

---

# Multi-objective Model Checking of Markov Automata

---

by  
**Tim Quatmann**

Master Thesis at RWTH Aachen University,  
Lehrstuhl für Informatik 2

Submitted to: Fakultät für Mathematik, Informatik und  
Naturwissenschaften der RWTH Aachen  
Submission Date: September 30, 2016  
  
First examiner: Prof. Dr. Ir. Joost-Pieter Katoen  
Second examiner: apl. Prof. Dr. Thomas Noll  
Thesis advisor: Sebastian Junges



## Eidesstattliche Versicherung

Quatmann, Tim

308888

Name, Vorname

Matrikelnummer (freiwillige Angabe)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit/Bachelorarbeit/~~ Masterarbeit\* mit dem Titel

Multi-objective Model Checking of Markov Automata

selbständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Aachen, 30.09.2016

Ort, Datum

Unterschrift

\*Nichtzutreffendes bitte streichen

### Belehrung:

#### § 156 StGB: Falsche Versicherung an Eides Statt

Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

#### § 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt

(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.

(2) Strafflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:

Aachen, 30.09.2016

Ort, Datum

Unterschrift



---

## Abstract

---

Markov automata (MAs) constitute a highly expressive formalism to model systems exhibiting nondeterminism, probabilistic branching, and continuously distributed random delays. Multi-objective model checking aims to analyze possible trade-offs between quantitative objectives of the modeled system, like the expected time for completing a job or the probability for an error-free behavior within a certain time interval.

Previous works analyze MAs under schedulers that minimize (or maximize) a single objective. Known approaches do not generalize to multiple possibly conflicting objectives. For example, minimizing the expected time for completing a job might require taking higher risks, increasing the likelihood of an error. So far, the trade-off analysis between multiple objectives has only been studied for discrete-time models such as Markov decision processes (MDPs).

In this thesis, we consider combinations of multiple (un)bounded probabilistic and expected value objectives for MAs. We lift the transformation from MAs to MDPs utilized in single-objective model checking to the multi-objective case. This requires new proofs as we need to consider a broader class of schedulers. Existing techniques for multi-objective MDPs are generalized to be compatible with the transformed models. Experiments on several case-studies are conducted to indicate the practical applicability of the presented results.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Probabilistic Models</b>	<b>7</b>
2.1	Probability Theory . . . . .	7
2.2	Markov Automata . . . . .	11
2.3	Markov Decision Processes . . . . .	16
2.4	Semantics of Probabilistic Models . . . . .	18
2.4.1	Relaxed Paths . . . . .	19
2.4.2	Measurable Space of MAs . . . . .	20
2.4.3	Schedulers . . . . .	22
2.4.4	Probability Measure of MAs . . . . .	23
2.4.5	Zeno Behavior . . . . .	25
2.4.6	Non-relaxed Paths . . . . .	26
2.5	Rewards and Costs . . . . .	27
<b>3</b>	<b>Multi-objective Model Checking</b>	<b>31</b>
3.1	Objectives . . . . .	31
3.2	Multi-objective Queries . . . . .	35
<b>4</b>	<b>Analysis of Markov Automata with Multiple Objectives</b>	<b>39</b>
4.1	Unbounded Until Objectives . . . . .	40
4.2	Expected Reachability Reward Objectives . . . . .	45
4.3	Bounded Until Objectives . . . . .	52
4.3.1	Digitization Approach . . . . .	54
4.3.2	A Lower Bound for $\Pr_{\sigma}^{\mathcal{M}}(H U^{\leq b} G)$ . . . . .	58
4.3.3	An Upper Bound for $\Pr_{\sigma}^{\mathcal{M}}(H U^{\leq b} G)$ . . . . .	63
4.3.4	Until Probabilities with Lower Time-bounds . . . . .	71
4.3.5	Lifting to Multiple Bounded Until Objectives . . . . .	75
4.4	Combinations of Different Types of Objectives . . . . .	78
<b>5</b>	<b>Approximation of the Set of Achievable Points</b>	<b>83</b>
5.1	Geometric Set Representations . . . . .	85
5.2	Exploration of Achievable Points . . . . .	87

5.3	Treatment of a Broader Class of Inputs . . . . .	94
5.3.1	Transformation to Expected Total Reward Objectives . . . . .	94
5.3.2	Infinite Rewards . . . . .	97
5.3.3	Transformation to Threshold Relations $(\geq, \dots, \geq)$ . . . . .	99
5.4	Approximation for Bounded Until Objectives . . . . .	104
<b>6</b>	<b>Experimental Evaluation</b>	<b>111</b>
6.1	Implementation . . . . .	111
6.2	Experiments on MDPs . . . . .	113
6.3	Experiments on MAs . . . . .	113
<b>7</b>	<b>Conclusion</b>	<b>117</b>
7.1	Summary . . . . .	117
7.2	Future Work . . . . .	118

# Chapter 1

## Introduction

*Stochastic systems* where the behavior is subject to probabilistic choices and random delays arise in many different areas such as communication, security, or biology. Verification of such systems establishes certain properties, giving crucial information for the user.

*Markov automata (MAs)* [EHZ10b] are transition systems that model nondeterminism, probabilistic branching, and exponentially distributed random delays. This is achieved by combining notions from Markov decision processes (MDPs) [Put94] and continuous time Markov chains (CTMCs) [Nor97]. The expressiveness of MAs allows us to model a rich class of probabilistic timed systems. In particular, MAs provide semantics for several specification languages such as generalized stochastic petri nets [MCB84, Kat12] and dynamic fault trees [BCS07]. The MAPA language [TKvdPS12] enables efficient modeling of large MAs in a compositional manner.

### **Example 1.1**

We consider a client for a video streaming service. The client consecutively receives  $N$  data packages and stores them into a buffer. The buffered packages are processed during the playback of the video. The time it takes to receive (or to process) a single package is modeled by an exponentially distributed delay. Whenever a package is received and the video is not playing, the client nondeterministically chooses whether it starts the playback or whether it keeps on buffering. The latter choice is not reliable, i.e., there is a 1% chance that the playback is started anyway. In case of a buffer underrun<sup>1</sup>, the playback is paused and the client waits for new packages to arrive.

Figure 1.1 depicts an MA that models the client described above for  $N = 2$ . A run of the model starts at the state  $(\mathbf{1}, 0, 0)$ . The first data package is received with rate  $\lambda$  which leads to the state  $(?, 1, 1)$ . The client decides whether the playback should be started or not (Action  $\alpha$  or Action  $\beta$ ), where the unreliability for the latter choice

---

<sup>1</sup>A buffer underrun occurs when the next package needs to be processed while the buffer is empty.

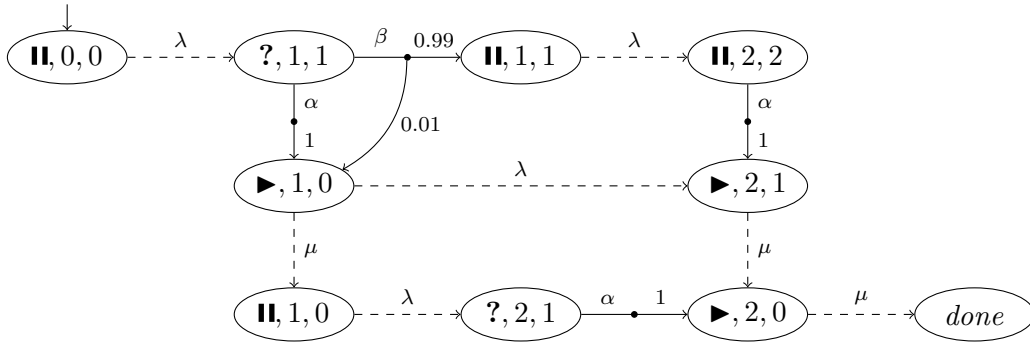


Figure 1.1: MA for a streaming client that receives  $N = 2$  packages (cf. Example 1.1).

is modeled by probabilistic branching. In state  $(\blacktriangleright, 1, 0)$ , the video is playing but we have only received one package and there are zero unprocessed packages left in the buffer. Hence, if the next package needs to be processed (rate  $\mu$ ) before it is received (rate  $\lambda$ ), we move to  $(\mathbf{II}, 1, 0)$  which represents a buffer underrun. All  $N = 2$  packages have been processed whenever the state *done* is reached. ■

*Model checking* [Cla08] is an automated technique to analyze properties of a given model. We are interested in quantitative properties, e.g.,

- the expected time until the video starts, or
- the probability for a buffer underrun (within one minute).

The quantitative analysis of MAs for different types of properties has been studied recently [HH12, GHH<sup>+</sup>13]. The existing approaches consider a single objective, i.e., a single quantitative property is maximized (or minimized) with respect to the possible resolutions of nondeterminism. Such an analysis finds applications in *controller synthesis* where one is aimed to find a scheduler for the nondeterminism such that the induced system fulfills the given requirement.

*Multi-objective model checking* is a method to analyze multiple objectives at once. The goal is to identify possible thresholds that can be assigned to the different objectives such that there is *one* scheduler achieving *all* these thresholds. To this end, the single-objective analysis is not feasible as it disregards possible conflicts between the objectives. For example, maximizing the expected reward of a system run might increase the expected costs. Multi-objective model checking reveals such trade-offs and helps the designer of the system to find a scheduler that is feasible for all objectives.

### **Example 1.2**

Consider an instance of the video streaming client and the two objectives “minimize the expected time until the video starts” and “minimize the probability for a buffer underrun within 60 seconds”. We observe the following conflict of objectives:

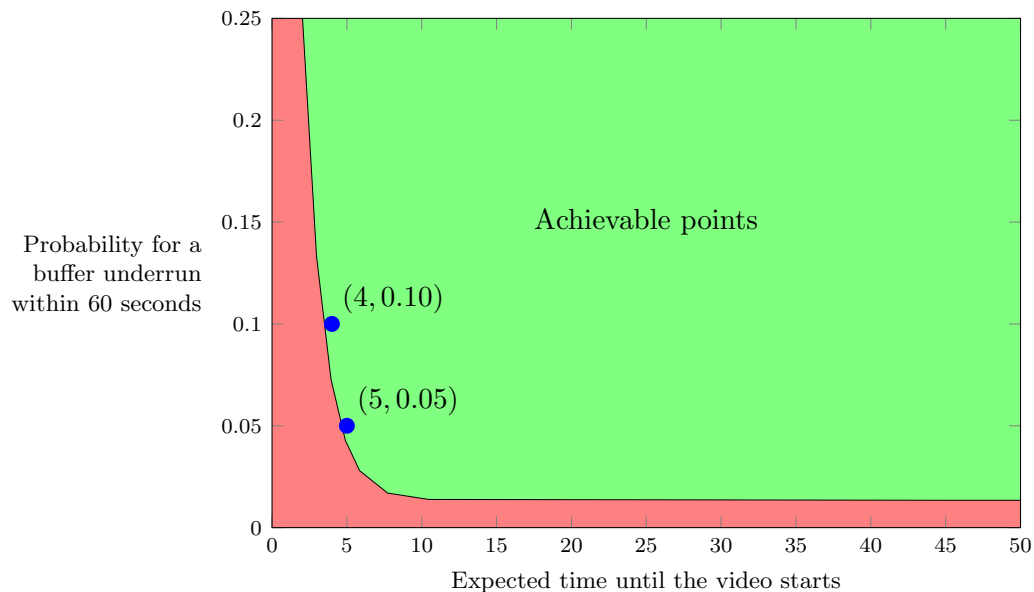


Figure 1.2: Illustration of set of achievable points (cf. Example 1.2).

- The expected time until the video starts is minimized by starting the playback when the first data package is received. However, this makes a buffer underrun very likely as we start the playback with an empty buffer.
- The probability for a buffer underrun is minimized when the start of the playback is delayed as long as possible.

We want to find the points  $\mathbf{p} = (p_1, p_2) \in \mathbb{R}^2$  such that there is *one* strategy for starting the video achieving that

- the expected time to start the video is at most  $p_1$  and
- the probability for a buffer underrun within 60 seconds is at most  $p_2$ .

We call these points *achievable*. The green area in Figure 1.2 illustrates a possible solution for the set of achievable points. From the figure, we can infer that, e.g., there is a scheduler such that the expected time to start the video is less than 5 seconds while the probability for a buffer underrun is at most 5%. In order to decrease the expected starting time to 4 seconds, an increase of the probability for a buffer underrun to approximately 10% has to be accepted.

As both objectives aim for a low value, the achievable points that lie on the border of the green area are of most interest. These points are the so-called *Pareto optimal points*. ■

**Related work.** Multi-objective model checking of probabilistic models has been studied extensively within the last decade [CMH06, EKVY08, BBC<sup>+</sup>11, FKN<sup>+</sup>11, FKP12, CFK<sup>+</sup>13]. However, these works focus on models with *discrete time* such as MDPs or stochastic games. The analysis of multiple objectives for MAs exhibiting *continuous time* has not been considered yet.

In [HH12], single-objective model checking of MAs against a variant of CSL [BHHK03] is considered. This includes the analysis of unbounded and time-bounded until objectives. Further types of objectives including expected time and expected reward objectives are discussed in [GHH<sup>+</sup>13, GTH<sup>+</sup>14]. The main idea of these approaches is to reduce the analysis of the MA to an analysis of an MDP. Given an unbounded until or an expected value objective, this is achieved by considering the underlying MDP of the MA. The result obtained from analyzing the objective on the underlying MDP also holds for the original model. For time-bounded until objectives, a digitization approach is employed in which a digitization step-bounded objective is analyzed on a digitized Markov automaton (in fact, an MDP). This approach yields a sound and arbitrarily precise approximation of the maximal (or minimal) time-bounded until probability.

Existing approaches for multi-objective model checking on MDPs are based on linear programming [FKN<sup>+</sup>11], multi-objective linear programming [CMH06, EKVY08], and value iteration [FKP12]. The latter originates from single-objective MDP analysis [Put94] and is lifted to multiple objectives by iteratively optimizing weighted combinations of the objectives. This yields a successively refined approximation of the set of achievable thresholds. The practical experiments given in [FKP12] indicate run-time efficiency and scalability of the presented procedure. Moreover, the approach supports step-bounded reachability objectives for MDPs which, according to the authors of [FKP12], “paves the way for the development of multi-objective techniques for richer, *timed* classes of models [...]”

**Contributions.** In this thesis, we enable multi-objective model checking of MAs by combining notions from single-objective MA analysis with approaches for multi-objective MDPs. Our results allow us to check multi-objective MAs with arbitrary combinations of

- unbounded until objectives,
- time-bounded until objectives,
- expected time objectives, and
- expected reward objectives.

In a first step, the reductions of MA analysis to MDP analysis from [HH12, GHH<sup>+</sup>13, GTH<sup>+</sup>14] are lifted to multiple objectives. We show the following results:

- 
- Multi-objective model checking of MAs considering arbitrary combinations of unbounded until and expected value objectives can be conducted on the underlying MDP.
  - If one or more time-bounded until objectives are considered, a multi-objective analysis of a digitized Markov automaton yields a sound and arbitrarily precise approximation of the achievable values in the original MA.

Previous results are tailored to the optimization of a single objective. The connection between an MA and the reduced MDP has only been shown under schedulers that maximize (or minimize) the considered property. However, multi-objective model checking requires to consider a larger class of schedulers. In particular, the objectives might be achievable with a scheduler that depends on the continuous-time behavior of the MA. For such cases we need to show that there is also a time-abstract scheduler for the corresponding MDP that achieves the same objectives. Our more general results require entirely new proofs in order to allow all schedulers of the MA.

In a second step, the value iteration-based approach of [FKP12] is extended. Ideas from [HH12] are incorporated into the algorithm to analyze (multiple) step-bounded objectives of a digitized Markov automaton. [FKP12] requires that either only minimizing or only maximizing expected reward objectives are considered. We avoid this rather unfavorable restriction by introducing additional preprocessing steps based on the elimination of end components.

