



Chapter 1

Probabilistic Systems

Course “Specification and Verification of Parallel Systems”

Martin Steffen, Paul Kobialka, Tobias John

Autumn 2025



Section

Markov Decision Processes

Autumn 2025

Motivation

- Markov Chains offer a way to model *probabilistic processes*
 - Non-deterministic transition systems offer a way to model *choices* of an agent
 - Goal: Have a model that combines both
- ⇒ We use **Markov Decision Processes (MDPs)**



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Motivation

- Markov Chains offer a way to model *probabilistic processes*
 - Non-deterministic transition systems offer a way to model *choices* of an agent
 - Goal: Have a model that combines both
- ⇒ We use **Markov Decision Processes (MDPs)**
- They allow to model a system with choices where the outcomes are probabilistic, i.e. the consequence of a choice is not always the same



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Motivation

- Markov Chains offer a way to model *probabilistic processes*
- Non-deterministic transition systems offer a way to model *choices* of an agent
- Goal: Have a model that combines both

⇒ We use **Markov Decision Processes (MDPs)**

- They allow to model a system with choices where the outcomes are probabilistic, i.e. the consequence of a choice is not always the same
- We can use such models to find *optimal strategies* (later called *schedulers*) that maximize the probability that the system behaves as desired
- E.g. from our running example: How should we treat customers in the ice-cream shop in order to increase the probability of making them happy?



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

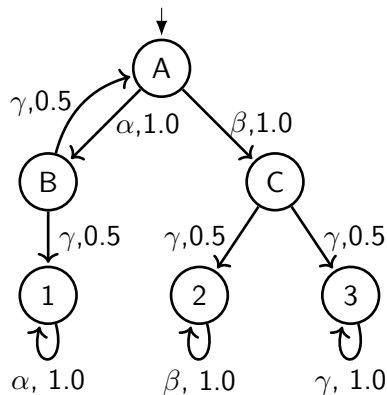
Markov Decision Processes (MDPs): Intuition

A *Markov decision process* is a “Markov chain with non-determinism”, i.e., a Markov chain where each transition

- has an associated probability, and
- an associated *action label*, making it possible to choose between several actions in one state.

If there are transitions with different action labels exiting from one state, one needs to choose an action (i.e., transitions can be chosen non-deterministically).

In the state labeled with A , one can choose between action α and action β .



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Definition (Markov Decision Process)

A *Markov decision process* is a tuple $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ where

- S is a countable, nonempty set of *states*,
- Act is a set of *actions*,
- $P : S \times \text{Act} \times S \rightarrow [0, 1]$ is the *transition probability function* such that for all states $s \in S$ and all actions $\alpha \in \text{Act}$ it holds:

$$\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\},$$

- $s_0 \in S$ is the *initial state*,
- AP is the set of atomic propositions, and
- $L : S \rightarrow \mathcal{P}(\text{AP})$ is a labelling function.

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

MDPs: Transition Probability Function

For all states $s \in S$ and all actions $\alpha \in \text{Act}$ it holds:

$$\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\},$$

What does that mean?

- P maps all combinations of states and actions to 0, if the action can not be chosen in the source state.

For example, $P(s_0, \alpha, s_2) = 0$ in the previous example where s_0 has label A and s_2 has label C .

- If there are transitions exiting from state s with label α , then the sum of the probabilities of these transitions should be 1 (as in Markov chains).
For example, $P(s_2, \gamma, s_4) = P(s_2, \gamma, s_5) = 0.5$ in the previous example where s_4 has label 2 and s_5 has label 3.



IN5110 –
Specification and
Verification of
Parallel Systems

**Markov Decision
Processes**

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

MDPs: Example formally defined

$\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$, where

$S = \{s_0, \dots, s_5\}$,

$\text{Act} = \{\alpha, \beta, \gamma\}$,

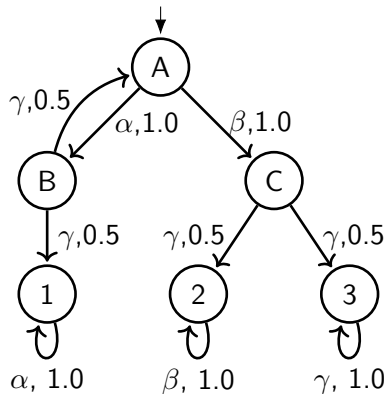
$$P(s, a, t) = \begin{cases} 1 & (s, a, t) \in \\ & \{(s_0, \alpha, s_1), (s_0, \beta, s_2)\} \\ 0.5 & (s, a, t) \in \\ & \{(s_1, \gamma, s_0), (s_1, \gamma, s_3), \\ & (s_2, \gamma, s_4), (s_2, \gamma, s_5)\} \\ 0 & \text{otherwise} \end{cases}$$

$\text{AP} = \{A, B, C, 1, 2, 3\}$,

$L : S \rightarrow \mathcal{P}(\text{AP})$ s.t.

