Modeling and Verification of Probabilistic Systems

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/teaching/ss-14/movep14/

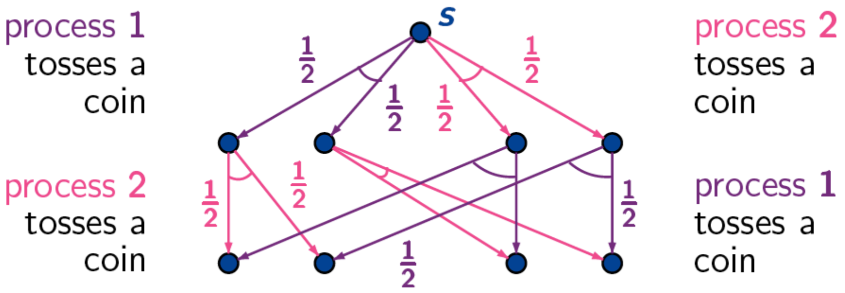
May 20, 2014

Overview

1. Nondeterminism
2. Markov Decision Processes
3. Probabilities in MDPs
4. Policies
5. Summary

Randomness and concurrency

Markov chains are not appropriate for modeling randomized distributed systems, since they cannot adequately model the interleaving behavior of the concurrent processes.



process 1 tosses a coin

process 2 tosses a coin

process 2 tosses a coin

process 1 tosses a coin

Nondeterminism

The use of nondeterminism

- Concurrency – scheduling of parallel components
  - in randomised distributed algorithms, several components run partly autonomously and interact asynchronously
- Abstraction
  - partition state space of a DTMC in similar (but not bisimilar) states
  - replace probabilistic branching by a nondeterministic choice
- Unknown environments
  - interaction with unknown environment
  - example: security in which the environment is an unknown adversary

Beware

Nondeterminism is not the same as a uniform distribution!

Overview

1. Nondeterminism
2. Markov Decision Processes
3. Probabilities in MDPs
4. Policies
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)

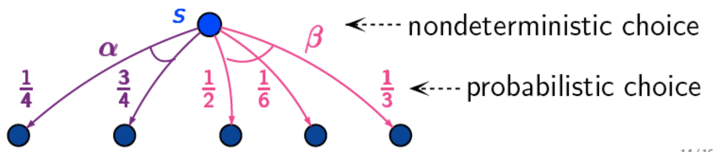
Markov decision process

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.

Markov decision process (MDP)

Markov decision process

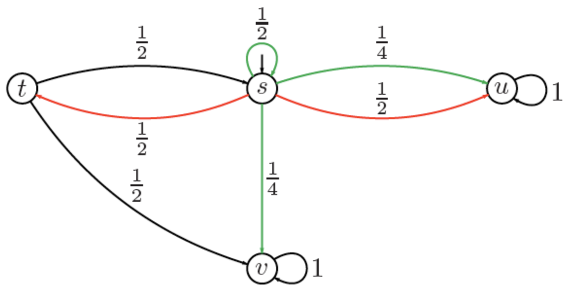
An MDP $\mathcal{M}$ is a tuple $(S, Act, \mathbf{P}, \iota_{\text{init}}, AP, L)$ where

- 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:

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

If $|Act(s)| = 1$ for any state s, then the nondeterministic choice in any state is over a singleton set. In this case, $\mathcal{M}$ is a DTMC. Vice versa, a DTMC is an MDP such that $|Act(s)| = 1$ for all 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

Example: randomized mutual exclusion

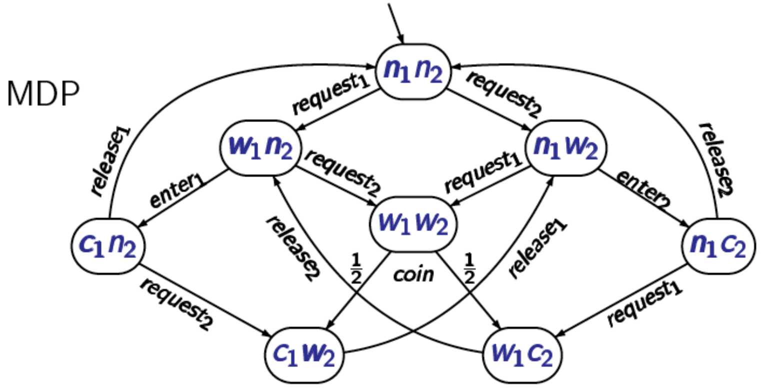
- **2** concurrent processes $\mathcal{P}_1$, $\mathcal{P}_2$ with **3** phases:

n_i	noncritical actions of process $\mathcal{P}_i$
w_i	waiting phase of process $\mathcal{P}_i$
c_i	critical section of process $\mathcal{P}_i$