We implemented a prototype of the presented approaches and obtained empirical data on different case studies. On MDPs, our implementation is competitive with the implementation in PRISM [KNP11]. For MAs (which are not supported by PRISM) we analyzed different combinations of objectives on large models with up to 1.5 million states.

**Structure of the thesis.** Chapter 2 gives an extensive overview of the syntax and semantics of Markov automata. The different types of considered objectives as well as multi-objective queries are discussed in Chapter 3. In Chapter 4, we generalize the results from single-objective MA analysis to multiple objectives. Chapter 5 describes the approach of [FKP12] and introduces the above-mentioned extensions. We consider experimental results in Chapter 6 and conclude the thesis in Chapter 7.



## Chapter 2

# Probabilistic Models

This chapter presents the theoretical foundations for models of *probabilistic systems* exhibiting *nondeterminism* and *continuous time*. After a brief overview of the required *probability theory* (Section 2.1), we formally introduce *Markov automata* as a powerful modeling formalism for such systems (Section 2.2). In Section 2.3 we present *Markov decision processes* as an important subclass of Markov automata where the timing is discrete. Section 2.4 defines the *semantics* of our models in terms of paths (representing a run of the system), schedulers (resolving the nondeterminism), and a probability space (assigning probabilities to sets of paths w.r.t. a scheduler). In Section 2.5 we extend Markov automata with *rewards* or, equivalently, *costs* which enable additional measures like, e.g., energy consumption, downtimes, or the number of failures.

Most of the presented definitions have been adopted from [HH12, GHH<sup>+</sup>13, GHH<sup>+</sup>14]. Further information can also be found in [Tim13, Neu10]. A gentle introduction to the analysis of probabilistic models (and model checking in general) is given in [BK08].

**Notations.** For a set  $S$ , we write  $2^S = \{S' \subseteq S\}$  for the power set of  $S$ . The set of real numbers is denoted by  $\mathbb{R}$  and the set of extended real numbers (including  $\pm\infty$ ) by  $\mathbb{R}^\infty = \mathbb{R} \cup \{-\infty, \infty\}$ . Further, we write  $\mathbb{R}_{>0} = \{x \in \mathbb{R} \mid x > 0\}$  for the positive and  $\mathbb{R}_{\geq 0} = \mathbb{R}_{>0} \cup \{0\}$  for the nonnegative real numbers.

### 2.1 Probability Theory

We present the aspects of probability theory that are most relevant for this thesis. To this end, we mainly adhere to the introductions given in [Tim13, Neu10, BK08]. A comprehensive overview of the mathematical backgrounds is provided by [ADD00].

**Definition 2.1 ( $\sigma$ -algebra, Measurable Space, Probability Space)**

A *sample space* is a set  $\Omega \neq \emptyset$ . A  $\sigma$ -*algebra* over  $\Omega$  is a set  $\mathcal{E} \subseteq 2^\Omega$  of *events* with

- $\Omega \in \mathcal{E}$ ,
- $E \in \mathcal{E}$  implies  $\Omega \setminus E \in \mathcal{E}$ , and
- $E_1, E_2, \dots \in \mathcal{E}$  implies  $\bigcup_{i=1}^{\infty} E_i \in \mathcal{E}$ .

The pair  $(\Omega, \mathcal{E})$  is called a *measurable space*. A function  $\Pr^{\mathcal{E}}: \mathcal{E} \rightarrow [0, 1]$  is called a *probability measure* on  $(\Omega, \mathcal{E})$  if it satisfies

- $\Pr^{\mathcal{E}}(\Omega) = 1$ , and
- $\Pr^{\mathcal{E}}(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \Pr^{\mathcal{E}}(E_i)$  for pairwise disjoint  $E_1, E_2, \dots \in \mathcal{E}$ .

The triple  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$  is a *probability space*. ■

A probability space  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$  can be used to mathematically model a stochastic experiment. The sample space  $\Omega$  describes the set of possible outcomes of the experiment.  $\Pr^{\mathcal{E}}(E)$  retrieves the probability that the actual outcome is in a given set  $E$  of outcomes. The set  $\mathcal{E}$  contains all sets of outcomes for which  $\Pr^{\mathcal{E}}$  is defined, i.e., for which the probability is measurable. We also refer to a set  $E \in \mathcal{E}$  as *measurable*.

**Example 2.2**

We model the toss of a (fair) coin by the probability space  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$ , where  $\Omega = \{\text{Heads}, \text{Tails}\}$ ,  $\mathcal{E} = 2^\Omega$ , and

- $\Pr^{\mathcal{E}}(\emptyset) = 0$ , the probability that the coin shows neither Heads nor Tails,
- $\Pr^{\mathcal{E}}(\{\text{Heads}\}) = 0.5$ , the probability for Heads,
- $\Pr^{\mathcal{E}}(\{\text{Tails}\}) = 0.5$ , the probability for Tails, and
- $\Pr^{\mathcal{E}}(\Omega) = 1$ , the probability that the coin shows Heads or Tails. ■

Within the scope of this thesis, the most relevant  $\sigma$ -algebra over the real numbers is the *Borel  $\sigma$ -algebra* [ADD00].

**Definition 2.3 (Borel  $\sigma$ -algebra)**

Let  $I \subseteq \mathbb{R}^\infty$  be an interval. The Borel  $\sigma$ -algebra  $\mathcal{B}(I)$  over  $I$  is the class of Borel sets of  $I$  given by the smallest  $\sigma$ -algebra that contains all intervals  $I'$  within  $I$ . ■

The *conditional probability*  $\Pr^{\mathcal{E}}(E_1 | E_2)$  intuitively describes the probability that the outcome of an experiment is in a set  $E_1 \in \mathcal{E}$ , providing that we already know that the outcome is in  $E_2 \in \mathcal{E}$ .

**Definition 2.4 (Conditional Probability)**

Let  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$  be a probability space with measurable sets  $E_1, E_2 \in \mathcal{E}$ . The conditional

probability for  $E_1$  under  $E_2$  is given by

$$\Pr^{\mathcal{E}}(E_1 | E_2) = \frac{\Pr^{\mathcal{E}}(E_1 \cap E_2)}{\Pr^{\mathcal{E}}(E_2)} . \quad \blacksquare$$

It is often convenient to add an additional layer on top of a probability space in form of *random variables*.

**Definition 2.5 (Random Variable)**

Let  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$  be a probability space and  $(\Omega', \mathcal{E}')$  be a measurable space. A random variable is a function  $X: \Omega \rightarrow \Omega'$ . ■

A random variable assigns outcomes of a stochastic experiment to another domain. In this thesis, it suffices to consider two types of random variables:

- *Discrete random variables*, where the target set  $\Omega'$  is a countable set  $S$ . In this case, we always consider the measurable space  $(S, 2^S)$ .
- *Continuous random variables*, where the target set  $\Omega'$  is given by  $\mathbb{R}^\infty$ . Here, we assume the measurable space  $(\mathbb{R}^\infty, \mathcal{B}(\mathbb{R}^\infty))$ .

Given some random variable  $X: \Omega \rightarrow \Omega'$  and some set  $A \subseteq \Omega'$ , we define the probability that  $X$  evaluates to a value in  $A$  as

$$\Pr(X \in A) = \Pr^{\mathcal{E}}(X^{-1}(A)) = \Pr^{\mathcal{E}}(\{w \in \Omega \mid X(w) \in A\}).$$

For simplicity, we may write, e.g.,  $\Pr(X \leq t)$  instead of  $\Pr(X \in \{x \in \mathbb{R}^\infty \mid x \leq t\})$ . To gain some intuition on random variables in the context of probabilistic timed systems, we now present a (rather informal) example.

**Example 2.6**

Let  $(\Omega, \mathcal{E}, \Pr^{\mathcal{E}})$  be a probability space that models the runs of a probabilistic timed system. Section 2.4 details how such a probability space can be defined. For now, we assume that  $\Omega$  contains all possible runs  $\pi$  of the system and  $\Pr^{\mathcal{E}}(\Pi)$  retrieves the probability for the system to exhibit a run contained in a set of measurable runs  $\Pi \in \mathcal{E}$ . Furthermore, we assume a set of dedicated goal states of the system.

We define random variable  $X: \Omega \rightarrow \mathbb{R}^\infty$  such that  $X(\pi)$  coincides with the time point at which the run  $\pi$  visits a goal state for the very first time. For the case that  $\pi$  never visits a goal state, we set  $X(\pi) = \infty$ . Then, the term  $\Pr(X \leq t) = \Pr^{\mathcal{E}}(\{\pi \mid X(\pi) \leq t\})$  corresponds to the probability that a goal state is visited within  $t$  time units. Note that this value is only defined if the set  $\{\pi \mid X(\pi) \leq t\}$  is measurable, i.e., contained in  $\mathcal{E}$ . ■

**Discrete probability theory.** A discrete random variable can be specified by a (*discrete*) *probability distribution function*  $\mu: S \rightarrow [0, 1]$  over a countable set  $S$  with

$\sum_{s \in S} \mu(s) = 1$ . Such a function uniquely specifies a random variable  $X: \Omega \rightarrow S$  by defining

$$\Pr(X = s) = \mu(s) \quad \text{and} \quad \Pr(X \in S') = \sum_{s \in S'} \mu(s)$$

for any  $s \in S$  and  $S' \subseteq S$ . The set of all discrete probability distribution functions over  $S$  is denoted by  $\text{Dist}(S)$ . We say that  $\mu \in \text{Dist}(S)$  is Dirac if there exists  $s \in S$  such that  $\mu(s) = 1$  (and thus  $\mu(s') = 0$  for all  $s' \in S \setminus \{s\}$ ).

**Continuous probability theory.** For a continuous random variable  $X$ , we assume that the probability for an individual outcome  $x$  is zero, i.e.,  $\Pr(X = x) = 0$ . A *probability density function*  $f: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$  with  $\int_{-\infty}^{\infty} f(x) dx = 1$  defines such a random variable  $X: \Omega \rightarrow \mathbb{R}^{\infty}$  as follows. Let  $I \subseteq \mathbb{R}^{\infty}$  be a (closed, semi-closed, or open) interval with bounds  $\inf_I = a$  and  $\sup_I = b$ . We set

$$\Pr(X \in I) = \int_a^b f(x) dx .$$

Intuitively, the *expected value* of a continuously distributed random variable  $X$  is the weighted average over all values to which  $X$  may evaluate. It is given by  $\int_{-\infty}^{\infty} x f(x) dx$ .

For  $\lambda \in \mathbb{R}_{>0}$ , we say that  $X$  is *exponentially distributed* with rate  $\lambda$  if  $X$  is defined by the probability density function

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x > 0 \\ 0 & \text{otherwise .} \end{cases}$$

A typical application of exponentially distributed random variables is to model uncertain time, e.g., the duration of a phone call or the time between two server requests. In such cases, a higher rate  $\lambda$  implies that short times are more likely. We illustrate some important properties of exponentially distributed random variables in terms of an example. More details are given in, e.g., [Neu10].

### **Example 2.7**

Consider a server that receives multiple requests from Client  $A$  at different time points. We assume that the time between two requests can be modeled by a random variable  $X_A$  that is exponentially distributed with rate  $\lambda_A \in \mathbb{R}_{>0}$  (we also say that the server receives requests from Client  $A$  with rate  $\lambda_A$ ). The probability that the time between two subsequent requests of Client  $A$  is less than  $t \in \mathbb{R}_{\geq 0}$  is given by

$$\Pr(X_A < t) = \int_{-\infty}^t f(x) dx = \int_0^t \lambda_A e^{-\lambda_A x} dx = 1 - e^{-\lambda_A t} .$$

The expected time between two requests of Client  $A$  coincides with the expected value of  $X_A$  which is given by

$$\int_{-\infty}^{\infty} x f(x) dx = \int_0^{\infty} x e^{-\lambda_A x} dx = \frac{1}{\lambda_A} .$$

Assume that the last request of Client  $A$  was  $t'$  time units ago. The probability that there is still no request after an additional  $t$  time units is

$$\Pr(X_A > t' + t \mid X_A > t') = \Pr(X_A > t) = e^{-\lambda_A t} .$$

Intuitively, this means that the probability for a request within the next  $t$  time units is independent of the time points of previous requests. This is called the *memoryless property*.

Now also consider Client  $B$  from which the server receives requests with rate  $\lambda_B \in \mathbb{R}_{>0}$  and let  $X_B$  be the corresponding exponentially distributed random variable. The time between two requests from any of the two clients is given by the random variable  $X = \min(X_A, X_B)$ . It can be shown that  $X$  is again exponentially distributed with rate  $\lambda_A + \lambda_B$ . In particular, the probability that there is a request from one of the two clients within the next  $t$  time units is given by

$$\Pr(\min(X_A, X_B) < t) = \Pr(X < t) = 1 - e^{-(\lambda_A + \lambda_B)t} .$$

Finally, the probability that the next request originates from Client  $A$  is given by

$$\Pr(X_A < X) = \frac{\lambda_A}{\lambda_A + \lambda_B} . \quad \blacksquare$$

## 2.2 Markov Automata

A *Markov automaton* [EHZ10b] can be interpreted as a transition system with two types of transitions:

- *Probabilistic transitions* which instantaneously pick a successor state according to a discrete probability distribution over the state space, and
- *Markovian transitions* that lead to a given successor state after an exponentially distributed delay.

Nondeterminism is incorporated into the formalism by allowing multiple probabilistic transitions (i.e., multiple probability distributions over successor states) originating from the same state.

### Definition 2.8 (Markov Automaton)

A *Markov automaton* ( $MA$ ) is a tuple  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  where

- $S$  is a finite set of *states* with *initial state*  $s_0 \in S$ ,
- $Act$  is a finite set of *actions* with  $\perp \in Act$  and  $Act \cap \mathbb{R}_{>0} = \emptyset$ ,
- $\rightarrow \subseteq S \times Act \times Dist(S)$  is a set of *probabilistic transitions*,

- $\dashrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$  is a set of *Markovian transitions* such that for each  $s, s' \in S$  there is at most one  $\lambda \in \mathbb{R}_{>0}$  with  $(s, \lambda, s') \in \dashrightarrow$ , and
- $\rho_1, \dots, \rho_\ell$  with  $\ell \geq 0$  are *reward functions*  $\rho_i: S \cup (S \times Act) \rightarrow \mathbb{R}_{\geq 0}$ . ■

For the rest of this chapter, we fix an MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$ . The initial state  $s_0$  marks the beginning of each run of the system. Successor states are determined by taking either a probabilistic or a Markovian transitions. For a state  $s \in S$ , we write  $\mathcal{M}^s$  for the MA  $(S, Act, \rightarrow, \dashrightarrow, s, \rho_1, \dots, \rho_\ell)$  that is obtained from  $\mathcal{M}$  by changing the initial state to  $s$ .

We denote probabilistic transitions  $(s, \alpha, \mu) \in \rightarrow$  by  $s \xrightarrow{\alpha} \mu$ . Taking such a transition from current state  $s$  means that with probability  $\mu(s')$  the next state is  $s' \in S$ . The purpose of denoting an action  $\alpha$  is to label the transition (either for synchronization with other system components or to distinguish the available probabilistic transitions at state  $s$ ). We define  $PS = \{s \in S \mid s \xrightarrow{\alpha} \mu\}$  as the set of *probabilistic states*, i.e., states that have at least one outgoing probabilistic transition.

While probabilistic transitions are instantaneous, a Markovian transition  $(s, \lambda, s') \in \dashrightarrow$  (denoted by  $s \xrightarrow{\lambda} s'$ ) is used to model a time delay specified by an exponentially distributed random variable with rate  $\lambda \in \mathbb{R}_{>0}$ . We also refer to  $\lambda$  as the rate of the transition. The rate of a state  $s \in S$  is defined as  $E(s) = \sum_{s \xrightarrow{\lambda} s'} \lambda$ , i.e., the sum of the rates of the outgoing Markovian transitions. The states with at least one outgoing Markovian transition are called *Markovian states*, denoted by  $MS = \{s \in S \mid s \xrightarrow{\lambda} s'\}$ .

