state space $S$ and $\mathfrak{S}$ be a policy on $\mathcal{M}$. The reachability probability of $G \subseteq S$ from state $s \in S$ under policy $\mathfrak{S}$ is:

$$Pr^{\mathfrak{S}}(s \models \Diamond G) \;\; = \;\; Pr_s^{\mathcal{M}_{\mathfrak{S}}}\{\, \pi \in Paths(s) \mid \pi \models \Diamond G \,\}$$

## Maximal and minimal reachability probabilities

The minimal reachability probability of $G \subseteq S$ from $s \in S$ is:

$$Pr^{\text{min}}(s \models \Diamond B) = \inf_{\mathfrak{S}} Pr^{\mathfrak{S}}(s \models \Diamond B)$$

In a similar way, the maximal reachability probability of $G \subseteq S$ is:

$$Pr^{\text{max}}(s \models \Diamond B) \;\; = \;\; \sup_{\mathfrak{S}} Pr^{\mathfrak{S}}(s \models \Diamond B).$$

where policy $\mathfrak{S}$ ranges over all, infinitely (countably) many, policies.

# Example

# Maximal reachability probabilities

## MInimal guarantees for safety properties

Reasoning about the maximal probabilities for $\Diamond G$ is needed, e.g., for showing that $Pr^{\mathfrak{S}}(s \models \Diamond G) \leqslant \varepsilon$ for all policies $\mathfrak{S}$ and some small upper bound $0 < \varepsilon \leqslant 1$. Then:

$$Pr^{\mathfrak{S}}(s \models \Box \neg G) \;\geqslant\; 1 - \varepsilon \quad \text{for all policies } \mathfrak{S}.$$

The task to compute $Pr^{\max}(s \models \Diamond G)$ can thus be understood as showing that a safety property (namely $\Box \neg G$) holds with sufficiently large probability, viz. $1 - \varepsilon$, regardless of the resolution of nondeterminism.

# Equation system for max-reach probabilities

## Equation system for max-reach probabilities

Let $\mathcal{M}$ be a finite MDP with state space $S$, $s \in S$ and $G \subseteq S$. The vector $(x_s)_{s \in S}$ with $x_s = Pr^{\max}(s \models \Diamond G)$ yields the unique solution of the following equation system:

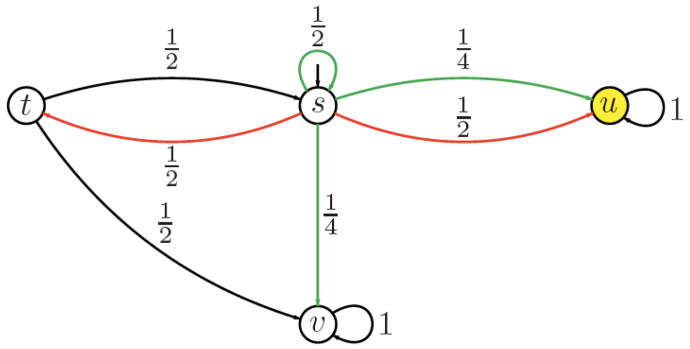
- If $s \in G$, then $x_s = 1$.
- If $s \notin Pre^*(G) \setminus G$, then $x_s = 0$.
- If $s \in Pre^*(G) \setminus G$

$$x_s \;=\; \max \left\{ \sum_{t \in S} \mathbf{P}(s, \alpha, t) \cdot x_t \;\middle|\; \alpha \in Act(s) \right\}$$

such that $\sum_{s \in S} x_s$ is minimal.

This is an instance of the Bellman equation for dynamic programming.

# Example



equation system for reachability objective $\Diamond\{\, u \,\}$ is:

$$x_u = 1 \text{ and } x_v = 0$$

$$x_s \;=\; \max\{\, \tfrac{1}{2}x_s + \tfrac{1}{4}x_u + \tfrac{1}{4}x_v, \tfrac{1}{2}x_u + \tfrac{1}{2}x_t \,\} \quad \text{and} \quad x_t = \tfrac{1}{2}x_s + \tfrac{1}{2}x_v$$

## Value iteration

The previous theorem suggests to calculate the values

$$x_s \;=\; Pr^{\max}(s \models \Diamond G)$$

by successive approximation.
For the states $s \in Pre^*(G) \setminus G$ we have $x_s \;=\; \lim_{n\to\infty} x_s^{(n)}$ where

$$x_s^{(0)} = 0 \quad \text{and} \quad x_s^{(n+1)} \;=\; \max\Big\{ \sum_{t\in S} \mathbf{P}(s,\alpha,t)\cdot x_t^{(n)} \;\Big|\; \alpha \in Act(s) \Big\}.$$