- competition of both processes are waiting
- resolved by a randomized arbiter who tosses a coin

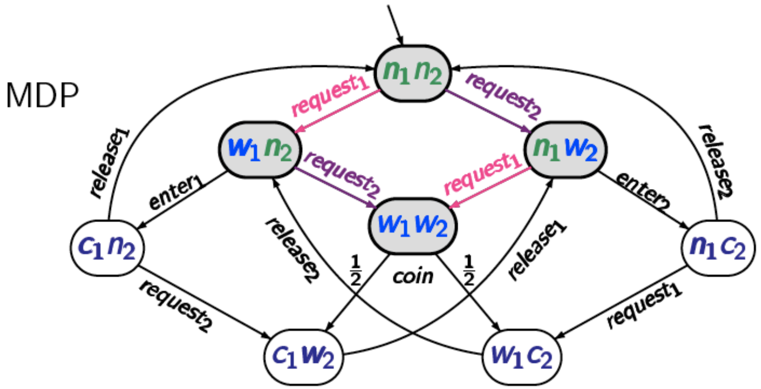
Randomized mutual exclusion

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting



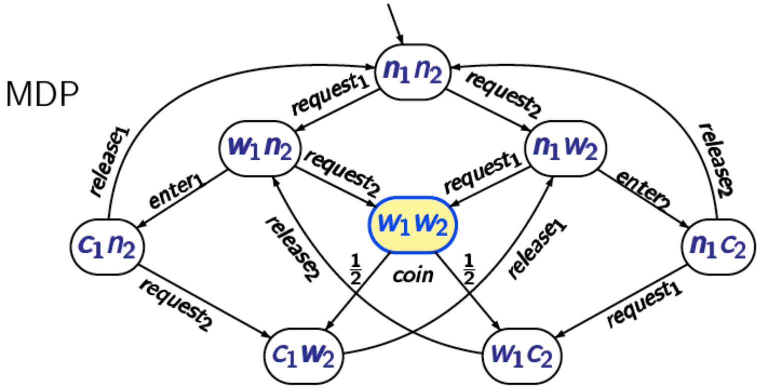
Randomized mutual exclusion

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting



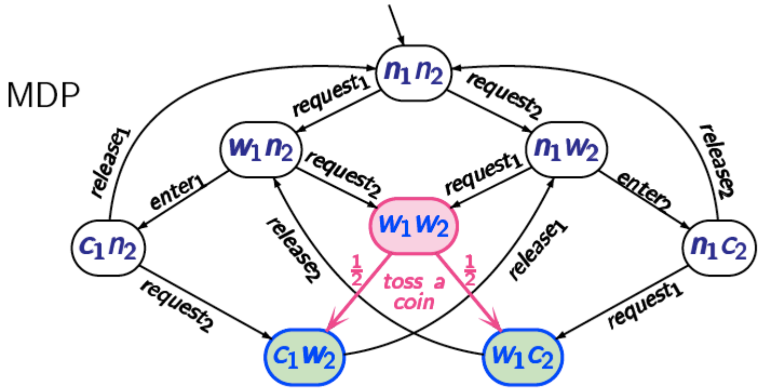
Randomized mutual exclusion

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting



Randomized mutual exclusion

- interleaving of the request operations
- competition if both processes are waiting
- randomized arbiter tosses a coin if both are waiting



Intuitive operational behavior

Intuitive operational MDP behavior

1. A stochastic experiment according to ι_{init} yields starting state s_0 with probability $\iota_{\text{init}}(s_0) > 0$.
2. On entering state s, a non-deterministic choice among $Act(s)$ determines the next action $\alpha \in Act(s)$, say.
3. The next state t is randomly chosen with probability $\mathbf{P}(s, \alpha, t)$.
4. If t is the unique α-successor of s, then almost surely t is the successor after selecting α, i.e., $\mathbf{P}(s, \alpha, t) = 1$.
5. Continue with step 2.

Overview

1 Nondeterminism

2 Markov Decision Processes

3 Probabilities in MDPs

4 Policies

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 $\mathcal{M}$, and $(s, s') \in E$ iff $\mathbf{P}(s, \alpha, s') > 0$ for some $\alpha \in Act$.