A reward function  $\rho_i: S \cup (S \times Act) \rightarrow \mathbb{R}_{\geq 0}$  defines *state rewards* and *action rewards*. When sojourning in a state  $s$  for  $t$  time units (due to the delay of Markovian transitions), the state reward  $\rho_i(s) \cdot t$  is obtained. Action rewards are collected upon taking a transition. A probabilistic transition  $s \xrightarrow{\alpha} \mu$  yields reward  $\rho_i(s, \alpha)$ . For taking a Markovian transition  $s \xrightarrow{\lambda} s'$ , the reward  $\rho_i(s, \perp)$  is obtained. More details regarding the incorporation of rewards and costs are given in Section 2.5.

### Example 2.9

We extend Example 2.7 on page 10, where we considered a server that receives requests from Client  $A$  and Client  $B$  with rates  $\lambda_A$  and  $\lambda_B$ , respectively. We assume that the server processes a request with rate  $\lambda \in \mathbb{R}_{>0}$  and that it is possible to reject requests of Client  $B$ . More precisely, whenever the server receives a request from Client  $B$ , there are two possible actions:

- Action  $\alpha$ : Process the request (with probability one).
- Action  $\beta$ : Toss a virtual coin and process the request only if the coin shows Heads. Otherwise, the request is rejected.

Requests of Client  $A$  have a higher priority and can therefore not be rejected, i.e., Action  $\beta$  is not available. If the server receives a new request while the previous one has not been completed yet, the new request gets lost, leading to an error.

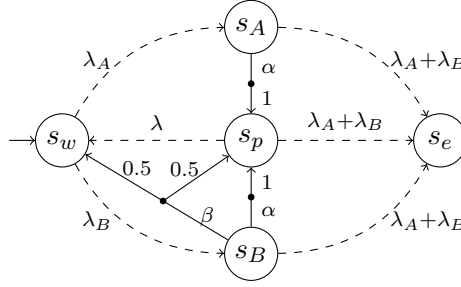


Figure 2.1: MA  $\mathcal{M}'$  that models a server with two clients (cf. Example 2.9).

Figure 2.1 shows an MA  $\mathcal{M}' = (S, Act, \rightarrow, \dashrightarrow, s_w, \emptyset)$  that models the described system. We depict the states  $S = \{s_w, s_A, s_B, s_p, s_e\}$  of  $\mathcal{M}'$  by nodes. Dashed arrows labeled with some  $\lambda \in \mathbb{R}_{>0}$  indicate Markovian transitions (e.g.,  $s_w \dashrightarrow^{\lambda_A} s_A$  holds). Note that every state of  $\mathcal{M}'$  is Markovian, i.e.,  $MS = S$ . Probabilistic transitions are depicted by solid, crotched arrows labeled with an action from the set  $Act$  and probabilities: We have  $s_B \xrightarrow{\beta} \mu$  with  $\mu$  given by  $\mu(s_w) = \mu(s_p) = 0.5$ . The set of probabilistic states is given by  $PS = \{s_A, s_B\}$ . The initial state  $s_w$  of  $\mathcal{M}'$  is marked with an unlabeled arrow.

The state  $s_w$  represents the situation where the server waits for a request from any of the two clients. When there is a request from Client A (or B), we move to state  $s_A$  ( $s_B$ ) where it is decided whether the request is processed or rejected. In  $s_B$ , the server can perform either action  $\alpha$  or action  $\beta$  as described above. In  $s_A$ , only  $\alpha$  is possible. If the current request is rejected, the server goes back to  $s_w$ , where it waits for the next request. Otherwise, it processes the request (state  $s_p$ ) with rate  $\lambda$  after which it also goes back to  $s_w$ . A new request from any of the two clients arrives with rate  $\lambda_A + \lambda_B$  (cf. Example 2.7). Whenever this is the case while the server is in a state different from  $s_w$ , we move to an error state  $s_e$ . ■

We discuss some important characteristics of Markovian states and Markovian transitions. Note that most of these results are generalizations of our observations from Example 2.7 on page 10. Let us fix some state  $s \in MS \setminus PS$ , i.e.,  $s$  has at least one Markovian transition and can not be left via a probabilistic transition. Note that  $E(s) > 0$ . Let  $X_\lambda$  denote the exponentially distributed random variable with rate  $\lambda \in \mathbb{R}_{>0}$ . It can be shown that  $\min\{X_\lambda \mid s \dashrightarrow^{\lambda} s'\}$  is exponentially distributed with rate  $E(s)$ , i.e.,  $\min\{X_\lambda \mid s \dashrightarrow^{\lambda} s'\} = X_{E(s)}$ . Thus,  $X_{E(s)}$  specifies the time at which an arbitrary Markovian transition originating from  $s$  is taken. It follows that the probability to take some transition from  $s$  within  $t$  time units is

$$\Pr(X_{E(s)} \leq t) = \int_0^t E(s)e^{-E(s)t'} dt' = (1 - e^{-E(s)t}).$$

To compute the probability to leave  $s$  via a given transition  $s \dashrightarrow^{\lambda} s'$ , one has to exclude

the cases where another transition was taken before. The desired value is given by

$$\Pr(X_\lambda = \min\{X_{\lambda'} \mid s \xrightarrow{\lambda'} s''\}) = \Pr(X_\lambda = X_{E(s)}) = \frac{\lambda}{E(s)}.$$

The probability for leaving  $s$  within  $t$  time units via a given transition  $s \xrightarrow{\lambda} s'$  is

$$\Pr(X_\lambda = X_{E(s)} \leq t) = \int_0^t \frac{\lambda}{E(s)} \cdot E(s) \cdot e^{-E(s)t'} dt' = \frac{\lambda}{E(s)} \cdot (1 - e^{-E(s)t}).$$

The expected time until some transition is taken (also referred to as the expected sojourn time of  $s$ ) is the expected value of  $X_{E(s)}$ , i.e.,

$$\int_0^\infty t \cdot E(s) \cdot e^{-E(s)t} dt = \frac{1}{E(s)}.$$

**Open and closed MAs.** The literature often distinguishes between *open* and *closed* MAs. An open MA is used to model interaction with the environment, i.e., other open MAs. This is achieved by applying a parallel composition between multiple models which means that probabilistic transitions labeled with some action  $\alpha \in Act$  may only be taken synchronously with  $\alpha$ -transitions of the other MAs. Probabilistic transitions that are not subject to any interactions are labeled with the internal action  $\tau \in Act$ . There is no synchronization for such  $\tau$ -transitions as well as for any Markovian transitions.

In contrast, closed MAs are independent of their environment, i.e., all transitions are Markovian or labeled with  $\tau$ . They are better suited for an automated analysis of the model as interactions do not need to be considered. Typically, several open MAs (modeling different parts of the system) are composed to a larger, closed MA which is then verified.

**Restrictions for MAs.** For simplification, we establish a few restrictions for the considered models. We assume that any interaction with the environment has already been incorporated, i.e., we from now on we restrict ourselves to closed MAs. For these models, it is convenient to use actions as identifier for the outgoing transitions of a state. We therefore require that each pair  $(s, \alpha) \in S \times Act$  uniquely identifies at most one probabilistic transition, i.e.,  $|\{\mu \in Dist(S) \mid s \xrightarrow{\alpha} \mu\}| \leq 1$ . This can be achieved for an arbitrary MA by renaming the actions accordingly. An MA that satisfies this restriction is called *action deterministic*.

We impose the *maximum progress assumption*: When a state of a closed MA has at least one outgoing transition of each type (probabilistic and Markovian), a probabilistic transition is taken instantaneously, i.e., without any delay. Note that the probability to take a Markovian transition without any delay is zero. Thus, the probabilistic transitions take precedence over the Markovian ones. It follows that Markovian transitions from such states can be removed and we can assume that  $PS \cap MS = \emptyset$ .

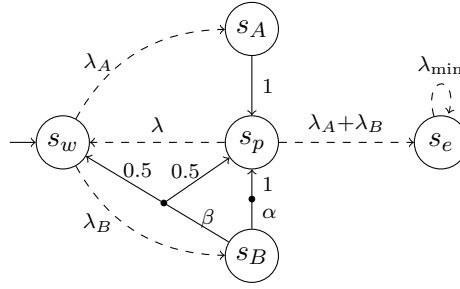


Figure 2.2: MA  $\mathcal{M}$  for the client-server example that satisfies our restrictions (cf. Example 2.9 and Example 2.10).

A state  $s \notin \text{PS} \cup \text{MS}$  which does not have any outgoing transition is called a *deadlock state*. We exclude such states by adding a transition  $s \xrightarrow{\lambda} s$  where  $\lambda$  is the minimal rate occurring in  $\mathcal{M}$ . It follows that each state is either probabilistic or Markovian, i.e.,  $\text{PS} \cup \text{MS} = S$ .

Finally, MAs that exhibit so-called *Zeno behavior* are excluded. Zeno behavior refers to the fact that there is a positive probability to observe an infinite system run within finite time. A more formal discussion of this problematic is given in Section 2.4.5.

**Example 2.10**

Reconsider the MA  $\mathcal{M}'$  depicted in Figure 2.1.  $\mathcal{M}'$  is action deterministic: Both probabilistic states  $s_A$  and  $s_B$  have exactly one transition labeled with  $\alpha$ . Furthermore, the number of outgoing  $\beta$ -transitions is zero for  $s_A$  and one for  $s_B$ .

Applying the maximum progress assumption, we observe that the outgoing Markovian transitions of  $s_A$  and  $s_B$  (i.e.,  $s_A \xrightarrow{\lambda_A + \lambda_B} s_e$  and  $s_B \xrightarrow{\lambda_A + \lambda_B} s_e$ ) can be removed. This corresponds to the assumption that the server does not delay its decision whether to perform  $\alpha$  or  $\beta$ .

The state  $s_e$  is a deadlock state as it has no outgoing transition. Hence, we add the Markovian transition  $s_e \xrightarrow{\lambda_{\min}} s_e$  with  $\lambda_{\min} = \min(\lambda_A, \lambda_B, \lambda)$ .

Figure 2.2 depicts the resulting MA  $\mathcal{M}$  obtained from  $\mathcal{M}'$ . As  $\mathcal{M}$  is considered to be closed, it is valid to omit the depiction of actions whenever there is only one outgoing probabilistic transition (see the outgoing transition of  $s_A$ ). Note that  $\mathcal{M}$  satisfies our restrictions for MAs explained above. In particular, it holds that  $\text{PS} \cup \text{MS} = \{s_A, s_B\} \cup \{s_w, s_p, s_e\} = S$ . ■

**Incorporated modeling formalisms.** MAs incorporate many well-known modeling formalisms for probabilistic systems.

- An interactive Markov chain (IMC) [Her02] is an MA where each distribution of a probabilistic transition is Dirac.

- A continuous time Markov chain (CTMC) [Nor97] is an MA without probabilistic transitions.
- A probabilistic automaton (PA) [Seg95] is an MA without Markovian transitions.
- A Markov Decision Process (MDP) [Put94] is an action deterministic PA.
- A discrete time Markov chain (DTMC) [Nor97] is a PA with at most one outgoing probabilistic transition at each state.

## 2.3 Markov Decision Processes

*Markov Decision Processes* [Put94] are particularly relevant for this thesis as the analysis of an MA can often be reduced to a (usually simpler) analysis of an MDP (see Chapter 4 for more details). To be consistent with traditional notations, we define MDPs explicitly.

### Definition 2.11 (Markov Decision Process)

A Markov Decision Process (MDP) is a tuple  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$ , where

- $S$  is a finite set of *states* with initial state  $s_0 \in S$ ,
- $Act$  is a finite set of *actions*,
- $\mathbf{P}: S \times Act \times S \rightarrow [0, 1]$  is a *transition probability function* satisfying

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

for all  $s \in S$  and  $\alpha \in Act$ , and

- $\rho_1, \dots, \rho_\ell$  with  $\ell \geq 0$  are *action reward functions*  $\rho_i: S \times Act \rightarrow \mathbb{R}_{\geq 0}$ . ■

An MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  is an MA  $\mathcal{M} = (S, Act, \rightarrow, \emptyset, s_0, \{\rho_1, \dots, \rho_\ell\})$  where the set of probabilistic transitions satisfies

$$s \xrightarrow{\alpha} \mu \iff \mu(s') = \mathbf{P}(s, \alpha, s') \text{ for all } s' \in S$$

and the reward functions  $\rho_1, \dots, \rho_\ell$  are extended with state rewards such that  $\rho_i(s) = 0$  for all  $s \in S$ . There is an outgoing transition at state  $s$  with action  $\alpha$  iff it holds that  $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') = 1$ . Deadlock states  $s \in S \setminus PS$  satisfy  $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') = 0$  for all  $\alpha \in Act$ . For MDPs, we treat such states by adding a probabilistic transition  $s \xrightarrow{\perp} \mu$  with  $\mu(s) = 1$ . Thus, it can be assumed that all states of an MDP are probabilistic, i.e.,  $PS = S$ .

**Remark 2.12**

MDPs are widely used to model nondeterministic probabilistic systems where the timing is either discrete (i.e., taking one transition means that one time unit has passed) or not relevant at all. Thus, the duration of a run through an MDP is not necessarily zero. This is in contrast to runs of MAs without Markovian transitions, where we assume that probabilistic transitions are taken instantaneously. However, the formal definitions for MAs presented in this chapter (in particular, the probability space for MAs defined in Section 2.4) carry over to MDPs in a straightforward way. ■

**Discrete time Markov chain.** *Discrete time Markov Chains* [Nor97] are MDPs without nondeterministic choices.

**Definition 2.13 (Discrete Time Markov Chain)**

A discrete time Markov chain (DTMC) is an MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  satisfying  $|\{\alpha \in Act \mid \mathbf{P}(s, \alpha, s') > 0, s' \in S\}| = 1$  for all  $s \in S$ . ■

**Definition 2.14 (Induced DTMC)**

For an MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  and a function  $\sigma: S \rightarrow Act$ , the DTMC induced by  $\sigma$  is given by  $\mathcal{D}_\sigma = (S, Act, \mathbf{P}', s_0, \{\rho_1, \dots, \rho_\ell\})$ , where

$$\mathbf{P}'(s, \alpha, s') = \begin{cases} \mathbf{P}(s, \alpha, s') & \text{if } \sigma(s) = \alpha \\ 0 & \text{otherwise.} \end{cases} \quad \blacksquare$$

The function  $\sigma: S \rightarrow Act$  in the definition above is referred to as a (stationary and deterministic) *scheduler* that resolves the nondeterminism of the considered MDP. We discuss more general classes of schedulers and their application to MAs in Section 2.4.

**Underlying MDP.** For an MA  $\mathcal{M}$  we denote by  $\mathcal{M}_{\mathcal{D}}$  the *underlying MDP* of  $\mathcal{M}$  which intuitively abstracts away from the continuous timing.