$s_0 \mapsto \{A\}, s_1 \mapsto \{B\}, s_2 \mapsto \{C\}$,

$s_3 \mapsto \{1\}, s_4 \mapsto \{2\}, s_5 \mapsto \{3\}$.



IN5110 –
Specification and
Verification of
Parallel Systems

**Markov Decision
Processes**

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Example: MDPs in the (Idea)

We have a list of (rather arbitrary) things that we want to model:

- Actions = actions of employee in 
- There are two types of ice cream (green and red)
- Customers order both types equally likely
- Green orders can be rejected or prepared
- Red orders can be prepared or green ice cream is prepared instead
- Customers might be happy or sad after serving
- Customers do not like getting the wrong type of ice cream
- Most customers enjoy green ice cream (after ordering it)
- Only half of the customers enjoy red ice cream



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Example: MDPs in the (Model)

- $\text{Act} = \{\text{to}, \text{co}, \text{r}, \text{g}, \text{s}, \text{n}\}$ are actions for serving ice cream

to: take order

r: prepare red

s: serve ice cream

co: cancel order

g: prepare green

n: do nothing



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Example: MDPs in the (Model)

- Act = {to, co, r, g, s, n} are actions for serving ice cream

to: take order

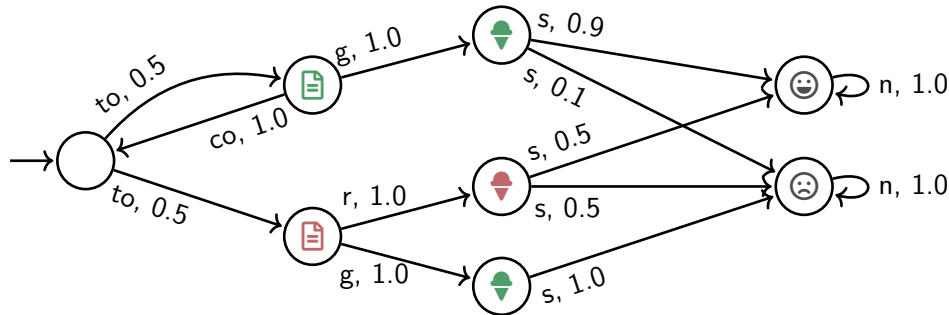
r: prepare red

s: serve ice cream

co: cancel order

g: prepare green

n: do nothing



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Markov Decision Processes

Definition of MDPs

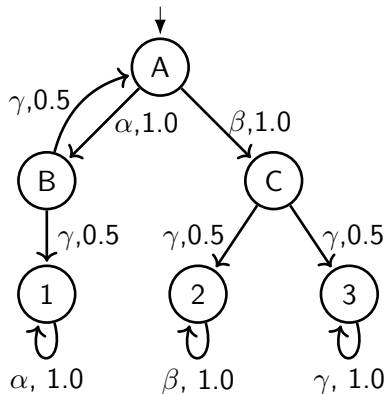
Schedulers

Optimal Schedulers

Costs

Computation

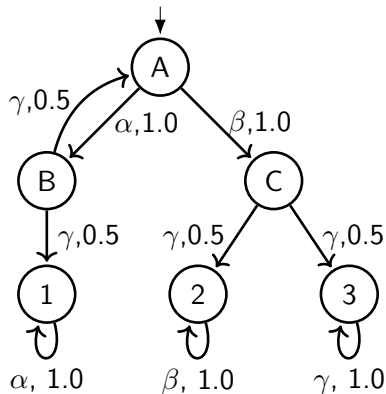
How to find the probability of reaching, e.g., state 2?



Schedulers: Intuition

How to find the probability of reaching, e.g., state 2?

⇒ Depends on the resolution of non-determinism in state A .



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

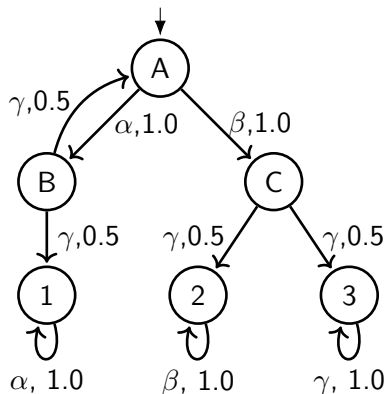
How to find the probability of reaching, e.g., state 2?

⇒ Depends on the resolution of non-determinism in state A .

⇒ Resolution is performed by a *scheduler*
A scheduler of an MDP

- chooses in every state s one enabled action, i.e., an “action with probability >0 ”, and
- does not impose constraints on the probabilistic choices (i.e., probabilities remain the same).

For example, a scheduler could choose α in A and γ in B and C .



Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Schedulers: Example in the



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

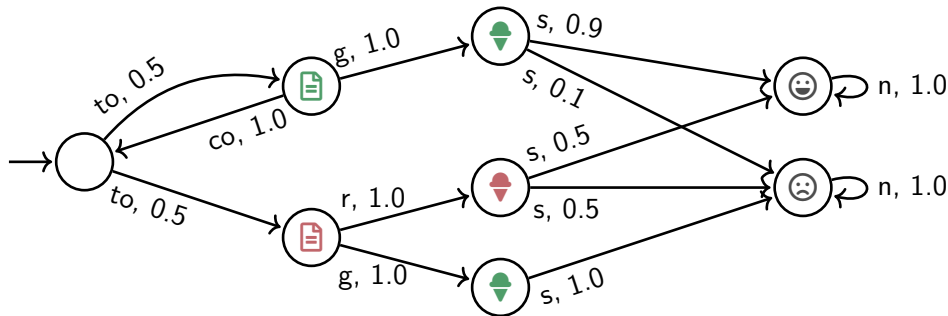
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



- Let's start by depicting the names of the states

Schedulers: Example in the



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

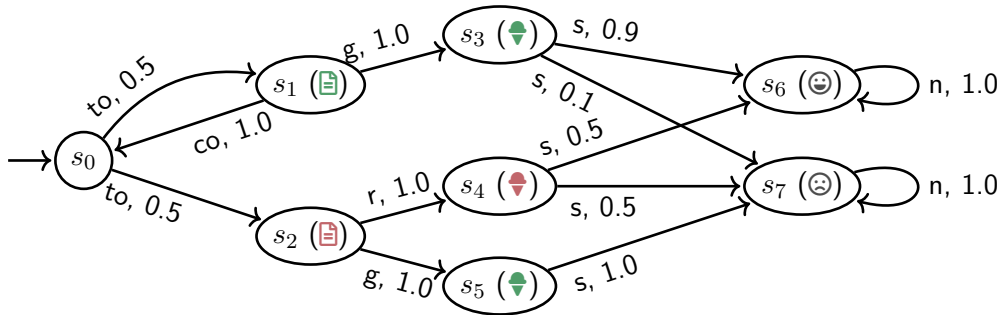
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



- Let's start by depicting the names of the states

Schedulers: Example in the



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

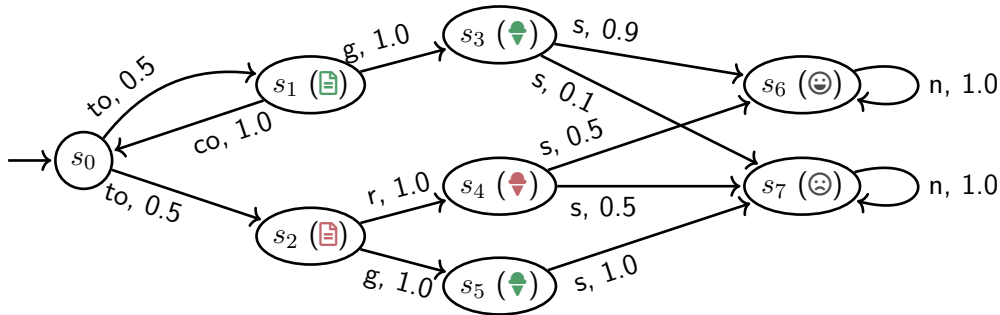
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



- Let's start by depicting the names of the states
- Only non-deterministic choices: in s_1 and s_2
- Scheduler that wants to make customer happy:

Schedulers: Example in the



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

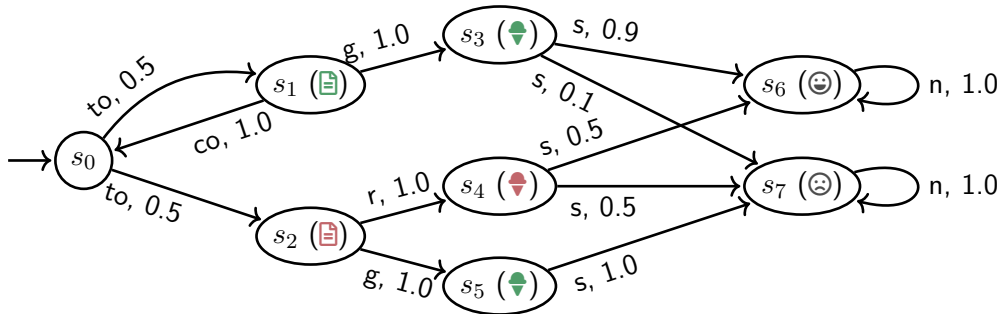
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



- Let's start by depicting the names of the states
- Only non-deterministic choices: in s_1 and s_2
- Scheduler that wants to make customer happy:
Choose "g" in state s_1
Choose "r" in state s_2



Definition

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP. An action $\alpha \in \text{Act}$ is *enabled* in a state $s \in S$ if and only if $\sum_{s' \in S} P(s, \alpha, s') = 1$, i.e., if there are outgoing transitions from s labeled with α and a probability greater than 0.

We denote by $\text{Act}(s)$ the set of enabled actions in a state $s \in S$. We require $\text{Act}(s) \neq \emptyset$ for every $s \in S$.

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Definition

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP. An action $\alpha \in \text{Act}$ is *enabled* in a state $s \in S$ if and only if $\sum_{s' \in S} P(s, \alpha, s') = 1$, i.e., if there are outgoing transitions from s labeled with α and a probability greater than 0.