Paths

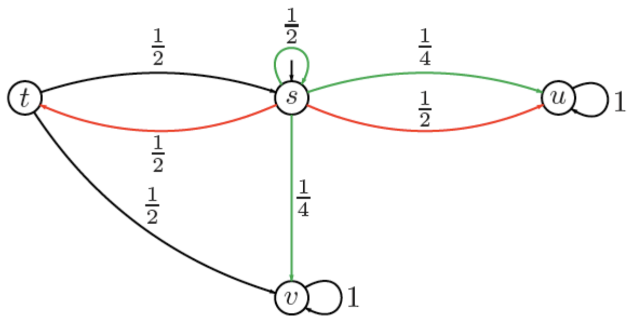
An infinite *path* in an MDP $\mathcal{M} = (S, Act, \mathbf{P}, \iota_{\text{init}}, AP, L)$ is an infinite sequence $s_0\, \alpha_1\, s_1\, \alpha_2\, s_2\, \alpha_3 \ldots \in (S \times Act)^\omega$, written as

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

such that $\mathbf{P}(s_i, \alpha_{i+1}, s_{i+1}) > 0$ for all $i \geqslant 0$. Any finite prefix of π 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.

Paths in MDPs



$$s \xrightarrow{\alpha} s \xrightarrow{\alpha} s \xrightarrow{\beta} t \xrightarrow{\alpha} s \xrightarrow{\beta} u \ldots$$

$$s \xrightarrow{\beta} t \xrightarrow{\alpha} s \xrightarrow{\beta} t \xrightarrow{\alpha} s \ldots\ldots$$

Probabilities in MDPs

- For DTMCs, a set of infinite paths is equipped with a σ-algebra and a probability measure that reflects the intuitive notion of probabilities for paths.
- Due to the presence of nondeterminism, MDPs are not augmented with a unique probability measure.
- Example: suppose we have two coins: a fair one, and a biased one, say $\frac{1}{6}$ for heads and $\frac{5}{6}$ for tails. We select nondeterministically one of the coins, and are interested in the probability of obtaining tails. This, however, is not specified! This also applies if we select one of the two coins repeatedly.
- Reasoning about probabilities of sets of paths of an MDP relies on the resolution of nondeterminism. This resolution is performed by a policy.[1] A policy chooses in any state s one of the actions $\alpha \in Act(s)$.

[1] Also called scheduler, strategy or adversary.

Overview

1. Nondeterminism
2. Markov Decision Processes
3. Probabilities in MDPs
4. Policies
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 $s_0\, s_1 \ldots s_n \in S^+$.

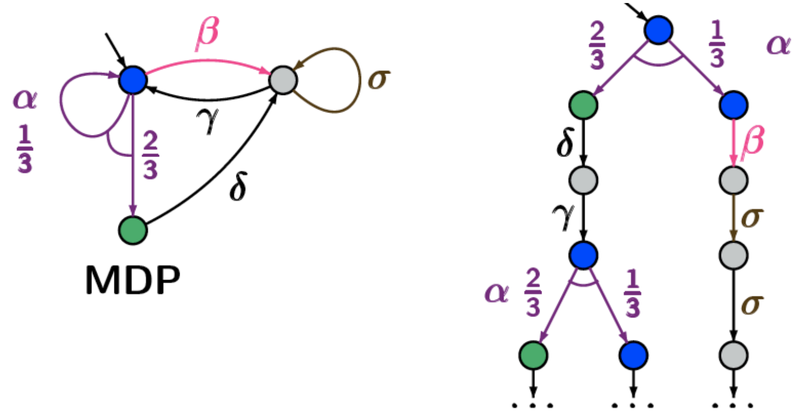
The path

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

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

For any scheduler, 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 α_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 Markov chain



Each policy induces an infinite DTMC. States are finite prefixes of paths in the MDP.

Induced DTMC of an MDP by a policy

DTMC of an MDP induced by a policy

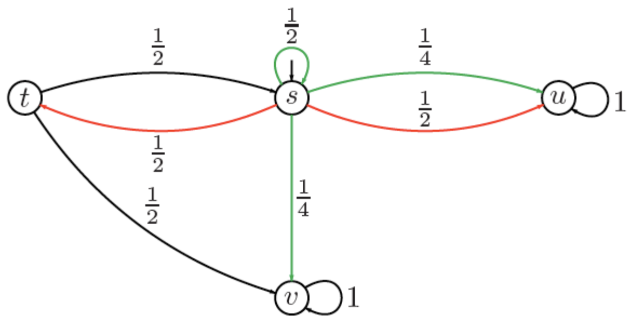
Let $\mathcal{M} = (S, Act, \mathbf{P}, \iota_{\mathrm{init}}, AP, L)$ be an MDP and $\mathfrak{S}$ a policy on $\mathcal{M}$. The DTMC induced by $\mathfrak{S}$, denoted $\mathcal{M}_{\mathfrak{S}}$, is given by

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

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 $L'(\sigma) \;=\; L(s_n)$.