**Definition 2.15 (Underlying MDP)**

For MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_\ell\})$  the underlying MDP of  $\mathcal{M}$  is given by  $\mathcal{M}_{\mathcal{D}} = (S, Act, \mathbf{P}, s_0, \{\rho_1^{\mathcal{D}}, \dots, \rho_\ell^{\mathcal{D}}\})$ , where

$$\mathbf{P}(s, \alpha, s') = \begin{cases} \mu(s') & \text{if } s \xrightarrow{\alpha} \mu \\ \lambda/E(s) & \text{if } \alpha = \perp, s \xrightarrow{\lambda} s' \\ 0 & \text{otherwise} \end{cases}$$

and for each  $i \in \{1, \dots, \ell\}$

$$\rho_i^{\mathcal{D}}(s, \alpha) = \begin{cases} \rho_i(s, \alpha) & \text{if } s \in \text{PS} \\ \rho_i(s, \perp) + 1/E(s) \cdot \rho_i(s) & \text{if } s \in \text{MS and } \alpha = \perp \\ 0 & \text{otherwise.} \end{cases} \quad \blacksquare$$

Let us fix the underlying MDP  $\mathcal{M}_{\mathcal{D}} = (S, Act, \mathbf{P}, s_0, \{\rho_1^{\mathcal{D}}, \dots, \rho_\ell^{\mathcal{D}}\})$  of  $\mathcal{M}$ . Note that the function  $\mathbf{P}$  is a valid transition probability function due to the restrictions we made for MAs. The value  $\mathbf{P}(s, \alpha, s')$  corresponds to the probability to move from current state  $s$  with action  $\alpha$  to  $s'$ . This is directly clear for  $s \in \text{PS}$ . For  $s \in \text{MS}$  recall that the probability to leave  $s$  via a given transition  $s \xrightarrow{\lambda} s'$  is given by  $\lambda/E(s)$ . We also say that  $\mathbf{P}$  is the *transition probability function* of  $\mathcal{M}$ .

The reward functions  $\rho_1^{\mathcal{D}}, \dots, \rho_\ell^{\mathcal{D}}$  incorporate the action and state rewards of  $\mathcal{M}$  where the state rewards are multiplied with the expected sojourn times  $1/E(s)$  of states  $s \in \text{MS}$ . The underlying MDP plays an important role for the analysis of  $\mathcal{M}$  since several properties of  $\mathcal{M}$  can be analyzed on the (more simple) model  $\mathcal{M}_{\mathcal{D}}$ . This connection is detailed in Chapter 4.

### Example 2.16

Let  $\mathcal{M}$  be the MA of our client-server example depicted in Figure 2.2. The underlying MDP  $\mathcal{M}_{\mathcal{D}}$  of  $\mathcal{M}$  is shown in Figure 2.3. Recall that actions of states with only one outgoing probabilistic transition are not depicted explicitly. This particularly holds for the action  $\perp$  of the former Markovian states  $s_w$ ,  $s_p$ , and  $s_e$ . ■

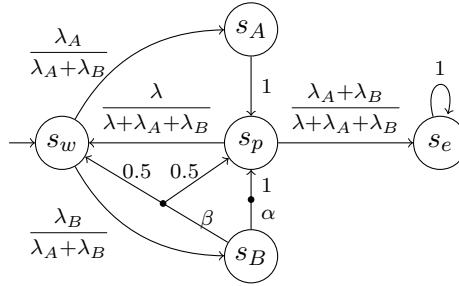


Figure 2.3: The underlying MDP  $\mathcal{M}_{\mathcal{D}}$  of the MA  $\mathcal{M}$  (cf. Example 2.16).

## 2.4 Semantics of Probabilistic Models

We give a formal definition for the behavior of MAs and MDPs by assigning probabilities to sets of runs of the modeled system. This allows us to consider the probability of, e.g., all erroneous runs or all runs that finish a task within a given time bound. To this end, we define *paths* of a model as a representation of system runs. A *measurable space* (cf. Definition 2.1 on page 7) over sets of paths is constructed and serves as a basis to measure probabilities for MAs and MDPs. Then, so-called *schedulers*<sup>1</sup> are introduced to handle the nondeterminism occurring in the model. Finally, a *probability measure* is defined for any resolution of nondeterminism in order to retrieve the probability for a given set of paths. To simplify these definitions, we initially consider

<sup>1</sup>also referred to as adversaries, policies, or strategies.

a relaxed notion for paths that also covers impossible system behavior. *Non-relaxed paths* are considered at the end of this section.

Besides the already mentioned literature, definitions from [ZN10, NSK09] have been adapted to MAs.

### 2.4.1 Relaxed Paths

#### Definition 2.17 (Infinite Relaxed Path)

An (*infinite*) *relaxed path* of MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  is an infinite sequence  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots$  of states  $s_0, s_1, \dots \in S$  and stamps  $\kappa_0, \kappa_1, \dots \in \mathbb{R}_{\geq 0} \times Act$ . ■

A relaxed path of an MA  $\mathcal{M}$  intuitively describes a single run of the modeled system, starting in the initial state  $s_0$  of  $\mathcal{M}$ . For some *stamp*  $\kappa_i = (t_i, \alpha_i) \in \mathbb{R}_{\geq 0} \times Act$  of a path  $s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots$ , the value  $t(\kappa_i) = t_i$  represents the sojourn time at state  $s_i$  and is referred to as *time-stamp*. The action  $\alpha(\kappa_i) = \alpha_i$  is called *action-stamp* and indicates the performed transition at  $s_i$  which leads to the successor state  $s_{i+1}$ . Markovian transitions are represented by the dedicated action  $\perp \in Act$ . The set of all infinite relaxed paths of  $\mathcal{M}$  is denoted by  $IPaths_{\text{rel}}^{\mathcal{M}}$ . Paths are referred to as relaxed since we allow arbitrary sequences of states and stamps even if an infeasible system run is represented (as shown in the example below). The purpose of this relaxation is to simplify the definition of a probability space for MAs. We also define non-relaxed paths that only represent feasible system runs in Section 2.4.6.

#### Example 2.18

Consider the MA  $\mathcal{M}$  from Figure 2.2 on page 15 modeling our client-server example and the path  $\pi \in IPaths_{\text{rel}}^{\mathcal{M}}$  of  $\mathcal{M}$  with

$$\pi = s_w \xrightarrow{3, \perp} s_B \xrightarrow{0, \beta} s_p \xrightarrow{1.4, \perp} s_e \xrightarrow{1, \perp} s_e \xrightarrow{1, \perp} \dots$$

$\pi$  corresponds to the following system run: A request of Client  $B$  is received after 3 time units. The server chooses action  $\beta$  which yields that the request is processed. After 1.4 additional time units, another request is received while still processing the current one which leads to an error.

The path  $\pi' = s_w \xrightarrow{1, \alpha} s_e \xrightarrow{1, \perp} s_e \xrightarrow{1, \perp} \dots$  represents an infeasible system run as it is not possible to perform the action  $\alpha$  at the Markovian state  $s_w$ . Furthermore,  $s_e$  can not be reached from  $s_w$  with a single transition. ■

A *finite relaxed path* is a finite prefix  $\pi' = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n$  of a path in  $IPaths_{\text{rel}}^{\mathcal{M}}$ . We refer to  $n$  as the *length* of  $\pi'$  (denoted by  $|\pi'|$ ) and to  $s_n$  as the last state of  $\pi'$  (denoted by  $last(\pi')$ ). The set of all finite relaxed paths of  $\mathcal{M}$  is given by  $FPaths_{\text{rel}}^{\mathcal{M}}$ .

We sometimes omit the superscript  $\mathcal{M}$  of the sets  $IPaths_{\text{rel}}^{\mathcal{M}}$  and  $FPaths_{\text{rel}}^{\mathcal{M}}$  if the model is clear from the context.

If  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots$  is a finite or infinite path of  $\mathcal{M}$ ,  $\text{pref}(\pi, n) = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n \in FPaths_{\text{rel}}$  denotes the *prefix* of  $\pi$  of length  $n$ . The *time duration* of finite  $\pi$  is given by  $T(\pi) = \sum_{0 \leq i < |\pi|} t(\kappa_i)$ . For some  $t \in \mathbb{R}_{\geq 0}$ , the prefix of  $\pi$  up to time point  $t$  is  $\text{pref}_T(\pi, t) = \text{pref}(\pi, \max\{n \mid T(\text{pref}(\pi, n)) \leq t\})$ .

Time-stamps are not relevant for paths of MDPs. We therefore omit them and consider paths of the form  $s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$  with  $\alpha_i \in Act$  for all  $i \geq 0$ . The *time-abstraction* of a path  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in IPaths_{\text{rel}}^{\mathcal{M}}$  of the MA  $\mathcal{M}$  is obtained by removing the time-stamps, denoted by  $\text{ta}(\pi) = s_0 \xrightarrow{\alpha(\kappa_0)} s_1 \xrightarrow{\alpha(\kappa_1)} \dots$ . The set of time-abstract paths of  $\mathcal{M}$  is given by  $\text{ta}(IPaths_{\text{rel}}^{\mathcal{M}}) = \{\text{ta}(\pi) \mid \pi \in IPaths_{\text{rel}}^{\mathcal{M}}\}$ . Note that  $\text{ta}(\pi)$  is a path of the underlying MDP  $\mathcal{M}_{\mathcal{D}}$ . It follows that  $\text{ta}(IPaths_{\text{rel}}^{\mathcal{M}}) = IPaths_{\text{rel}}^{\mathcal{M}_{\mathcal{D}}}$ . Similar notions hold for finite paths.

### Example 2.19

Consider again the path  $\pi$  from Example 2.18 given by

$$\pi = s_w \xrightarrow{3, \perp} s_B \xrightarrow{0, \beta} s_p \xrightarrow{1.4, \perp} s_e \xrightarrow{0.6, \perp} s_e \xrightarrow{1, \perp} \dots$$

Applying the notions defined above yields, e.g.:

- The first stamp  $\kappa = (3, \perp)$  comprises the time-stamp  $t(\kappa) = 3$  and the action-stamp  $\alpha(\kappa) = \perp$
- $\text{pref}(\pi, 2) = \pi' = s_w \xrightarrow{3, \perp} s_B \xrightarrow{0, \beta} s_p \in FPaths_{\text{rel}}^{\mathcal{M}}$  is the prefix of  $\pi$  of length  $|\pi'| = 2$  with  $\text{last}(\pi) = s_p$
- $\pi'$  is also the prefix of  $\pi$  up to time point 4 as the time durations of  $\text{pref}(\pi, 2)$  and  $\text{pref}(\pi, 3)$  satisfy  $T(\text{pref}(\pi, 2)) = 3 + 0 \leq 4 < T(\text{pref}(\pi, 3)) = 3 + 0 + 1.4$
- The time-abstraction of  $\pi$  is given by  $\text{ta}(\pi) = s_w \xrightarrow{\perp} s_B \xrightarrow{\beta} s_p \xrightarrow{\perp} s_e \xrightarrow{\perp} s_e \xrightarrow{\perp} \dots$  ■

## 2.4.2 Measurable Space of MAs

To argue about probabilities of an MA  $\mathcal{M}$ , the first step is to define a measurable space  $(IPaths_{\text{rel}}, \mathcal{E}^{IPaths})$  on infinite relaxed paths of  $\mathcal{M}$ . Intuitively, the  $\sigma$ -algebra  $\mathcal{E}^{IPaths}$  defines for which sets of paths the probability can be retrieved. Put differently,  $\mathcal{E}^{IPaths}$  provides the foundation of the definition of a probability measure. It is assembled by a combination of more simple  $\sigma$ -algebras.

Let  $\mathcal{E}_1, \dots, \mathcal{E}_n$  be  $\sigma$ -algebras. The smallest  $\sigma$ -algebra containing all sets  $\{E_1 \times \dots \times E_n \mid E_i \in \mathcal{E}_i \text{ for } 1 \leq i \leq n\}$  is called the *product  $\sigma$ -algebra* [ADD00] over  $\mathcal{E}_1, \dots, \mathcal{E}_n$  and is

denoted by  $\sigma(\mathcal{E}_1 \times \cdots \times \mathcal{E}_n)$ . Moreover, if  $\mathcal{E} \subseteq 2^\Omega$  is a set of events from some sample space  $\Omega$ , we denote by  $\sigma(\mathcal{E})$  the smallest  $\sigma$ -algebra containing all sets in  $\mathcal{E}$ .

**Definition 2.20 ( $\sigma$ -algebra for Finite Paths)**

For an MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$ , let  $\mathcal{E}_S = 2^S$ ,  $\mathcal{E}_{s_0} = 2^{\{s_0\}}$ , and  $\mathcal{E}_{Act} = 2^{Act}$  denote  $\sigma$ -algebras over the states, initial state, and actions of  $\mathcal{M}$ , respectively. Further, let  $\mathcal{B}(\mathbb{R}_{\geq 0})$  be the class of Borel sets of  $\mathbb{R}_{\geq 0}$ . The  $\sigma$ -algebra over *time-stamped steps* of  $\mathcal{M}$  is given by

$$\mathcal{E}^{Steps} = \sigma(\mathcal{B}(\mathbb{R}_{\geq 0}) \times \mathcal{E}_{Act} \times \mathcal{E}_S).$$

The  $\sigma$ -algebras over *finite relaxed paths* of fixed and arbitrary length are given by

$$\mathcal{E}_n^{FPaths} = \sigma(\mathcal{E}_{s_0} \times \underbrace{\mathcal{E}^{Steps} \times \cdots \times \mathcal{E}^{Steps}}_{n \text{ times}}) \quad \text{and} \quad \mathcal{E}^{FPaths} = \sigma\left(\bigcup_{n \geq 0} \mathcal{E}_n^{FPaths}\right). \quad \blacksquare$$

Let  $\Omega^{Steps}$  and  $\Omega_n^{FPaths}$  denote the sample spaces of the  $\sigma$ -algebras  $\mathcal{E}^{Steps}$  and  $\mathcal{E}_n^{FPaths}$ , respectively. A tuple  $(t, \alpha, s) \in \Omega^{Steps}$  describes an infix  $\xrightarrow{t, \alpha} s$  of a path of  $\mathcal{M}$ , representing a single transition.  $\Omega_n^{FPaths}$  contains tuples of the form  $(s_0, (t_0, \alpha_0, s_1), \dots, (t_{n-1}, \alpha_{n-1}, s_n))$  which are interpreted as finite relaxed paths  $s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n$  where  $\kappa_i = (t_i, \alpha_i)$  for all  $i$ . It follows that the sample space of  $\mathcal{E}^{FPaths}$  is the set of finite relaxed paths  $FPaths_{\text{rel}}$ . To lift the presented notions to infinite paths, we employ a standard *cylinder set construction* [ADD00].

**Definition 2.21 (Cylinder of a Set of Finite Paths)**

The cylinder of a set  $\Pi \in \mathcal{E}^{FPaths}$  is defined as

$$Cyl(\Pi) = \{\pi \xrightarrow{\kappa_n} s_{n+1} \xrightarrow{\kappa_{n+1}} \dots \mid \pi \in \Pi \text{ and } \xrightarrow{\kappa_i} s_{i+1} \in \Omega^{Steps} \text{ for } i \geq n = |\pi|\}. \quad \blacksquare$$

The cylinder of  $\Pi \in \mathcal{E}^{FPaths}$  is the set of all extensions of a path  $\pi \in \Pi$  to an infinite path. This notion allows us to define the  $\sigma$ -algebra  $\mathcal{E}^{IPaths}$ .

**Definition 2.22 ( $\sigma$ -algebra for Infinite Paths)**

Let  $\mathcal{M}$  be an MA. The  $\sigma$ -algebra over *infinite relaxed paths* of  $\mathcal{M}$  is given by the smallest  $\sigma$ -algebra containing all cylinders, i.e.,

$$\mathcal{E}^{IPaths} = \sigma\left(\bigcup_{n \geq 0} \{Cyl(\Pi) \mid \Pi \in \mathcal{E}_n^{FPaths}\}\right). \quad \blacksquare$$

Analogous to the finite case, the elements of the sample space of  $\mathcal{E}^{IPaths}$  are seen as the infinite paths  $\pi \in IPaths_{\text{rel}}$  of  $\mathcal{M}$ . The pair  $(IPaths_{\text{rel}}, \mathcal{E}^{IPaths})$  forms the *measurable space* of  $\mathcal{M}$ .