We denote by $\text{Act}(s)$ the set of enabled actions in a state $s \in S$. We require $\text{Act}(s) \neq \emptyset$ for every $s \in S$.

Definition (Scheduler)

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP. A *memoryless scheduler* for $\mathcal{M}$ is a function $\mathfrak{S} : S \rightarrow \text{Act}$ such that $\mathfrak{S}(s) \in \text{Act}(s)$ for all $s \in S$.

Note: There exist more complex schedulers, but for our purposes, memoryless schedulers are sufficient.

Markov Decision Processes

Definition of MDPs

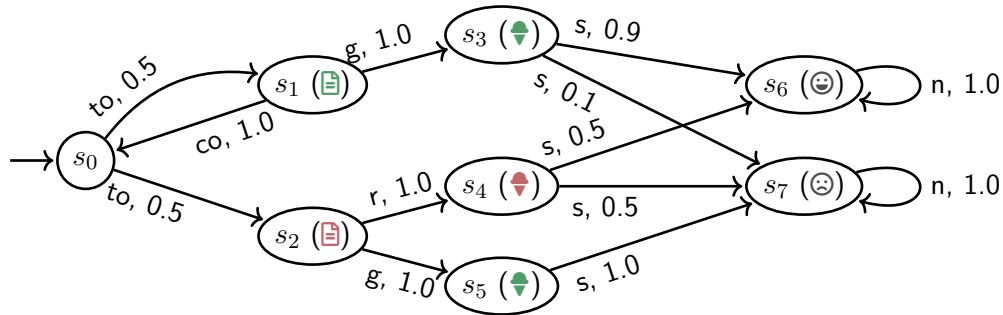
Schedulers

Optimal Schedulers

Costs

Computation

Schedulers: Example in an the



Informal description:

- Choose "g" in s_1 $\mathfrak{S}_1(s_0) = to$ $\mathfrak{S}_1(s_1) = g$ $\mathfrak{S}_1(s_2) = r$ $\mathfrak{S}_1(s_3) = s$
 Choose "r" in s_2 $\mathfrak{S}_1(s_4) = s$ $\mathfrak{S}_1(s_5) = s$ $\mathfrak{S}_1(s_6) = n$ $\mathfrak{S}_1(s_7) = n$



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Schedulers: Example in an the



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

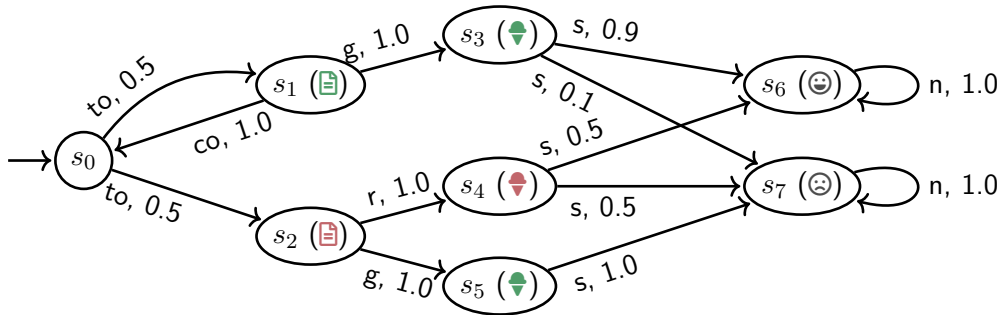
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



Informal description:

- Choose “g” in s_1 $\mathfrak{S}_1(s_0) = to$ $\mathfrak{S}_1(s_1) = g$ $\mathfrak{S}_1(s_2) = r$ $\mathfrak{S}_1(s_3) = s$
 Choose “r” in s_2 $\mathfrak{S}_1(s_4) = s$ $\mathfrak{S}_1(s_5) = s$ $\mathfrak{S}_1(s_6) = n$ $\mathfrak{S}_1(s_7) = n$
- Choose “co” in s_1 $\mathfrak{S}_2(s_0) = to$ $\mathfrak{S}_2(s_1) = co$ $\mathfrak{S}_2(s_2) = g$ $\mathfrak{S}_2(s_3) = s$
 Choose “g” in s_2 $\mathfrak{S}_2(s_4) = s$ $\mathfrak{S}_2(s_5) = s$ $\mathfrak{S}_2(s_6) = n$ $\mathfrak{S}_2(s_7) = n$

- Markov Chains and MDPs are closely related

Markov chain = MDP s.t. $\text{Act}(s)$ is a singleton set for every state s

(Check yourself that this is true with the definitions.)

⇒ Markov chains are a proper subset of MDPs

- Action names of MDP are irrelevant in this case and can be omitted.