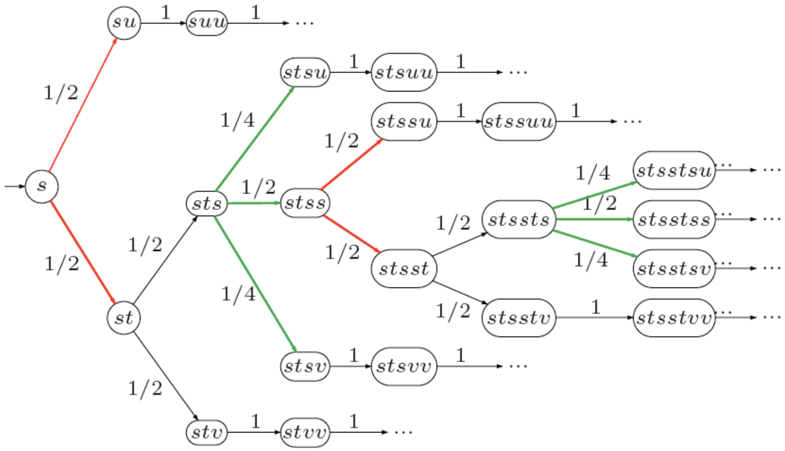
$\mathcal{M}_{\mathfrak{S}}$ is infinite, even if the MDP $\mathcal{M}$ is finite. Intuitively, state $s_0\, s_1 \ldots s_n$ of DTMC $\mathcal{M}_{\mathfrak{S}}$ represents the configuration where the MDP $\mathcal{M}$ is in state s_n and $s_0\, s_1 \ldots s_{n-1}$ stands for the history. Since policy $\mathfrak{S}$ 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.

MDP paths versus paths in the induced DTMC

There is a one-to-one correspondence between the $\mathfrak{S}$-paths of the MDP $\mathcal{M}$ and the paths in the Markov chain $\mathcal{M}_{\mathfrak{S}}$.

For $\mathfrak{S}$-path $\pi \; = \; s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots$, the corresponding path in DTMC $\mathcal{M}_{\mathfrak{S}}$ is:

$$\pi^{\mathfrak{S}} \; = \; \widehat{\pi}_0 \, \widehat{\pi}_1 \, \widehat{\pi}_2 \, \ldots \quad \text{where} \quad \widehat{\pi}_n \; = \; s_0 \, s_1 \ldots s_n.$$

Vice versa, for a path $\widehat{\pi}_0 \, \widehat{\pi}_1 \, \widehat{\pi}_2 \, \ldots$ in the DTMC $\mathcal{M}_{\mathfrak{S}}$, $\widehat{\pi}_0 = s_0$ for some state s_0 such that $\iota_{\text{init}}(s_0) > 0$ and, for each $n > 0$, $\widehat{\pi}_n \; = \; \widehat{\pi}_{n-1} \, s_n$ for some state s_n in the MDP $\mathcal{M}$ such that $\mathbf{P}(s_{n-1}, \mathfrak{S}(\widehat{\pi}_{n-1}), s_n) > 0$. Hence:

$$s_0 \xrightarrow{\mathfrak{S}(\widehat{\pi}_0)} s_1 \xrightarrow{\mathfrak{S}(\widehat{\pi}_1)} s_2 \xrightarrow{\mathfrak{S}(\widehat{\pi}_2)} \ldots$$

is a $\mathfrak{S}$-path in $\mathcal{M}$.

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 ω-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,

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

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.

Overview

Summary

Important points

1. An MDP is a model exhibiting non-determinism and probabilities.
2. Non-determinism is important for e.g., randomized distributed algorithms.
3. Policies are functions that select enabled actions in states.
4. A policy on an MDP induces an infinite DTMC, even if the MDP is finite.
5. Probability measures on MDP paths are defined using induced DTMC paths.
6. A positional policy selects in a state always the same action.