### 2.4.3 Schedulers

A scheduler defines for each finite path  $\pi$  a probability distribution over the actions that are possible at  $last(\pi)$ , effectively resolving the nondeterminism of  $\mathcal{M}$ . We define the set of actions *enabled* at some state  $s \in S$  by

$$Act(s) = \begin{cases} \{\alpha \in Act \mid s \xrightarrow{\alpha} \mu\} & \text{if } s \in \text{PS} \\ \{\perp\} & \text{if } s \in \text{MS}. \end{cases}$$

The most general class of schedulers of MAs is defined as follows.

**Definition 2.23 (Generic Scheduler)**

A *generic scheduler* for MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  is a function

$$\sigma: FPaths_{\text{rel}} \times Act \rightarrow [0, 1] \quad \blacksquare$$

such that for each  $\pi \in FPaths_{\text{rel}}$

- $\sigma(\pi, \cdot) \in \text{Dist}(Act)$ , i.e.,  $\sum_{\alpha \in Act} \sigma(\pi, \alpha) = 1$ , and
- $\sigma(\pi, \alpha) = 0$  for all  $\alpha \notin Act(last(\pi))$ .

A generic scheduler  $\sigma$  retrieves for any finite path  $\pi$  a probability distribution over the actions enabled at  $last(\pi)$ . If  $last(\pi) \in \text{PS}$ , the value  $\sigma(\pi, \alpha)$  is the probability that the transition  $last(\pi) \xrightarrow{\alpha} \mu$  is taken when the run  $\pi$  has been observed. Note that the case  $last(\pi) \in \text{MS}$  implies that  $\sigma(\pi, \cdot)$  is a Dirac distribution defined by  $\sigma(\pi, \perp) = 1$ . To ensure that the resolving of nondeterminism induces measurable probabilities, we restrict ourselves to generic schedulers that satisfy  $\{\pi \in FPaths_{\text{rel}} \mid \sigma(\pi, \alpha) \in B\} \in \mathcal{E}^{FPaths}$  for every  $\alpha \in Act$  and Borel set  $B \in \mathcal{B}([0, 1])$ . This class of *generic measurable schedulers* is denoted by  $\text{GM}^{\mathcal{M}}$  (or simply GM if  $\mathcal{M}$  is clear from the context).

*Time-abstract schedulers* behave independent of the time-stamps of the given path. Formally, a scheduler  $\sigma \in \text{GM}$  is time-abstract if for each pair of paths  $\pi, \pi' \in FPaths_{\text{rel}}$  with  $\text{ta}(\pi) = \text{ta}(\pi')$  and for each action  $\alpha \in Act$  we have  $\sigma(\pi, \alpha) = \sigma(\pi', \alpha)$ . Note that this is the most general scheduler class for MDPs.

A scheduler  $\sigma \in \text{GM}$  is *stationary*, if the specified probability distributions only depend on the current state (rather than the entire path), i.e., for each pair of paths  $\pi, \pi' \in FPaths_{\text{rel}}$  with  $last(\pi) = last(\pi')$  and for each action  $\alpha \in Act$  it holds that  $\sigma(\pi, \alpha) = \sigma(\pi', \alpha)$ . Note that stationary schedulers are also time abstract.

*Deterministic schedulers* specify only Dirac probability distributions, i.e.,  $\sigma \in \text{GM}$  is deterministic if for all  $\pi \in FPaths_{\text{rel}}$  there is exactly one  $\alpha \in Act$  with  $\sigma(\pi, \alpha) = 1$ .

Let  $\text{TA}^{\mathcal{M}}$  and  $\text{ST}^{\mathcal{M}}$  denote the sets of time-abstract and stationary schedulers of  $\mathcal{M}$ . For the sake of simplicity, we sometimes deviate from Definition 2.23 and write, e.g.,

- $\sigma \in \text{TA}$  as a function  $\sigma: \text{ta}(FPaths_{\text{rel}}^{\mathcal{M}}) \times \text{Act} \rightarrow [0, 1]$ ,
- $\sigma \in \text{ST}$  as a function  $\sigma: S \times \text{Act} \rightarrow [0, 1]$ , or
- deterministic  $\sigma \in \text{ST}$  as a function  $\sigma: S \rightarrow \text{Act}$ .

**Example 2.24**

Again consider the client-server example modeled by MA  $\mathcal{M}$  shown in Figure 2.2 on page 15. Note that  $s_B$  is the only state at which the choice of a scheduler is not unique. Thus, a scheduler  $\sigma$  is uniquely given by describing the situations at which  $\sigma$  chooses  $\beta$ . Consider the following schedulers.

- $\sigma_1 \in \text{GM}^{\mathcal{M}}$  which chooses  $\beta$  whenever the last request was processed less than ten time units ago, i.e.,

$$\sigma_1(\pi, \beta) = \begin{cases} 1 & \text{if } t(\kappa_{n-1}) < 10 \\ 0 & \text{otherwise.} \end{cases}$$

for all  $\pi = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n \in FPaths_{\text{rel}}$  with  $s_n = s_B$ .  $\sigma_1$  is deterministic but not time-abstract (and therefore also not stationary).

- $\sigma_2 \in \text{TA}^{\mathcal{M}}$  which chooses  $\beta$  if there was no request from Client  $A$  yet, i.e.,

$$\sigma_2(\pi, \beta) = \begin{cases} 1 & \text{if } s_A \text{ does not occur in } \pi \\ 0 & \text{otherwise} \end{cases}$$

for all  $\pi \in \text{ta}(FPaths_{\text{rel}})$  with  $\text{last}(\pi) = s_B$ .  $\sigma_2$  is deterministic and time-abstract but not stationary.

- $\sigma_3 \in \text{ST}^{\mathcal{M}}$  which chooses  $\beta$  with probability 0.9, i.e.,

$$\sigma_3(s_B, \beta) = 0.9 .$$

$\sigma_2$  is stationary (and therefore also time-abstract) but not deterministic. ■

#### 2.4.4 Probability Measure of MAs

We define a probability measure  $\Pr_{\sigma}^{\mathcal{M}}$  for the measurable space  $(IPaths_{\text{rel}}, \mathcal{E}^{IPaths})$  of an MA  $\mathcal{M}$ . In fact, a different probability measure is defined for any possible resolution of nondeterminism, i.e., for any given scheduler. Let  $\sigma \in \text{GM}$  be some scheduler for  $\mathcal{M}$ .

The first step is to define a measure  $\Pr_{\sigma, \pi}^{\text{Steps}}$  for measurable sets of time-stamped steps  $E \in \mathcal{E}^{\text{Steps}}$ . Note that  $\Pr_{\sigma, \pi}^{\text{Steps}}$  depends on  $\sigma$  as well as a path  $\pi \in FPaths_{\text{rel}}$  that describes the history leading to the considered steps. Intuitively,  $\Pr_{\sigma, \pi}^{\text{Steps}}$  defines

probabilities of transitions emerging from  $last(\pi)$  under the distribution over actions given by  $\sigma(\pi, \cdot)$ . For  $E \in \mathcal{E}^{Steps}$  and  $\pi \in FPaths_{rel}$  with  $s = last(\pi)$  we set

$$\Pr_{\sigma, \pi}^{Steps}(E) = \begin{cases} \sum_{(0, \alpha, s') \in E} \sigma(\pi, \alpha) \cdot \mathbf{P}(s, \alpha, s') & \text{if } s \in \text{PS} \\ \int_{\substack{t \in \mathbb{R}_{\geq 0} \\ (t, \perp, s) \in E}} \mathbf{E}(s) \cdot e^{-\mathbf{E}(s)t} \cdot \sum_{(t, \perp, s') \in E} \mathbf{P}(s, \perp, s') dt & \text{if } s \in \text{MS} \end{cases}$$

where  $\mathbf{P}$  refers to the transition probability function of  $\mathcal{M}$ . We now define a probability measure  $\Pr_{\sigma, n}^{FPaths}$  for  $\mathcal{E}_n^{FPaths}$  that assigns probabilities to measurable sets of paths  $\Pi_n \in \mathcal{E}_n^{FPaths}$  of some fixed length  $n$ . The definition is inductive. For  $n = 0$  we set

$$\Pr_{\sigma, 0}^{FPaths}(\Pi_0) = \begin{cases} 1 & \text{if } s_0 \in \Pi_0 \\ 0 & \text{otherwise.} \end{cases}$$

For  $n > 0$  we consider a set  $\Pi_{n-1} \in \mathcal{E}_{n-1}^{FPaths}$  of paths of length  $n-1$  and a set  $E \in \mathcal{E}^{Steps}$  of time-stamped steps. Let  $\Pi_{n-1} \circ E = \{\pi \xrightarrow{\kappa} s \mid \pi \in \Pi_{n-1} \text{ and } \xrightarrow{\kappa} s \in E\} \in \mathcal{E}_n^{FPaths}$  be the element-wise concatenation of the two sets. We define

$$\Pr_{\sigma, n}^{FPaths}(\Pi_{n-1} \circ E) = \int_{\pi \in \Pi_{n-1}} \Pr_{\sigma, \pi}^{Steps}(E) d\Pr_{\sigma, n-1}^{FPaths}(\{\pi\}).$$

The integral in the equation above is a so-called *Lebesgue-integral* [ADD00]. Roughly, it describes a sum over the (possibly uncountable) domain  $\Pi_{n-1}$  where we add for any  $\pi \in \Pi_{n-1}$  the probability of the steps in  $E$  (term  $\Pr_{\sigma, \pi}^{Steps}(E)$ ) weighted with the probability of the previous steps (term  $d\Pr_{\sigma, n-1}^{FPaths}(\{\pi\})$ ). Finally, we lift the definition to sets of infinite paths.

### Definition 2.25 (Probability Measure of MA)

The probability measure of an MA  $\mathcal{M}$  for a given scheduler  $\sigma \in \text{GM}$  is the unique probability measure  $\Pr_{\sigma}^{\mathcal{M}}$  for  $(IPaths_{rel}, \mathcal{E}^{IPaths})$  that satisfies

$$\Pr_{\sigma}^{\mathcal{M}}(Cyl(\Pi)) = \Pr_{\sigma, n}^{FPaths}(\Pi)$$

for all  $n \geq 0$  and  $\Pi \in \mathcal{E}_n^{FPaths}$ . ■

It can be shown that  $\Pr_{\sigma}^{\mathcal{M}}$  is well-defined and unique. More details are given in, e.g., [Neu10]. To simplify the notation, we sometimes omit the superscript  $\mathcal{M}$ . Further, we also apply  $\Pr_{\sigma}$  to measurable sets of finite paths  $\Pi \in \mathcal{E}^{FPaths}$ , i.e., we write  $\Pr_{\sigma}(\Pi)$  instead of  $\Pr_{\sigma}(Cyl(\Pi))$ . For a singleton set  $\{\pi\}$ , we usually omit the braces and write  $\Pr_{\sigma}(\pi)$  instead of  $\Pr_{\sigma}(\{\pi\})$ . We refer to  $(IPaths_{rel}, \mathcal{E}^{IPaths}, \Pr_{\sigma})$  as the *probability space of  $\mathcal{M}$  under  $\sigma$* .

### Example 2.26

Let  $\mathcal{M}$  be the MA for the client-server example shown in Figure 2.2 on page 15. Furthermore, consider the unique deterministic scheduler  $\sigma \in \text{ST}$  that always chooses

$\beta$  at  $s_B$ , i.e.,  $\sigma(\pi, \beta) = 1$  for all  $\pi \in FPaths_{\text{rel}}$  with  $\text{last}(\pi) = s_B$ . The probability under  $\sigma$  for the first request to be rejected within one time unit is given by the probability  $\Pr_{\sigma}^{\mathcal{M}}(\Pi_2)$  of the paths

$$\Pi_2 = \{s_w \xrightarrow{t, \perp} s_B \xrightarrow{0, \beta} s_w \mid 0 \leq t \leq 1\}.$$

To compute this probability, we start with the computation of the probabilities of the individual time-stamped steps. The set  $E_1 = \{(t, \perp, s_B) \mid 0 \leq t \leq 1\} \in \mathcal{E}^{\text{Steps}}$  and the history  $s_w$  are considered for the first step. We obtain

$$\begin{aligned} \Pr_{\sigma, s_w}^{\text{Steps}}(E_1) &= \int_0^1 \mathbf{E}(s) \cdot e^{-\mathbf{E}(s)t} \cdot \mathbf{P}(s_w, \perp, s_B) dt \\ &= (1 - e^{-\lambda_A - \lambda_B}) \cdot \frac{\lambda_B}{\lambda_A + \lambda_B}. \end{aligned}$$

For the second step, we consider the set  $E_2 = \{(0, \beta, s_w)\} \in \mathcal{E}^{\text{Steps}}$  and the history  $\pi$  for some fixed  $\pi \in \Pi_1 = \{s_w \xrightarrow{t, \perp} s_B \mid 0 \leq t \leq 1\}$ . It follows that

$$\Pr_{\sigma, \pi}^{\text{Steps}}(E_2) = \underbrace{\sigma(\pi, \beta)}_{=1} \cdot \mathbf{P}(s_B, \beta, s_w) = 0.5.$$

The desired value is obtained by combining the two results which yields

$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\Pi_2) &= \Pr_{\sigma, 2}^{\text{FPaths}}(\Pi_2) \\ &= \Pr_{\sigma, 2}^{\text{FPaths}}(\Pi_1 \circ E_2) \\ &= \int_{\pi \in \Pi_1} \Pr_{\sigma, \pi}^{\text{Steps}}(E_2) d\Pr_{\sigma, 1}^{\text{FPaths}}(\pi) \\ &= \int_{\pi \in \Pi_1} 0.5 d\Pr_{\sigma, 1}^{\text{FPaths}}(\pi) \\ &= 0.5 \cdot \Pr_{\sigma, 1}^{\text{FPaths}}(\Pi_1) \\ &= 0.5 \cdot \Pr_{\sigma, 1}^{\text{FPaths}}(\{s_w\} \circ E_1) \\ &= 0.5 \cdot \Pr_{\sigma, \pi}^{\text{Steps}}(E_1) \cdot \Pr_{\sigma, 0}^{\text{FPaths}}(s_w) \\ &= 0.5 \cdot (1 - e^{-\lambda_A - \lambda_B}) \cdot \frac{\lambda_B}{\lambda_A + \lambda_B}. \quad \blacksquare \end{aligned}$$

### 2.4.5 Zeno Behavior

Our definition of MAs allow that the probability to take infinitely many transitions within finite time might be greater than zero. This phenomenon is called *Zeno behavior*. Formally, we say that a path  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in IPaths_{\text{rel}}$  of an MA  $\mathcal{M}$  is *Zeno* if

$$\sum_{i=0}^{\infty} t(\kappa_i) < \infty.$$

$\mathcal{M}$  exhibits Zeno behavior if there is a scheduler  $\sigma \in \text{GM}$  for which the probability of the set of Zeno paths is positive, i.e.,

$$\Pr_{\sigma}^{\mathcal{M}}(\{\pi \in \text{IPaths} \mid \pi \text{ is Zeno}\}) > 0 .$$

Note that this is the case whenever there is a probabilistic state  $s \in \text{PS}$  and a scheduler  $\sigma \in \text{GM}$  such that the probability to reach a Markovian state from  $s$  under  $\sigma$  is zero.

Zeno paths represent unrealistic system runs where the time stops increasing at some point. Furthermore, there are several inconveniences regarding Zeno paths, e.g., the prefix  $\text{pref}_T(\pi, t)$  of infinite Zeno path  $\pi$  up to time point  $t \in \mathbb{R}_{\geq 0}$  is not defined for a sufficiently large  $t$ . Therefore, MAs exhibiting Zeno behavior will be excluded from our analysis.

We emphasize that this restriction does *not* apply for MDPs as transitions of MDPs are not considered to be taken in zero time (cf. Remark 2.12 on page 17).