Induced Markov Chains

- As seen on last slide: MDPs and Markov Chains are closely related
- Consequence: Given an MDP and a memoryless scheduler for it, we obtain a Markov Chain by selecting in each state the action that the scheduler suggests



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Induced Markov Chains

- As seen on last slide: MDPs and Markov Chains are closely related
- Consequence: Given an MDP and a memoryless scheduler for it, we obtain a Markov Chain by selecting in each state the action that the scheduler suggests

Definition

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP and $\mathfrak{S}$ a memoryless scheduler on $\mathcal{M}$. The *induced Markov Chain* $\mathcal{M}_{\mathfrak{S}}$ is given by

$$\mathcal{M}_{\mathfrak{S}} = (S, P_{\mathfrak{S}}, s_0, \text{AP}, L)$$

with

$$P_{\mathfrak{S}}(s, s') = P(s, \mathfrak{S}(s), s').$$



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Induced Markov Chains

- As seen on last slide: MDPs and Markov Chains are closely related
- Consequence: Given an MDP and a memoryless scheduler for it, we obtain a Markov Chain by selecting in each state the action that the scheduler suggests

Definition

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP and $\mathfrak{S}$ a memoryless scheduler on $\mathcal{M}$. The *induced Markov Chain* $\mathcal{M}_{\mathfrak{S}}$ is given by

$$\mathcal{M}_{\mathfrak{S}} = (S, P_{\mathfrak{S}}, s_0, \text{AP}, L)$$

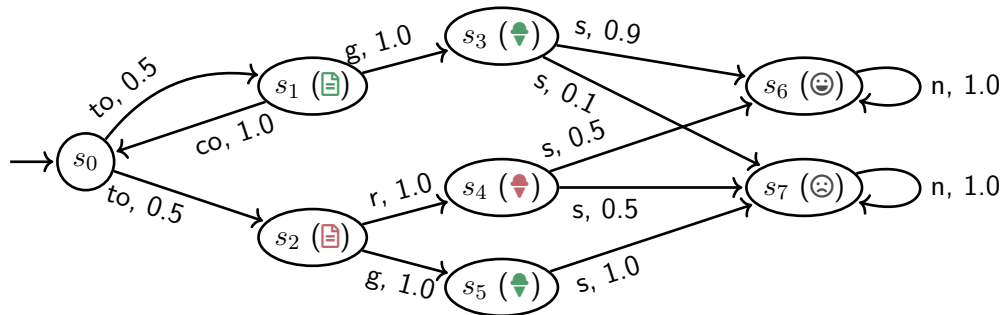
with

$$P_{\mathfrak{S}}(s, s') = P(s, \mathfrak{S}(s), s').$$

- The next slide contains an example from the ice store domain with a scheduler that maximizes customer satisfaction.



Example: Induced Markov Chain in the



- Scheduler to maximize happiness in MDP $\mathcal{M}$:

$$\mathfrak{S}(s_0) = \text{to}$$

$$\mathfrak{S}(s_1) = g$$

$$\mathfrak{S}(s_2) = r$$

$$\mathfrak{S}(s_3) = s$$

$$\mathfrak{S}(s_4) = s$$

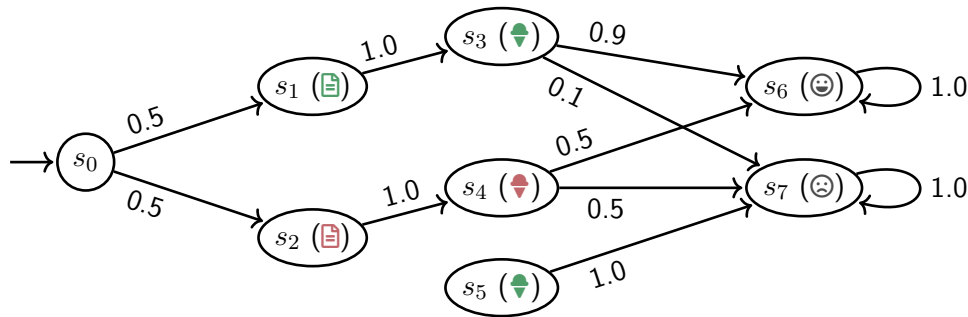
$$\mathfrak{S}(s_5) = s$$

$$\mathfrak{S}(s_6) = n$$

$$\mathfrak{S}(s_7) = n$$



Example: Induced Markov Chain in the



- Induced Markov Chain $\mathcal{M}_{\mathcal{G}}$



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Optimal Schedulers for MDPs

- Goal: Find “optimal” scheduler for an MDP
- More formal: Analyze which memoryless scheduler maximizes the probability that the runs in the induced MC satisfy a given property
- We denote by $\Pr_{\mathcal{M}}^{\max}(\varphi)$ the expected probability of satisfying φ , given the optimal scheduler



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Optimal Schedulers for MDPs



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

- Goal: Find “optimal” scheduler for an MDP
- More formal: Analyze which memoryless scheduler maximizes the probability that the runs in the induced MC satisfy a given property
- We denote by $\Pr_{\mathcal{M}}^{\max}(\varphi)$ the expected probability of satisfying φ , given the optimal scheduler