Note that $x_s^{(0)} \leqslant x_s^{(1)} \leqslant x_s^{(2)} \leqslant \ldots$. Thus, the values $Pr^{\max}(s \models \Diamond G)$ can be approximated by successively computing the vectors

$$\big(\, x_s^{(0)} \,\big),\ \big(\, x_s^{(1)} \,\big),\ \big(\, x_s^{(2)} \,\big),\ \ldots,$$

until $\max_{s\in S} |x_s^{(n+1)} - x_s^{(n)}|$ is below a certain (typically very small) threshold.

## Positional policies for max-reach probabilities

### Existence of optimal positional policies

Let $\mathcal{M}$ be a finite MDP with state space $S$, and $G \subseteq S$. There exists a positional policy $\mathfrak{S}$ such that for any $s \in S$ it holds:

$$Pr^{\mathfrak{S}}(s \models \Diamond G) \;=\; Pr^{\max}(s \models \Diamond G).$$

### Proof:

On the blackboard.

## Equation system for min-reach probabilities

### Equation system for min-reach probabilities

Let $\mathcal{M}$ be a finite MDP with state space $S$, $s \in S$ and $G \subseteq S$. The vector $(x_s)_{s\in S}$ with $x_s = Pr^{\min}(s \models \Diamond G)$ yields the unique solution of the following equation system:

- If $s \in G$, then $x_s = 1$.
- If $Pr^{\min}(s \models G) = 0$, then $x_s = 0$.
- If $Pr^{\min}(s \models G) > 0$ and $s \notin G$, then

$$x_s \;=\; \min\Big\{ \sum_{t\in S} \mathbf{P}(s,\alpha,t)\cdot x_t \;\Big|\; \alpha \in Act(s) \Big\}$$

such that $\sum_{s\in S} x_s$ is maximal.

## Preprocessing

The preprocessing required to compute the set

$$S^{\min}_{=0} \;=\; \{\, s \in S \mid Pr^{\min}(s \models \Diamond G) \,\} = 0$$

can be performed by graph algorithms. The set $S^{\min}_{=0}$ is given by $S \setminus T$ where

$$T = \bigcup_{n \geqslant 0} T_n$$

and $T_0 = G$ and, for $n \geqslant 0$:

$$T_{n+1} \;=\; T_n \cup \{\, s \in S \mid \forall \alpha \in Act(s)\, \exists t \in T_n.\, \mathbf{P}(s,\alpha,t) > 0 \,\}.$$

As $T_0 \subseteq T_1 \subseteq T_2 \subseteq \ldots \subseteq S$ and $S$ is finite, the sequence $(T_n)_{n\geqslant 0}$ eventually stabilizes, i.e., for some $n \geqslant 0$, $T_n = T_{n+1} = \ldots = T$.

Then: $Pr^{\min}(s \models \Diamond G) > 0$ if and only if $s \in T$.

## Preprocessing

**Algorithm 46** Computing the set of states $s$ with $Pr^{\min}(s \models \Diamond B) = 0$

*Input:* finite MDP $\mathcal{M}$ with state space $S$ and $B \subseteq S$
*Output:* $\{ s \in S \mid Pr^{\min}(s \models \Diamond B) = 0 \}$

```
$T := B$;
$R := B$;
while $R \neq \varnothing$ do
    let $t \in R$;
    $R := R \setminus \{ t \}$;
    for all $(s, \alpha) \in Pre(t)$ with $s \notin T$ do
        remove $\alpha$ from $Act(s)$
        if $Act(s) = \varnothing$ then
            add $s$ to $R$ and $T$
        fi
    od
od
return $T$
```

## Positional policies for min-reach probabilities

**Existence of optimal positional policies**

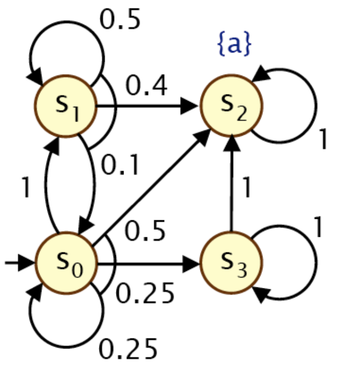
Let $\mathcal{M}$ be a finite MDP with state space $S$, and $G \subseteq S$. There exists a positional policy $\mathfrak{S}$ such that for any $s \in S$ it holds:

$$Pr^{\mathfrak{S}}(s \models \Diamond G) \;=\; Pr^{\min}(s \models \Diamond G).$$

**Proof:**

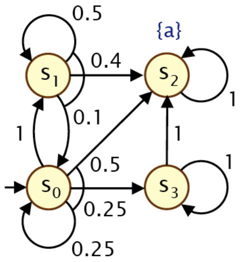
Similar to the case for maximal reachability probabilities.