### 2.4.6 Non-relaxed Paths

As mentioned in Section 2.4.1, relaxed paths might represent impossible system behavior by, e.g., specifying actions that are not enabled at the current state or depicting positive sojourn times at probabilistic states (violating the maximum progress assumption). Since dealing with such paths is rather inconvenient, we introduce (non-relaxed) paths that only represent feasible system runs.

Let  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in \text{IPaths}_{\text{rel}}$ . To satisfy the maximum progress assumption, a time-stamp  $t(\kappa_i)$  following a probabilistic state  $s_i \in \text{PS}$  has to be zero. On the other hand, an action-stamp  $\alpha(\kappa_i)$  following a Markovian state  $s_i \in \text{MS}$  can only be  $\perp$  since there is no other action enabled at  $s_i$ . It follows that one of the two entries of a stamp  $\kappa_i \in \mathbb{R}_{\geq 0} \times \text{Act}$  can be considered redundant. For non-relaxed paths we therefore consider stamps  $\kappa$  from the set  $\mathbb{R}_{\geq 0} \cup \text{Act}$  and define

$$\alpha(\kappa) = \begin{cases} \kappa & \text{if } \kappa \in \text{Act} \\ \perp & \text{if } \kappa \in \mathbb{R}_{\geq 0} \end{cases} \quad \text{and} \quad t(\kappa) = \begin{cases} 0 & \text{if } \kappa \in \text{Act} \\ \kappa & \text{if } \kappa \in \mathbb{R}_{\geq 0} . \end{cases}$$

#### Definition 2.27 (Infinite Paths)

An *infinite path* of MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  with transition probability function  $\mathbf{P}$  is an infinite sequence  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots$  of states  $s_0, s_1, \dots \in S$  and stamps  $\kappa_0, \kappa_1, \dots \in \mathbb{R}_{\geq 0} \cup \text{Act}$  such that

1.  $\sum_{i=0}^{\infty} t(\kappa_i) = \infty$

and for any  $i \geq 0$  it holds that

2.  $\mathbf{P}(s_i, \alpha(\kappa_i), s_{i+1}) > 0,$

3.  $s_i \in \text{PS}$  implies  $\kappa_i \in \text{Act}$ , and
4.  $s_i \in \text{MS}$  implies  $\kappa_i \in \mathbb{R}_{\geq 0}$ . ■

Condition 1 in the definition above excludes Zeno paths. Condition 2 ensures that any transition depicted by a path can actually be taken (with a positive probability). Conditions 3 and 4 assert that a time-stamp always specifies the relevant information for the current state.

From now on, we only consider paths as in Definition 2.27. The notions for relaxed paths presented in previous sections are adapted in a straightforward way. We denote the sets of finite and infinite paths of MA  $\mathcal{M}$  by  $FPaths^{\mathcal{M}}$  and  $IPaths^{\mathcal{M}}$ , respectively. Note that  $\Pr_{\sigma}^{\mathcal{M}}(IPaths^{\mathcal{M}}) = 1$ , i.e., the probability of relaxed paths that do not correspond to a path in  $IPaths^{\mathcal{M}}$  is zero.

**Example 2.28**

The relaxed path  $\pi$  from Example 2.18 given by

$$\pi = s_w \xrightarrow{3, \perp} s_B \xrightarrow{0, \beta} s_p \xrightarrow{1.4, \perp} s_e \xrightarrow{0.6, \perp} s_e \xrightarrow{1, \perp} \dots$$

is also a path according to Definition 2.27. By omitting the redundant information of the stamps, we obtain

$$\pi = s_w \xrightarrow{3} s_B \xrightarrow{\beta} s_p \xrightarrow{1.4} s_e \xrightarrow{0.6} s_e \xrightarrow{1} \dots \quad \blacksquare$$

## 2.5 Rewards and Costs

MAs can be incorporated with rewards or, equivalently, costs by defining the reward functions  $\rho_1, \dots, \rho_\ell$  of an MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_\ell\})$  accordingly. This enables us to analyze expected values of various measures of the modeled system like, e.g., the energy consumption, the time until a failure, or the number of required retries to deliver a message via a lossy channel. Since there is no formal difference between rewards and costs (costs can be interpreted as “bad rewards”), we restrict ourselves to rewards. The definitions in this section have been adapted from [GTH<sup>+</sup>14].

**Example 2.29**

We add two reward functions  $\rho_1$  and  $\rho_2$  to the MA  $\mathcal{M}$  of the client-server example shown in Figure 2.2 on page 15, i.e., we set  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \rho_2\})$ . The function  $\rho_1$  models the number of incoming requests while  $\rho_2$  models the energy consumption of the server. To this end, it is assumed that the server consumes ten energy units per time unit while processing a request (state  $s_p$ ) and one energy unit per time unit otherwise (states  $s_w$  and  $s_e$ ). Furthermore, every request has a basic cost of four

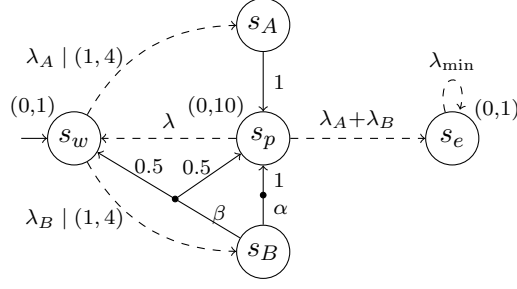


Figure 2.4: MA  $\mathcal{M}$  with reward functions  $\rho_1$  and  $\rho_2$  (cf. Example 2.29).

energy units. For  $\xi \in S \cup (S \times Act)$  we set

$$\rho_1(\xi) = \begin{cases} 1 & \text{if } \xi = (s_w, \perp) \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \rho_2(\xi) = \begin{cases} 10 & \text{if } \xi = s_p \\ 1 & \text{if } \xi \in \{s_w, s_e\} \\ 4 & \text{if } \xi = (s_w, \perp) \\ 0 & \text{otherwise.} \end{cases}$$

Figure 2.4 depicts the MA with the two reward functions. We use tuples  $(x_1, x_2) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$  to denote the values of  $\rho_1$  and  $\rho_2$  ( $x_i$  refers to the value of  $\rho_i$  for  $i \in \{1, 2\}$ ). A tuple next to a state represents state rewards. Tuples next to transition arrows denote action rewards, where we use  $|$  to separate them from rates, actions, or probabilities. Rewards that are not depicted explicitly are assumed to be zero. ■

### Definition 2.30 (Reward of a Path)

Let  $\rho$  be a reward function of an MA  $\mathcal{M}$ . For a path  $\pi' = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n \in FPaths$ , the reward of  $\pi'$  w.r.t.  $\rho$  is given by

$$rew^{\mathcal{M}}(\rho, \pi') = \sum_{i=0}^{|\pi'|-1} \rho(s_i) \cdot t(\kappa_i) + \rho(s_i, \alpha(\kappa_i)).$$

For a path  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in IPaths$ , the reward of  $\pi$  up to a set of goal states  $G \subseteq S$  w.r.t.  $\rho$  is given by

$$rew^{\mathcal{M}}(\rho, \pi, G) = \begin{cases} rew^{\mathcal{M}}(\rho, \text{pref}(\pi, n)) & \text{if } n = \min\{i \geq 0 \mid s_i \in G\} \\ \lim_{n \rightarrow \infty} rew^{\mathcal{M}}(\rho, \text{pref}(\pi, n)) & \text{if } s_i \notin G \text{ for all } i \geq 0. \end{cases} \quad \blacksquare$$

Intuitively, the reward of  $\pi' \in FPaths$  w.r.t. reward function  $\rho$  is the sum over the rewards obtained in every step  $s_i \xrightarrow{\kappa_i}$  depicted in the path. The reward obtained in step  $i$  is composed of the state reward of  $s_i$  multiplied with the sojourn time  $t(\kappa_i)$  as well as the action reward given by  $s_i$  and  $\alpha(\kappa_i)$ . Note that state rewards assigned to probabilistic states do not affect the reward of a path as the sojourn time in such

states is always zero. Consequently, state rewards do not play a role for MDPs and have therefore been omitted in Definition 2.11. Further, note that  $rew^{\mathcal{M}}(\rho, \pi')$  is independent of the reward defined for  $last(\pi')$ .

The intuition of the reward of  $\pi \in IPaths$  up to  $G \subseteq S$  is that we stop measuring the reward as soon as a state in  $G$  is reached. If no state in  $G$  is reached, reward is accumulated along the infinite path which potentially yields an infinite reward. This particularly holds for  $rew^{\mathcal{M}}(\rho, \pi, \emptyset)$  which is also referred to as the *total reward* of the path  $\pi$ .

**Example 2.31**

Consider the MA  $\mathcal{M}$  from Figure 2.4 with reward functions  $\rho_1$  and  $\rho_2$  and the path  $\pi' = s_w \xrightarrow{3} s_B \xrightarrow{0} s_p \xrightarrow{1.4} s_e$  of  $\mathcal{M}$ . The reward of  $\pi'$  w.r.t. the two reward functions is

$$\begin{aligned} rew(\rho_1, \pi') &= \rho_1(s_w, \perp) = 1 \quad \text{and} \\ rew(\rho_2, \pi') &= \rho_2(s_w) \cdot 3 + \rho_2(s_w, \perp) + \rho_2(s_p) \cdot 1.4 = 1 \cdot 3 + 4 + 10 \cdot 1.4 = 21 . \end{aligned}$$

Now consider the extension of  $\pi'$  to an infinite path  $\pi = \pi' \xrightarrow{1} s_e \xrightarrow{1} \dots$ . The reward of  $\pi$  up to  $G = \{s_e\}$  for reward functions  $\rho_i$  with  $i \in \{1, 2\}$  is

$$rew(\rho_i, \pi, G) = rew^{\mathcal{M}}(\rho_i, pref(\pi, 3)) = rew^{\mathcal{M}}(\rho_i, \pi') = \begin{cases} 1 & \text{if } i = 1 \\ 21 & \text{if } i = 2 . \end{cases}$$

For the set  $G' = \emptyset$  we note that  $\pi$  does not visit a state in  $G'$ . Thus, we keep accumulating rewards, yielding the total rewards

$$\begin{aligned} rew(\rho_1, \pi, \emptyset) &= 1 \quad \text{and} \\ rew(\rho_2, \pi, \emptyset) &= 21 + \sum_{i=0}^{\infty} \rho_2(s_e) \cdot 1 = \infty . \end{aligned} \quad \blacksquare$$

For fixed  $\mathcal{M}$ ,  $\rho$ , and  $G$ , the function  $rew^{\mathcal{M}}(\rho, \cdot, G): IPaths \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$  describes a random variable mapping infinite paths of  $\mathcal{M}$  to the corresponding rewards up to  $G$ . The expected reachability reward is the expected value of this random variable.

**Definition 2.32 (Expected Reachability Reward)**

Given an MA  $\mathcal{M}$ , a reward function  $\rho$ , a scheduler  $\sigma \in GM^{\mathcal{M}}$ , and a set of goal states  $G$ , the *expected reachability reward* is given by

$$eR_{\sigma}^{\mathcal{M}}(\rho, G) = \int_{\pi \in IPaths^{\mathcal{M}}} rew^{\mathcal{M}}(\rho, \pi, G) dPr_{\sigma}^{\mathcal{M}}(\pi). \quad \blacksquare$$

The superscript  $\mathcal{M}$  is omitted from the notation whenever  $\mathcal{M}$  is clear from the context. We write  $eR_{\sigma}(\rho)$  for the *expected total reward* which is an abbreviation for  $eR_{\sigma}(\rho, \emptyset)$ .



## Chapter 3

# Multi-objective Model Checking

The previous chapter detailed Markov automata as formalism to model probabilistic systems with nondeterminism and continuous time. The next step is to automatically analyze certain properties of such models which is commonly referred to as *model checking* [Cla08]. Standard model checking considers the properties (we call them *objectives*) individually. For instance, we may compute the maximal expected profit of a system run w.r.t. the possible resolutions of nondeterminism. However, such an analysis is not feasible when we are interested in multiple objectives that should be fulfilled by the same scheduler (e.g., a scheduler that maximizes the expected profit might violate certain safety constraints). *Multi-objective model checking* aims to analyze multiple objectives at once and reveals possible trade-offs .

This chapter discusses the different types of considered objectives (Section 3.1) and formalizes the multi-objective model checking problem comprising *achievability*, *quantitative*<sup>1</sup>, and *Pareto queries* [FKP12] (Section 3.2).

### 3.1 Objectives

We start with an enumeration of the types of objectives that are relevant for the scope of this thesis. Note that these are standard objectives as considered by, e.g., [BK08, FKP12, GHH<sup>+</sup>13, GTH<sup>+</sup>14]. For this and subsequent sections, let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  denote an MA. An objective  $\mathbb{O}_i$  is a representation of a *quantitative* property of  $\mathcal{M}$ , e.g., the probability to reach an error state, or the expected energy consumption.  $\mathbb{O}_i$  can be combined with some threshold  $\triangleright_i p_i$  in order to represent a *qualitative* property expressing, e.g., whether the probability to reach an error state is less than  $p_i$ .

---

<sup>1</sup>In [FKP12], quantitative queries are referred to as *numerical queries*

**Definition 3.1 (Threshold)**

A *threshold* is a pair  $\triangleright_i p_i$  with *threshold relation*  $\triangleright_i \in \{<, \leq, >, \geq\}$  and a *threshold value*  $p_i \in \mathbb{R}$ . ■

The quantitative value  $v_i$  represented by the objective  $\mathbb{O}_i$  is defined for a given scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ . If  $v_i \triangleright_i p_i$ , we say that the objective is *satisfied* w.r.t. the threshold  $\triangleright_i p_i$  and write  $\mathcal{M}, \sigma \models \mathbb{O}_i \triangleright_i p_i$ .

**Probabilistic objectives.** Probabilistic objectives represent the probability of certain events. We consider *until objectives* which for two given sets of states  $H, G \subseteq S$  express the probability that only states in  $H$  are visited until eventually some state in  $G$  is reached. In addition, a time interval is specified which allows us to restrict the possible time points for reaching  $G$ .

**Definition 3.2 (Time Interval)**

For  $a, b \in \mathbb{R}$  such that  $0 \leq a < b$ , a *time interval* is an interval  $I \subseteq \mathbb{R}$  of the form  $I = [a, b]$  or  $I = [a, \infty)$ . ■

Let  $I$  be a time interval. The set  $HU^I G$  is defined as the infinite paths that visit states in  $H$  until a state in  $G$  is reached at some point in  $I$ , i.e.,

$$HU^I G = \{\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in \text{IPaths} \mid \exists n \geq 0: s_n \in G, \forall i < n: s_i \in H, \text{ and } I \cap [t, t + t(\kappa_n)] \neq \emptyset \text{ for } t = T(\text{pref}(\pi, n))\}.$$

We also say that  $\pi \in HU^I G$  reaches  $G$  via  $H$  at some point in  $I$ . Until objectives represent the probabilities for sets of the form  $HU^I G$ .