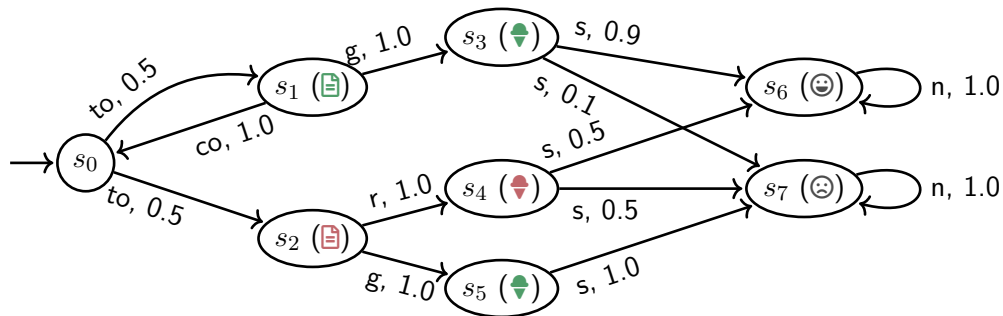
Definition

Let $\mathcal{M}$ be an MDP and φ an LTL formula. Then, we define

$$\Pr_{\mathcal{M}}^{\max}(\varphi) = \max_{\mathfrak{S}} \Pr^{\mathcal{M}_{\mathfrak{S}}}(\varphi)$$

where $\max_{\mathfrak{S}}$ is the maximum over all schedulers for $\mathcal{M}$.

Example: Optimal Schedulers for



φ **optimal scheduler (only s_1, s_2)** $\Pr_{\mathcal{M}}^{\max}(\varphi)$



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

Definition of MDPs

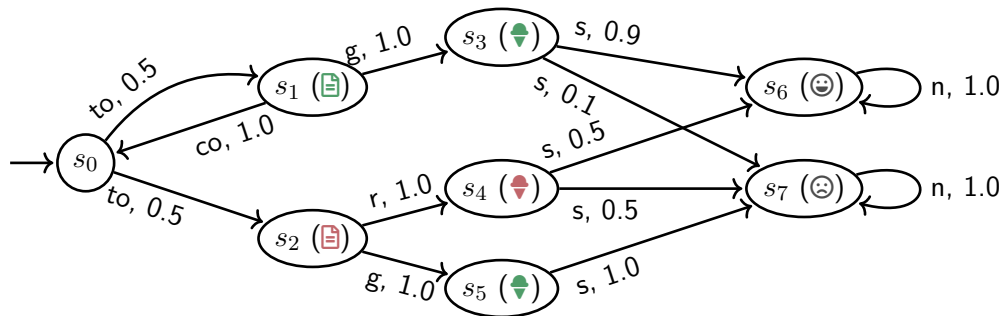
Schedulers

Optimal Schedulers

Costs

Computation

Example: Optimal Schedulers for



φ **optimal scheduler (only s_1, s_2)** $\Pr_{\mathcal{M}}^{\max}(\varphi)$
 $\diamond \text{😊}$ $\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = r$ 0.7



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Example: Optimal Schedulers for



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

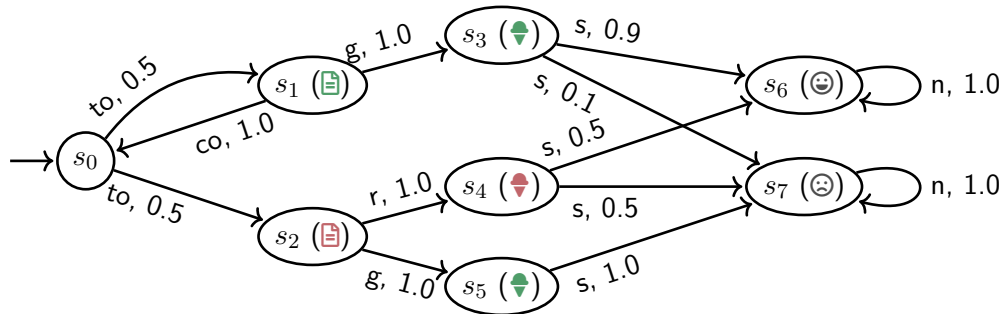
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



φ	optimal scheduler (only s_1, s_2)	$\Pr_{\mathcal{M}}^{\max}(\varphi)$
$\diamond \text{😊}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = r$	0.7
$\diamond \text{🍀}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = g$	1.0

Example: Optimal Schedulers for



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

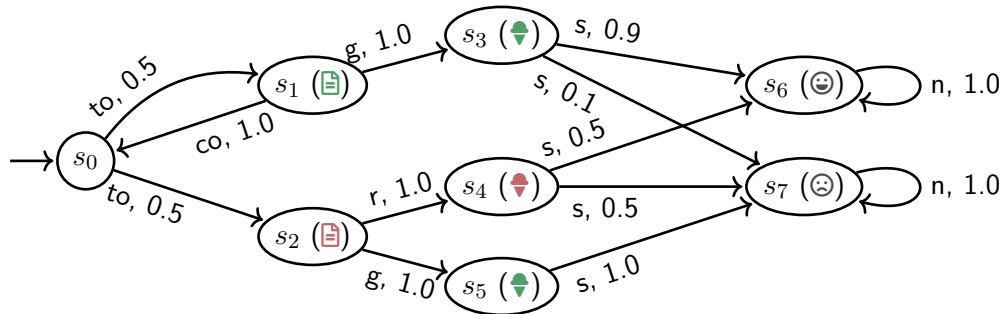
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