## Example value iteration



Determine $Pr^{\min}(s_i \models \Diamond\{ s_2 \})$.

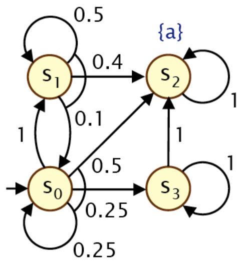
## Example value iteration



Determine $Pr^{\min}(s_i \models \Diamond\{ s_2 \})$

1. $G = \{ s_2 \}$, $Pre*(G) \setminus G = \{ s_0, s_1 \}$.
2. $(x_s^{(0)}) = (0, 0, 1, 0)$
3. $(x_s^{(1)}) = (\min(1{\cdot}0, 0.25{\cdot}0{+}0.25{\cdot}0{+}0.5{\cdot}1),$ $0.1{\cdot}0{+}0.5{\cdot}0{+}0.4{\cdot}1, 1, 0)$
4. $= (0, 0.4, 1, 0)$
5. $(x_s^{(2)}) = (\min(1{\cdot}0.4, 0.25{\cdot}0{+}0.25{\cdot}0{+}0.5{\cdot}1),$ $0.1{\cdot}0{+}0.5{\cdot}0.4{+}0.4{\cdot}1, 1, 0)$
6. $= (0.4, 0.6, 1.0)$
7. $(x_s^{(3)}) = \ldots\ldots$

## Example value iteration

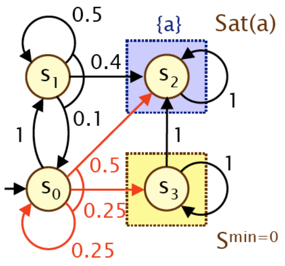


Determine

$Pr^{\min}(s_i \models \Diamond\{ s_2 \})$

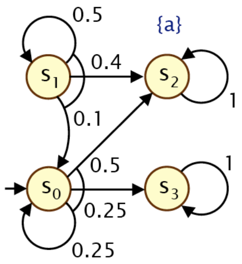
| | $[\, x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)} \,]$ |
|---|---|
| n=0: | [ 0.000000, 0.000000, 1, 0 ] |
| n=1: | [ 0.000000, 0.400000, 1, 0 ] |
| n=2: | [ 0.400000, 0.600000, 1, 0 ] |
| n=3: | [ 0.600000, 0.740000, 1, 0 ] |
| n=4: | [ 0.650000, 0.830000, 1, 0 ] |
| n=5: | [ 0.662500, 0.880000, 1, 0 ] |
| n=6: | [ 0.665625, 0.906250, 1, 0 ] |
| n=7: | [ 0.666406, 0.919688, 1, 0 ] |
| n=8: | [ 0.666602, 0.926484, 1, 0 ] |
| ··· | |
| n=20: | [ 0.666667, 0.933332, 1, 0 ] |
| n=21: | [ 0.666667, 0.933332, 1, 0 ] |
| | ≈ [ 2/3, 14/15, 1, 0 ] |

## Optimal positional policy



- Outcome of the value iteration $(\, x_s \,) = (\frac{14}{15}, \frac{2}{3}, 1, 0)$
- How to obtain the optimal policy from this result?
- $x_{s_0} = \min(1 \cdot \frac{14}{15}, 0.5 \cdot 1 + 0.5 \cdot 0 + 0.25 \cdot \frac{2}{3})$
  $\min(\frac{14}{15}, \frac{2}{3})$
- Thus the optimal policy always selects red.

## Induced DTMC



- Outcome of the value iteration $(\, x_s \,) = (\frac{14}{15}, \frac{2}{3}, 1, 0)$
- How to obtain the optimal policy from this reults?
- $x_{s_0} = \min(1 \cdot \frac{14}{15}, 0.5 \cdot 1 + 0.5 \cdot 0 + 0.25 \cdot \frac{2}{3})$
  $\min(\frac{14}{15}, \frac{2}{3})$
- Thus the optimal policy always selects red.

An alternative to value iteration is linear programming.