**Definition 3.3 (Until Objectives)**

Let  $\mathcal{M}$  be an MA with scheduler  $\sigma \in \text{GM}$  and subsets of states  $H$  and  $G$ . Further, let  $I$  be a time interval. An *until objective* has the form  $\mathbb{P}(HU^I G)$  and its satisfaction w.r.t. a threshold  $\triangleright_i p_i$  is defined by

$$\mathcal{M}, \sigma \models \mathbb{P}(HU^I G) \triangleright_i p_i \iff \text{Pr}_\sigma^{\mathcal{M}}(HU^I G) \triangleright_i p_i . \quad \blacksquare$$

The shorthand  $\diamond^I G = SU^I G$  is considered for the paths that reach  $G$  at some point in  $I$ . We refer to the corresponding objectives of the form  $\mathbb{P}(\diamond^I G)$  as *reachability objectives*. Similarly, we define the set  $\square^I G = \text{IPaths} \setminus \diamond^I(S \setminus G)$ , containing the paths that stay in  $G$  during the time period given by  $I$ . Objectives of the form  $\mathbb{P}(\square^I G)$  are also referred to as *invariant objectives*. Note that reachability and invariant objectives can always be replaced by equivalent unbounded until objectives since

$\mathcal{M}, \sigma \models \mathbb{P}(\diamond^I G) \triangleright_i p_i \iff \mathcal{M}, \sigma \models \mathbb{P}(S\mathcal{U}^I G) \triangleright_i p_i$  and

$$\begin{aligned} \mathcal{M}, \sigma \models \mathbb{P}(\square^I G) \triangleright_i p_i &\iff \Pr_\sigma^{\mathcal{M}}(IPaths \setminus \diamond^I(S \setminus G)) \triangleright_i p_i \\ &\iff 1 - \Pr_\sigma^{\mathcal{M}}(\diamond^I(S \setminus G)) \triangleright_i p_i \\ &\iff \Pr_\sigma^{\mathcal{M}}(\diamond^I(S \setminus G)) \triangleleft_i 1 - p_i \\ &\iff \mathcal{M}, \sigma \models \mathbb{P}(S\mathcal{U}^I(S \setminus G)) \triangleleft_i 1 - p_i, \end{aligned}$$

where  $\triangleleft_i \in \{<, \leq, >, \geq\}$  is the relation satisfying  $q_i \triangleleft_i p_i$  iff  $p_i \triangleright_i q_i$ .

A *(time-)unbounded until objective* is an until objective of the form  $\mathbb{P}(H\mathcal{U}^{[0,\infty)} G)$ . On the other hand, a *(time-)bounded until objective*  $\mathbb{P}(H\mathcal{U}^I G)$  always considers an interval  $I \neq [0, \infty)$ . The depiction of time intervals is simplified by writing, e.g.,  $H\mathcal{U}^{\leq b} G$  instead of  $H\mathcal{U}^{[0,b]} G$  or  $H\mathcal{U} G$  instead of  $H\mathcal{U}^{[0,\infty)} G$ . Similar notions hold for reachability objectives  $\mathbb{P}(\diamond^I G)$  and invariant objectives  $\mathbb{P}(\square^I G)$ .

#### Example 3.4

We consider objectives for the MA  $\mathcal{M}$  modeling the client-server example as presented in the previous chapter (cf. Example 2.9 on page 12). The MA is also depicted in Figure 3.1.

- The unbounded until objective  $\mathbb{P}(\{s_w, s_B\}\mathcal{U}\{s_p\})$  represents the probability that  $s_p$  is reached via  $s_w$  and  $s_B$ , i.e., the first processed request originates from Client  $B$ .
- The bounded until objective  $\mathbb{P}(\{s_w, s_B\}\mathcal{U}^{\geq 0.5}\{s_p\})$  represents the probability that the first processed request originates from Client  $B$  and is received after 0.5 time units.
- The unbounded reachability objective  $\mathbb{P}(\diamond\{s_A\})$  represents the probability that any request of Client  $A$  is received.
- The bounded invariant objective  $\mathbb{P}(\square^{[1,2]}\{s_w\})$  represents the probability that there is no request between time points one and two. ■

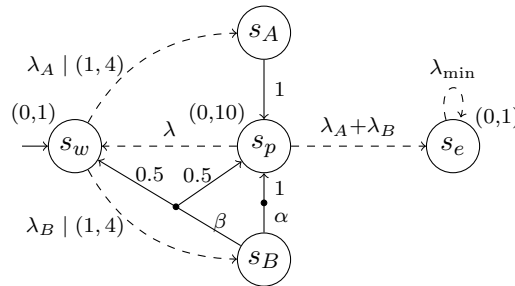


Figure 3.1: MA  $\mathcal{M}$  for the client-server example (cf. Example 3.4 and Example 3.6).

**Expected value objectives.** Secondly, we discuss objectives that represent expected values of the model. More precisely, we introduce objectives that refer to expected reachability rewards (cf. Definition 2.32 on page 29).

**Definition 3.5 (Expected Reachability Reward Objective)**

Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with scheduler  $\sigma \in \text{GM}$  and goal states  $G \subseteq S$ . Further, let  $j \in \{1, \dots, \ell\}$  be an index of a reward function of  $\mathcal{M}$ . An *expected reachability reward objective* has the form  $\mathbb{E}(\#j, G)$  and its satisfaction w.r.t. a threshold  $\triangleright_i p_i$  is defined by

$$\mathcal{M}, \sigma \models \mathbb{E}(\#j, G) \triangleright_i p_i \iff \text{eR}_\sigma^{\mathcal{M}}(\rho_j, G) \triangleright_i p_i . \quad \blacksquare$$

For a reward function  $\rho_j$ , we also define the *expected total reward objective*  $\mathbb{E}(\#j)$  which is an abbreviation for  $\mathbb{E}(\#j, \emptyset)$ , representing the expected value obtained when reward is collected along infinite paths. Furthermore, we consider *expected time objectives*  $\mathbb{E}(T, G)$  which represent the expected time to reach a certain set of goal states  $G \subseteq S$ . The satisfaction w.r.t. a threshold  $\triangleright_i p_i$  is given by

$$\mathcal{M}, \sigma \models \mathbb{E}(T, G) \triangleright_i p_i \iff \text{eR}_\sigma^{\mathcal{M}}(\rho_T, G) \triangleright_i p_i ,$$

where  $\rho_T$  is a reward function for  $\mathcal{M}$  which for each  $\xi \in S \cup (S \times Act)$  satisfies

$$\rho_T(\xi) = \begin{cases} 1 & \text{if } \xi \in \text{MS} \\ 0 & \text{otherwise.} \end{cases}$$

We assume w.l.o.g. that every MA contains such a reward function. Hence, any expected time objective can be replaced by an equivalent expected reachability reward objective.

**Example 3.6**

We continue the list of example objectives for the model  $\mathcal{M}$  of the client-server example depicted in Figure 3.1. Recall from Example 2.29 on page 27 that the reward function  $\rho_1$  of  $\mathcal{M}$  models the number of incoming requests while  $\rho_2$  retrieves the energy consumption.

- The expected reachability reward objective  $\mathbb{E}(\#2, \{s_e\})$  represents the expected energy consumption until an error occurs.
- The expected total reward objective  $\mathbb{E}(\#1)$  represents the expected number of requests that the server processes.
- The expected time objective  $\mathbb{E}(T, \{s_e\})$  represents the expected time until an error occurs. ■

In summary, any objective considered in the scope of this thesis can be expressed as either

- an (unbounded or bounded) until objective  $\mathbb{P}(HU^I G)$ , or
- an expected reachability reward objective  $\mathbb{E}(\#j, G)$ .

We restrict our explanations to these types of objectives.

## 3.2 Multi-objective Queries

A *multi-objective query* is a composition of several objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  for an MA  $\mathcal{M}$ . The number of objectives  $d$  is referred to as the *dimension* of the query. We lift the satisfaction relation  $\models$  (as defined for the individual objectives above) to  $\mathbb{O}$ .

### Definition 3.7 (Satisfaction of Multiple Objectives)

Let  $\mathcal{M}$  be an MA with scheduler  $\sigma \in \text{GM}$ . For a list of objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d) \in \{<, \leq, >, \geq\}^d$  and threshold values  $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{R}^d$ , the relation  $\models$  satisfies

$$\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} \iff \mathcal{M}, \sigma \models \mathbb{O}_i \triangleright_i p_i \text{ for all } 1 \leq i \leq d. \quad \blacksquare$$

If  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p}$ , we also say that point  $\mathbf{p} \in \mathbb{R}^d$  is *achievable* in  $\mathcal{M}$  with scheduler  $\sigma$ .

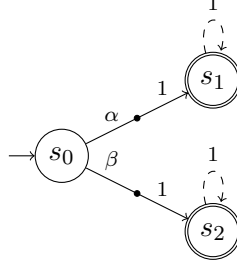
The individual objectives  $\mathbb{O}_1, \dots, \mathbb{O}_d$  are interpreted as either *qualitative* or *quantitative* objectives. For a qualitative objective, the query provides a threshold  $\triangleright_i p_i$  that is to be satisfied. For a quantitative objective, an *optimization direction*  $\text{opt}_i \in \{\max, \min\}$  is given with the interpretation that the represented value should be preferably high (or low). We say that a (qualitative or quantitative) objective  $\mathbb{O}_i$  is *maximizing* if it is associated with either threshold relation  $\triangleright_i \in \{>, \geq\}$  or optimization direction  $\text{opt}_i = \max$ . Consequently, objectives with threshold relation  $\triangleright_i \in \{<, \leq\}$  or optimization direction  $\text{opt}_i = \min$  are *minimizing*. We now list the different types of multi-objective queries as presented in [FKP12]. Each type of query considers a specific combination of qualitative and quantitative objectives.

**Achievability queries.** Achievability queries provide the most basic query type. Given a list of qualitative objectives, the goal is to check whether there is one scheduler for which all objectives are satisfied. More precisely, let  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$  and  $\mathbf{p} = (p_1, \dots, p_d)$  specify thresholds for the objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$ . An achievability query denoted by  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  retrieves whether  $\mathbf{p}$  is achievable, i.e.,

$$\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p}) \iff \mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} \text{ for some } \sigma \in \text{GM}.$$

### Example 3.8

Consider the MA  $\mathcal{M}$  from Figure 3.2 and the qualitative objectives  $\mathbb{O} = (\mathbb{O}_1, \mathbb{O}_2)$ , with objectives  $\mathbb{O}_1 = \mathbb{P}(\diamond\{s_1\})$  and  $\mathbb{O}_2 = \mathbb{P}(\diamond\{s_2\})$ , threshold relations  $\triangleright = \{\geq, \geq\}$ , and

Figure 3.2: MA  $\mathcal{M}$  (cf. Example 3.8 and Example 3.9).

threshold values  $\mathbf{p} = (0.6, 0.7)$ . Notice that the objectives can be satisfied individually: The scheduler  $\sigma_1$  with  $\sigma_1(s_0, \alpha) = 1$  satisfies  $\mathcal{M}, \sigma_1 \models \mathbb{O}_1 \geq 0.6$ . Similarly, we have  $\mathcal{M}, \sigma_2 \models \mathbb{O}_2 \geq 0.7$  for the scheduler  $\sigma_2$  with  $\sigma_2(s_0, \beta) = 1$ . However, there is no scheduler under which both objectives are true, i.e.,  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  does *not* hold. On the other hand,  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{q})$  with  $\mathbf{q} = (0.6, 0.4)$  does hold as  $\mathbf{q}$  is achievable with scheduler  $\sigma_3$  satisfying  $\sigma_3(s_0, \alpha) = 0.6$  and  $\sigma_3(s_0, \beta) = 0.4$ . Notice that  $\mathbf{q}$  is not achievable with a deterministic scheduler. ■

**Quantitative queries.** Quantitative queries specify one quantitative objective  $\mathbb{O}_1$  and any number of qualitative objectives  $\mathbb{O} = (\mathbb{O}_2, \dots, \mathbb{O}_d)$ . The goal is to find a scheduler that satisfies the quantitative objectives and for which the value represented by  $\mathbb{O}_1$  is as high (or low) as possible. We therefore consider an optimization direction  $\text{opt} \in \{\max, \min\}$ , threshold relations  $\triangleright = (\triangleright_2, \dots, \triangleright_d)$ , and threshold values  $\mathbf{p} = (p_2, \dots, p_d) \in \mathbb{R}^{d-1}$ .

Let  $\mathbb{O}' = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  and  $\triangleright' = (\triangleright_1, \dots, \triangleright_d)$ , where  $\triangleright_1 = \geq$  if  $\mathbb{O}_1$  is maximizing and  $\triangleright_1 = \leq$  if  $\mathbb{O}_1$  is minimizing. Further, let  $A_1 = \{p_1 \in \mathbb{R} \mid \text{achieve}^{\mathcal{M}}(\mathbb{O}' \triangleright' (p_1, \dots, p_d))\}$  be the set of values for  $\mathbb{O}_1$  such that  $(p_1, \dots, p_d)$  is achievable. A quantitative query is of the form  $\text{quantitative}^{\mathcal{M}}(\text{opt } \mathbb{O}_1, \mathbb{O} \triangleright \mathbf{p})$  and retrieves the value

$$\text{quantitative}^{\mathcal{M}}(\text{opt } \mathbb{O}_1, \mathbb{O} \triangleright \mathbf{p}) = \begin{cases} \sup A_1 & \text{if } A_1 \neq \emptyset \text{ and } \text{opt} = \max \\ \inf A_1 & \text{if } A_1 \neq \emptyset \text{ and } \text{opt} = \min \\ \text{false} & \text{otherwise.} \end{cases}$$

**Example 3.9**

Consider the MA  $\mathcal{M}$  from Figure 3.2 and the query  $\text{quantitative}^{\mathcal{M}}(\max \mathbb{O}_1, (\mathbb{O}_2) \geq 0.4)$  with  $\mathbb{O}_1 = \mathbb{P}(\diamond\{s_1\})$  and  $\mathbb{O}_2 = \mathbb{P}(\diamond\{s_2\})$ . The solution for the query is 0.6 since the point  $\mathbf{q} = (0.6, 0.4)$  is achievable (as seen in Example 3.8) and every other point  $\mathbf{r} = (r_1, r_2)$  with  $r_1 > 0.6$  and  $r_2 \geq 0.4$  is unachievable. ■

**Pareto queries.** For Pareto queries, we are interested in *all* optimal values w.r.t. *multiple* quantitative objectives<sup>2</sup>. To this end, we consider a list of  $d$  objectives  $\mathbb{O}$  with optimization directions  $\mathbf{opt} \in \{\max, \min\}^d$ . A Pareto query retrieves the *Pareto curve* which is the set of Pareto optimal points.

We write  $\mathbf{p} \triangleright \mathbf{q}$  for  $\mathbf{p} = (p_1, \dots, p_d)$ ,  $\mathbf{q} = (q_1, \dots, q_d)$ , and  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$  iff the relations hold entry-wise, i.e.,  $p_i \triangleright_i q_i$  for each  $i \in \{1, \dots, d\}$ .

**Definition 3.10 (Pareto Optimal Point)**

Let  $\mathcal{M}$  be an MA and  $\mathbb{O}$  be a list of  $d$  quantitative objectives with optimization directions  $\mathbf{opt} = (\text{opt}_1, \dots, \text{opt}_d)$ . We define  $\triangleright_{\mathbf{opt}} = (\triangleright_1, \dots, \triangleright_d)$ , where

$$\triangleright_i = \begin{cases} \geq & \text{if } \text{opt}_i = \max \\ \leq & \text{if } \text{opt}_i = \min \end{cases}$$

for each  $i \in \{1, \dots, d\}$ . A point  $\mathbf{q} \in \mathbb{R}^d$  is called *Pareto optimal* if

1.  $\mathbf{q}$  is achievable, i.e.,  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright_{\mathbf{opt}} \mathbf{q})$  holds and
2. all points  $\mathbf{p} \neq \mathbf{q}$  with  $\mathbf{p} \triangleright_{\mathbf{opt}} \mathbf{q}$  are unachievable, i.e.,  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright_{\mathbf{opt}} \mathbf{p})$  does not hold. ■

Intuitively, a Pareto optimal point is an achievable point such that all superior points (w.r.t. the order given by  $\triangleright_{\mathbf{opt}}$ ) are unachievable. We lift this notion to schedulers and say that  $\sigma \in \text{GM}$  is Pareto optimal iff  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright_{\mathbf{opt}} \mathbf{q}$  for some Pareto optimal point  $\mathbf{q}$ . A Pareto query given by  $\text{pareto}^{\mathcal{M}}(\mathbf{opt} \mathbb{O})$  retrieves the set of points

$$\text{pareto}^{\mathcal{M}}(\mathbf{opt} \mathbb{O}) = \{\mathbf{q} \in \mathbb{R}^d \mid \mathbf{q} \text{ is Pareto optimal}\}.$$

**Set of achievable points.** In our explanations above, the optimization directions of quantitative objectives  $\mathbb{O}_i$  are translated to threshold relations  $\triangleright_i$  such that we have  $\triangleright_i = \geq$  for maximizing and  $\triangleright_i = \leq$  for minimizing objectives. Thus, any type of multi-objective query yields a list of (qualitative or quantitative) objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  and threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . The result for the query can then be inferred from the corresponding set of achievable points.