φ	optimal scheduler (only s_1, s_2)	$\Pr_{\mathcal{M}}^{\max}(\varphi)$
$\diamond \text{😊}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = r$	0.7
$\diamond \text{🌱}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = g$	1.0
$\diamond \text{😞}$	$\mathfrak{S}(s_1) = co, \mathfrak{S}(s_2) = g$	1.0

Example: Optimal Schedulers for



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

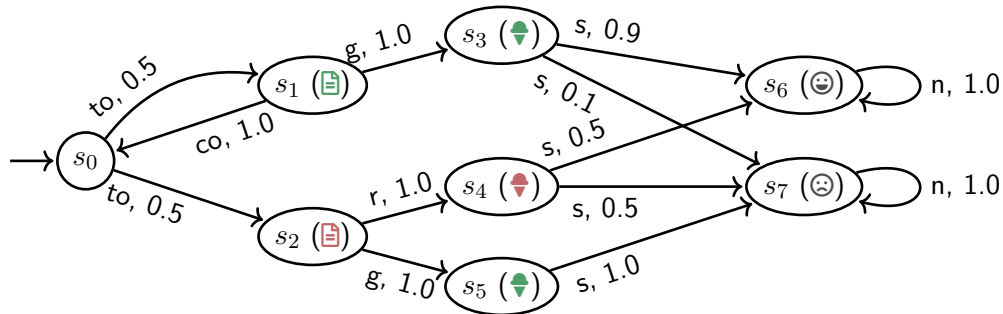
Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation



φ	optimal scheduler (only s_1, s_2)	$\Pr_{\mathcal{M}}^{\max}(\varphi)$
$\diamond \text{😊}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = r$	0.7
$\diamond \text{🌳}$	$\mathfrak{S}(s_1) = g, \mathfrak{S}(s_2) = g$	1.0
$\diamond \text{😞}$	$\mathfrak{S}(s_1) = co, \mathfrak{S}(s_2) = g$	1.0
$\diamond (\text{🌳} \wedge \bigcirc \text{😊})$	$\mathfrak{S}(s_1) = co, \mathfrak{S}(s_2) = r$	0.5

Finding Optimal Schedulers



IN5110 –
Specification and
Verification of
Parallel Systems

- There are algorithms to compute the optimal schedulers.
- The solution goes beyond this lecture. An approach similar to the one for quantitative LTL model checking for Markov chains can be used.
- Complexity: Linear in size of MDP $\mathcal{M}$ and double-exponential in size of φ . That is, it is harder to solve than the model checking problems for Markov chains.

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

MDPs: Extension with Costs



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

- So far, we only analyzed: What is the *probability* that the system has some behavior?
- Often interesting: What is the *expected cost* to reach the goal?
- We use the concept of **costs** but other sources call them “rewards”.
- Possible “costs”:
 - Time to perform actions
 - Money spend
 - Number of tasks completed
 - Money earned
- “Costs” can be positive or negative, i.e. we might want to have a scheduler that maximizes or minimizes expected cost.

MDPs: Extension with Costs

- It is possible to extend MDPs with *costs* for quantitative analysis.
- The costs are added to the states and transitions.
 - A cost is associated to each state, reflecting the cost for being in the state.
 - For each state, we assign to each action the cost that is involved in taking the action in this state.

Definition (MDP with Costs)

A *Markov decision process with costs* is a tuple $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L, C_s, C_t)$ where $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ is an MDP and $C_s : S \rightarrow \mathbb{N}$ and $C_t : S \times \text{Act} \rightarrow \mathbb{N}$ are a *cost functions* that associates a cost to each state and each pair of state and action, respectively.



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision
Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

MDPs: Extension with Costs

- It is possible to extend MDPs with *costs* for quantitative analysis.
- The costs are added to the states and transitions.
 - A cost is associated to each state, reflecting the cost for being in the state.
 - For each state, we assign to each action the cost that is involved in taking the action in this state.

Definition (MDP with Costs)

A *Markov decision process with costs* is a tuple $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L, C_s, C_t)$ where $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ is an MDP and $C_s : S \rightarrow \mathbb{N}$ and $C_t : S \times \text{Act} \rightarrow \mathbb{N}$ are a *cost functions* that associates a cost to each state and each pair of state and action, respectively.

- The cost functions can be used to describe the *cost of a path* by adding the costs for all states and transitions in the path.
- Costs of paths can be used to describe the *expected cost* of reaching a set of goal states by summing over all paths that lead to the goal states and multiplying their cost with their probability.