## Linear programming

### Linear programming

Let $x_1, \ldots, x_n$ be real-valued variables. Maximise (or minimise) the objective function:

$$c_1 \cdot x_1 + c_2 \cdot x_2 + \ldots + c_n \cdot x_n \quad \text{for constants} c_1, \ldots, c_n \in \mathbb{R}$$

subject to the constraints

$$\begin{aligned} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1n} \cdot x_n &\leqslant b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \ldots + a_{2n} \cdot x_n &\leqslant b_2 \\ \ldots\ldots a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n &\leqslant b_m. \end{aligned}$$

Solution techniques: e.g., Simplex, ellipsoid method, interior point method.

### Linear programming

Optimisation of a linear objective function subject to linear (in)equality

# Maximal reach probabilities as a linear program

## Linear program for max-reach probabilities

Consider a finite MDP with state space $S$, and $G \subseteq S$. The values $x_s = Pr^{\max}(s \models \Diamond G)$ are the unique solution of the *linear program*:

- If $s \in G$, then $x_s = 1$.
- If $s \notin Pre^*(G) \setminus G$, then $x_s = 0$.
- If $s \in Pre^*(G) \setminus G$ then $0 \leqslant x_s \leqslant 1$ and for all actions $\alpha \in Act(s)$:

$$x_s \geqslant \sum_{t \in S} \mathbf{P}(s, \alpha, t) \cdot x_t$$

where $\sum_{s \in S} x_s$ is minimal.

## Proof:

See lecture notes.

# Minimal reach probabilities as a linear program

## Linear program for min-reach probabilities

Consider a finite MDP with state space $S$, and $G \subseteq S$. The values $x_s = Pr^{\min}(s \models \Diamond G)$ are the unique solution of the *linear program*:

- If $s \in G$, then $x_s = 1$.
- If $Pr^{\min}(s \models \Diamond G) = 0$, then $x_s = 0$.
- If $Pr^{\min}(s \models \Diamond G) > 0$ and $s \notin G$ then $0 \leqslant x_s \leqslant 1$ and for all actions $\alpha \in Act(s)$:

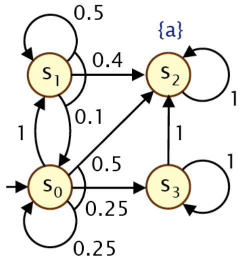
$$x_s \leqslant \sum_{t \in S} \mathbf{P}(s, \alpha, t) \cdot x_t$$

where $\sum_{s \in S} x_s$ is maximal.

## Proof:

See lecture notes.

# Example linear programming

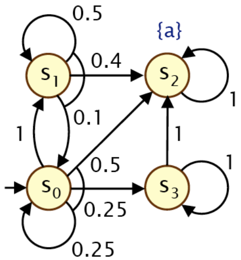


Determine $Pr^{\min}(s_i \models \Diamond\{ s_2 \})$

- $G = \{ s_2 \}$, $Pre*(G) \setminus G = \{ s_0, s_1 \}$.
- Maximise $x_0 + x_1$ subject to the constraints:

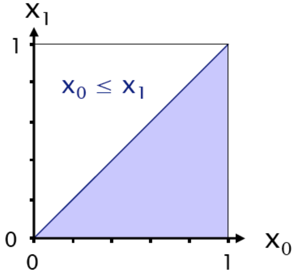
$$\begin{aligned} x_0 &\leqslant x_1 \\ x_0 &\leqslant \tfrac{1}{4}\cdot x_0 + \tfrac{1}{2} \\ x_1 &\leqslant \tfrac{1}{10}\cdot x_0 + \tfrac{1}{2}\cdot x_1 + \tfrac{2}{5} \end{aligned}$$

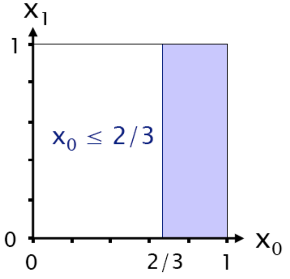
# Example linear programming

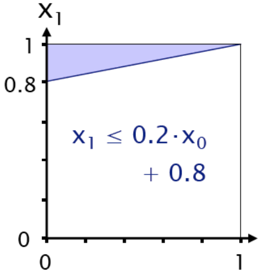


- $G = \{ s_2 \}$, $Pre*(G) \setminus G = \{ s_0, s_1 \}$.
- Maximise $x_0 + x_1$ subject to the constraints:

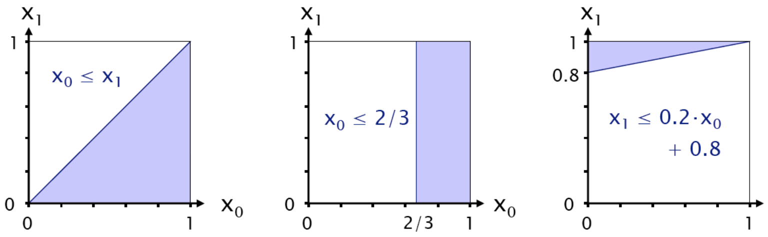
$$\begin{aligned} x_0 &\leqslant x_1 \\ x_0 &\leqslant \tfrac{2}{3} \\ x_1 &\leqslant \tfrac{2}{5}\cdot x_0 + \tfrac{4}{5} \end{aligned}$$

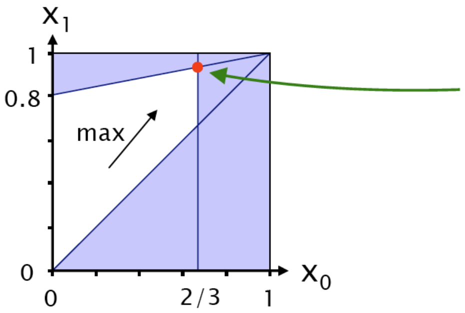






## Example linear programming





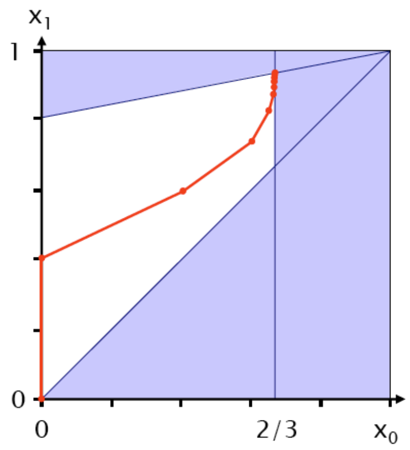
Solution:

$(x_0, x_1)$

$=$

$(2/3, 14/15)$

## Example linear programming



| | $[\,x_0^{(n)}, x_1^{(n)}, x_2^{(n)}, x_3^{(n)}\,]$ |
|---|---|
| n=0: | [ 0.000000, 0.000000, 1, 0 ] |
| n=1: | [ 0.000000, 0.400000, 1, 0 ] |
| n=2: | [ 0.400000, 0.600000, 1, 0 ] |
| n=3: | [ 0.600000, 0.740000, 1, 0 ] |
| n=4: | [ 0.650000, 0.830000, 1, 0 ] |
| n=5: | [ 0.662500, 0.880000, 1, 0 ] |
| n=6: | [ 0.665625, 0.906250, 1, 0 ] |
| n=7: | [ 0.666406, 0.919688, 1, 0 ] |
| n=8: | [ 0.666602, 0.926484, 1, 0 ] |
| ... | |
| n=20: | [ 0.666667, 0.933332, 1, 0 ] |
| n=21: | [ 0.666667, 0.933332, 1, 0 ] |
| | ≈ [ 2/3, 14/15, 1, 0 ] |

## Time complexity

### Time complexity

For finite MDP $\mathcal{M}$ with state space $S$, $G \subseteq S$ and $s \in S$, the values $Pr^{\max}(s \models \Diamond G)$ can be computed in time polynomial in the size of $\mathcal{M}$. The same holds for $Pr^{\min}(s \models \Diamond G)$.

### Proof:

Thanks to the characterisation as a linear program and polynomial time techniques to solve such linear programs such as ellipsoid methods.

### Corollary

For finite MDPs, the question whether $Pr^{\mathfrak{S}}(s \models \Diamond G) \leqslant p$ for some rational $p \in [0, 1[$ is decidable in polynomial time.

## Overview

1. Markov Decision Processes
2. Probabilities in MDPs
3. Policies
   - Positional policies
   - Finite-memory policies
4. Reachability probabilities
   - Mathematical characterisation
   - Value iteration
   - Linear programming
5. Summary

# Summary

## Important points

1. Maximal reachability probabilities are suprema over reachability probabilities for all, potentially infinitely many, policies.
2. They are characterised by equation systems with maximal operators.
3. There exists a positional policy that yields the maximal reachability probability.
4. Such policies can bet determined using value iteration.
5. Or, alternatively, in polynomial time using linear programming.
6. Positional policies are not powerful enough for arbitrary $\omega$-regular properties.