Examples of Usage of Costs in Models of



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

- Model the number of ice cream scoops sold with costs:
 - Cost of 1 means 1 scoop of ice cream sold
 - States associated with served ice cream get a cost of 1, all other states and all transitions get cost 0.
- Model the time taken to serve customer:
 - Costs of 1 means 10s spend
 - Depending on time necessary for actions, the costs of transitions vary. E.g. time to prepare scoop is longer than to take order. If preparing a scoop of ice cream takes 30s, the cost is 3 for all transitions with action “prepare ice cream”.



- Similar to finding a scheduler that maximizes the probability to adhere to some behavior, we can search for schedulers that “optimize expected costs”.
- Typical Problems:
 1. Find a scheduler that *minimizes* the expected cost until a goal is reached.
 2. Find a scheduler that *maximizes* the expected cost until a goal is reached.
- There are algorithmic solutions to solve these problems.

Examples of Usage of Costs in Models of

- Model the number of ice cream scoops sold with costs:
 - Cost of 1 means 1 scoop of ice cream sold
 - States associated with served ice cream get a cost of 1, all other states and all transitions get cost 0.
 - Problem: Which scheduler maximizes the cost for serving a customer? I.e. we want to sell each customer as much ice cream as possible.
- Model time taken to serve customer:
 - Costs of 1 means 10s spend
 - Depending on time necessary for actions, the costs of transitions vary. E.g. time to prepare scoop is longer than to take order.
 - Problem: Which scheduler minimizes the time for serving a customer? I.e. we want to be as fast as possible.



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Examples of Usage of Costs in Models of



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

- Model the number of ice cream scoops sold with costs:
 - Cost of 1 means 1 scoop of ice cream sold
 - States associated with served ice cream get a cost of 1, all other states and all transitions get cost 0.
 - Problem: Which scheduler maximizes the cost for serving a customer? I.e. we want to sell each customer as much ice cream as possible.
- Model time taken to serve customer:
 - Costs of 1 means 10s spend
 - Depending on time necessary for actions, the costs of transitions vary. E.g. time to prepare scoop is longer than to take order.
 - Problem: Which scheduler minimizes the time for serving a customer? I.e. we want to be as fast as possible.
- These examples show: If there are multiple costs, it is often not possible to find a scheduler that optimizes both. (First problem prioritizes long interactions with customer to sell more ice cream while the second problem prioritizes short (fast) interactions.)

Summary: Markov Decision Processes



IN5110 –
Specification and
Verification of
Parallel Systems

- MDPs are a model that combine probabilistic effects with non-deterministic behavior
- Main problem: find optimal scheduler that
 - Maximizes probability that run satisfies LTL formula
 - Maximizes / Minimizes expected cost
- Model checking problems are decidable and can be solved algorithmically, but are harder to solve than problems for Markov Chains

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Theorem (Existence of Optimal Memoryless Schedulers [1])

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP, and $B \subseteq S$. There exists a memoryless scheduler $\mathfrak{S}$ such that for any $s \in S$

$$Pr_{\mathcal{M}}^{\mathfrak{S}}(s \models \Diamond B) = Pr_{\mathcal{M}}^{\max}(s \models \Diamond B)$$

Thus, for reachability properties, memoryless schedulers are sufficient.

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

- Optimization problem where objective and constraints are linear

Definition (Linear Program [2])

Let $x = (x_1, \dots, x_n)$ be a vector of real-valued optimization variables, $c, a_0, \dots, a_m \in \mathbb{R}^n$ vectors, and $b_1, \dots, b_m \in \mathbb{R}$ constants. A *linear problem* has the form:

$$\begin{array}{ll} \min & c^T x, \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \dots, m. \end{array}$$

- Can be efficiently solved using Simplex or interior-point methods

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Encoding Memoryless Schedulers

- Reachability problems for MDPs can be encoded as linear program

Theorem (LP for Max Reachability [1])

Let $\mathcal{M} = (S, \text{Act}, P, s_0, \text{AP}, L)$ be an MDP, and $B \subseteq S$. The vector $(x_s)_{s \in S}$ with $x_s = \text{Pr}_{\mathcal{M}}^{\max}(s \models \Diamond B)$ yields the unique solution of the following LP:

- If $s \in B$, then $x_s = 1$,
- If $s \not\models \exists \Diamond B$, then $x_s = 0$,
- If $s \notin B$ and $s \models \exists \Diamond B$, then $0 \leq x_s \leq 1$ and for all actions $\alpha \in \text{Act}(s)$:

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

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



IN5110 –
Specification and
Verification of
Parallel Systems

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

Example: MDP

See [1] for an Example.



**IN5110 –
Specification and
Verification of
Parallel Systems**

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation

References I



IN5110 –
Specification and
Verification of
Parallel Systems

Bibliography

- [1] Baier, C. and Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.
- [2] Boyd, S. P. and Vandenberghe, L. (2014). *Convex Optimization*. Cambridge University Press.

Markov Decision Processes

Definition of MDPs

Schedulers

Optimal Schedulers

Costs

Computation