**Definition 3.11 (Set of Achievable Points)**

For an MA  $\mathcal{M}$ , objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  and threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ , the *set of achievable points* of  $\mathcal{M}$  refers to the set  $A = \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})\}$ . ■

In the following, we often assume that a list of objectives  $\mathbb{O}$  with relations  $\triangleright$  are given and discuss the resulting set of achievable points. The corresponding results carry over to the different types of multi-objective queries as presented in this section.

<sup>2</sup>We follow the notions of [FKP12] and assume that Pareto queries consider only quantitative objectives. An extension to combinations of at least two quantitative and arbitrarily many qualitative objectives can be defined as well.

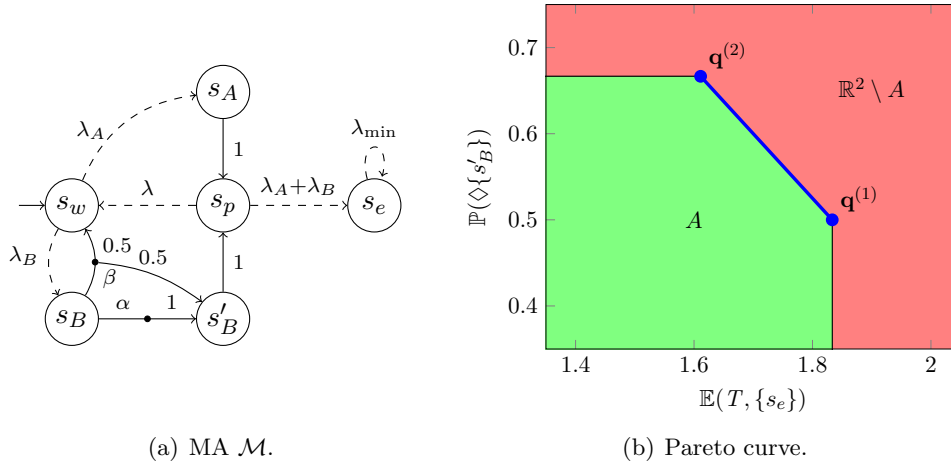


Figure 3.3: MA  $\mathcal{M}$  and Pareto curve (cf. Example 3.12).

### Example 3.12

We consider two objectives for the client-server example:

1. maximize the expected time until an error happens and
2. maximize the probability that at least one request of Client  $B$  is processed.

To specify the second objective, an intermediate probabilistic state  $s'_B$  is added to the model from Figure 3.1, resulting in the MA  $\mathcal{M}$  depicted in Figure 3.3(a). Formally, the objectives are given by  $\mathbb{O} = (\mathbb{E}(T, \{s_e\}), \mathbb{P}(\diamond\{s'_B\}))$  and  $\mathbf{opt} = (\max, \max)$ .

Figure 3.3(b) illustrates the result of the query  $\mathit{pareto}^{\mathcal{M}}(\mathbf{opt} \mathbb{O})$ , where we assume the rates  $\lambda_A = \lambda_B = 1$  for the incoming requests and  $\lambda = 2$  for the processing duration. The Pareto curve is depicted by the thick blue line. The green area below this curve marks the set of achievable points  $A = \{\mathbf{p} \in \mathbb{R}^2 \mid \mathit{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright_{\mathbf{opt}} \mathbf{p})\}$ .

We describe two Pareto optimal schedulers  $\sigma_1$  and  $\sigma_2$ , inducing the points  $\mathbf{q}^{(1)}$  and  $\mathbf{q}^{(2)}$  depicted in Figure 3.3(b). The scheduler  $\sigma_1$  always chooses  $\beta$  at state  $s_B$ . It is Pareto optimal since any other scheduler yields that  $s_p$  is visited more frequently, resulting in a lower expected time to reach  $s_e$ . Under this scheduler, we obtain  $\mathit{eR}_{\sigma_1}^{\mathcal{M}}(\rho_T, s_e) = 11/6$  and  $\Pr_{\sigma_1}^{\mathcal{M}}(\diamond s'_B) = 1/2$ . It follows that  $\mathbf{q}^{(1)} = (11/6, 1/2)$  is Pareto optimal. Similarly, the scheduler  $\sigma_2$  that chooses  $\alpha$  as long as  $s'_B$  has not been reached (and  $\beta$  afterwards) induces the Pareto optimal point  $\mathbf{q}^{(2)} = (29/18, 2/3)$ . The remaining Pareto optimal points lie on the straight line between  $\mathbf{q}^{(1)}$  and  $\mathbf{q}^{(2)}$ . They can be achieved by considering combinations of  $\sigma_1$  and  $\sigma_2$ . ■

## Chapter 4

# Analysis of Markov Automata with Multiple Objectives

The common approach for single-objective model checking of Markov automata (as presented in, e.g., [HH12, GHH<sup>+</sup>13, GTH<sup>+</sup>14]) is to reduce the MA to a simpler structure that can be analyzed with techniques similar to the approaches for MDP model checking. Our goal for this chapter is to lift this approach to the multi-objective case. To this end, assume an MA  $\mathcal{M}$  and a list of objectives  $\mathbb{O}$  with thresholds relations  $\triangleright$ . We discuss how the set of achievable points of  $\mathcal{M}$  relates to the set of achievable points of a simplified model. This chapter presents the following results:

- For unbounded until and expected reachability reward objectives, the achievable points of  $\mathcal{M}$  correspond to the achievable points of the underlying MDP of  $\mathcal{M}$  (cf. Theorem 4.1 on the next page and Theorem 4.9 on page 45).
- If one or more bounded until objectives are considered, a digitization technique can be employed. This yields an MDP  $\mathcal{M}_\delta$  whose set of achievable points represents a sound approximation of the set of achievable points of  $\mathcal{M}$  (cf. Theorem 4.51 on page 81).

These results pave the way for an adaption of techniques for multi-objective MDP model checking to the analysis of MAs.

We start with the treatment of unbounded until objectives (Section 4.1) and expected reachability reward objectives (Section 4.2). Then, we discuss the approximation of the set of achievable points for bounded until objectives (Section 4.3). Finally, our results are extended to combinations of the three objective types (Section 4.4). For the whole chapter, assume an MA  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  with underlying MDP  $\mathcal{M}_\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1^\mathcal{D}, \dots, \rho_\ell^\mathcal{D}\})$ .

## 4.1 Unbounded Until Objectives

Our first goal is to prove the following theorem which implies that a multi-objective query consisting of unbounded until objectives can be answered by analyzing the underlying MDP.

### Theorem 4.1

For MA  $\mathcal{M}$ , list of unbounded until objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$ , threshold relations  $\triangleright$ , and point  $\mathbf{p} \in \mathbb{R}^d$  it holds that

$$\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p}) \iff \text{achieve}^{\mathcal{M}_{\mathcal{D}}}(\mathbb{O} \triangleright \mathbf{p}) \quad (\text{Claim 1})$$

and for any  $\sigma \in \text{TA}$  we have

$$\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} \iff \mathcal{M}_{\mathcal{D}}, \sigma \models \mathbb{O} \triangleright \mathbf{p} . \quad (\text{Claim 2})$$

■

Note that the set of time-abstract schedulers for  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$  coincide, allowing us to omit the superscript, i.e.,  $\text{TA}^{\mathcal{M}} = \text{TA}^{\mathcal{M}_{\mathcal{D}}} = \text{TA}$ . Claim 1 states that the sets of achievable points of  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$  are equal. This allows us to answer a multi-objective query for  $\mathcal{M}$  by conducting the corresponding analysis on  $\mathcal{M}_{\mathcal{D}}$ . Claim 2 enables scheduler synthesis for MAs as we infer that a scheduler  $\sigma$  for  $\mathcal{M}_{\mathcal{D}}$  satisfying the given objectives also satisfies them when applied to  $\mathcal{M}$ . The correctness of the theorem is not obvious: A point  $\mathbf{p} \in \mathbb{R}^d$  might be achievable in  $\mathcal{M}$  by considering a scheduler that depends on the time-stamps of the given path. For such cases, we have to show that there is also a time-abstract scheduler for which  $\mathbf{p}$  is achievable in  $\mathcal{M}_{\mathcal{D}}$ .

### Example 4.2

Consider the MA  $\mathcal{M}$  with underlying MDP  $\mathcal{M}_{\mathcal{D}}$  shown in Figure 4.1 as well as the objectives  $\mathbb{O} = (\mathbb{P}(\diamond\{s_3\}), \mathbb{P}(\diamond\{s_3\}))$  with threshold relations  $\triangleright = (\geq, \leq)$  and point  $\mathbf{p} = (0.6, 0.7)$ . Note that  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  (or  $\text{achieve}^{\mathcal{M}_{\mathcal{D}}}(\mathbb{O} \triangleright \mathbf{p})$ ) holds iff there is a scheduler under which the probability to reach  $s_3$  lies in the interval  $[0.6, 0.7]$ .

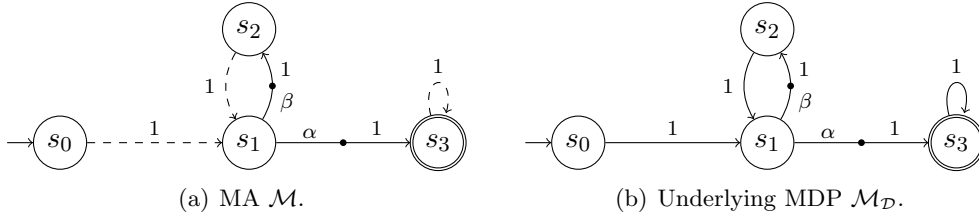
For a path  $\pi \in \text{FPaths}^{\mathcal{M}}$  with  $|\pi| > 0$ , let  $T_{s_0}(\pi)$  denote the sojourn time of  $\pi$  in the initial state  $s_0$ . We uniquely define the scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  such that

$$\sigma(\pi, \alpha) = \begin{cases} 1 & \text{if } T_{s_0}(\pi) \leq 1 \\ 0 & \text{if } T_{s_0}(\pi) > 1 \end{cases}$$

for each  $\pi \in \text{FPaths}^{\mathcal{M}}$  with  $\text{last}(\pi) = s_1$ . We observe that under this scheduler there are two cases: Either action  $\alpha$  is performed at the first visit of  $s_1$  or the scheduler uniformly chooses  $\beta$  every time  $s_1$  is visited. The decision for one of the two cases depends on the sojourn time at  $s_0$ .

The probability to reach  $s_3$  coincides with the probability that  $\alpha$  is performed, i.e., the probability that  $s_0$  is left within one time unit. More precisely, we have

$$\Pr_{\sigma}^{\mathcal{M}}(\diamond s_3) = 1 - e^{-E(s_0) \cdot 1} = 1 - e^{-1} \approx 0.6321 \in [0.6, 0.7].$$

Figure 4.1: MA  $\mathcal{M}$  with underlying MDP  $\mathcal{M}_{\mathcal{D}}$  (cf. Example 4.2).

It follows that  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  holds (with scheduler  $\sigma$ ). Note that  $\sigma$  is not time-abstract and thus can not be applied to the MDP  $\mathcal{M}_{\mathcal{D}}$ . ■

The main idea for the proof of Theorem 4.1 is to construct for any scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  a time-abstract scheduler  $\text{ta}(\sigma) \in \text{TA}$  such that both schedulers induce the same unbounded until probabilities. To this end, we discuss the connection between probabilities of paths of  $\mathcal{M}$  and paths of  $\mathcal{M}_{\mathcal{D}}$ .

**Definition 4.3 (Induced Paths of a Time-abstract Path)**

Let  $\mathcal{M}$  be an MA. For some time-abstract path  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$ , the set of *induced paths* of  $\hat{\pi}$  is given by

$$\langle \hat{\pi} \rangle = \text{ta}^{-1}(\hat{\pi}) = \{ \pi \in \text{FPaths}^{\mathcal{M}} \mid \text{ta}(\pi) = \hat{\pi} \}. \quad \blacksquare$$

**Example 4.4**

Let  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$  be as in Figure 4.1. For  $\hat{\pi} = s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3 \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  we have

$$\langle \hat{\pi} \rangle = \{ s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \in \text{FPaths}^{\mathcal{M}} \mid t \geq 0 \}. \quad \blacksquare$$

Let us fix some  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  and a scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ . The set  $\langle \hat{\pi} \rangle$  contains all paths of  $\mathcal{M}$  that correspond to  $\hat{\pi}$  when replacing the occurring time-stamps by  $\perp$ . The probability distribution  $\sigma(\pi, \cdot) \in \text{Dist}(\text{Act})$  might depend on the time-stamps of the given path  $\pi$ . The idea is to encode these dependencies for each  $\pi \in \langle \hat{\pi} \rangle$  within the probability distribution  $\text{ta}(\sigma)(\hat{\pi}, \cdot)$ , where  $\text{ta}(\sigma)$  is the *time-abstract* of  $\sigma$ .

**Definition 4.5 (Time-abstract of a Scheduler)**

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA. The time-abstract of scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  is defined as  $\text{ta}(\sigma) \in \text{TA}$  such that

$$\text{ta}(\sigma)(\hat{\pi}, \alpha) = \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \text{dPr}_{\sigma}^{\mathcal{M}}(\pi \mid \langle \hat{\pi} \rangle)$$

for  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  and  $\alpha \in \text{Act}$ . ■

The term  $\Pr_\sigma^{\mathcal{M}}(\pi \mid \langle \hat{\pi} \rangle)$  denotes a conditional probability (cf. Definition 2.4 on page 8). Intuitively, it represents the probability for a path in  $\langle \hat{\pi} \rangle$  to have time-stamps as given by  $\pi$ . The value  $\text{ta}(\sigma)(\hat{\pi}, \alpha)$  coincides with the probability that scheduler  $\sigma$  picks action  $\alpha$ , given that the time-abstract path  $\hat{\pi}$  has been observed.

**Example 4.6**

We consider the time-abstraction  $\text{ta}(\sigma)$  of the scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  from Example 4.2. Let  $\hat{\pi} = s_0 \xrightarrow{\perp} s_1$ . Since  $\Pr_\sigma^{\mathcal{M}}(\langle \hat{\pi} \rangle) = \mathbf{P}(s_0, \perp, s_1) = 1$ , the conditional probability in the definition of  $\text{ta}(\sigma)$  can be simplified which yields

$$\begin{aligned} \text{ta}(\sigma)(\hat{\pi}, \alpha) &= \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \, d\Pr_\sigma^{\mathcal{M}}(\pi \mid \langle \hat{\pi} \rangle) \\ &= \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \, d\Pr_\sigma^{\mathcal{M}}(\pi) \\ &= \int_0^\infty \sigma(s_0 \xrightarrow{t} s_1, \alpha) \cdot \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} \, dt \\ &= \int_0^1 \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} \, dt = 1 - e^{-1} . \end{aligned}$$

Note that this value coincides with the probability that  $\sigma$  eventually chooses  $\alpha$  (as computed in Example 4.2).

Now assume a path  $\hat{\pi}_m \in \text{FPaths}^{\mathcal{M}\mathcal{D}}$  that ends at the  $(m+1)$ -th visit of  $s_1$ , i.e.  $\hat{\pi}_m = s_0 \xrightarrow{\perp} (s_1 \xrightarrow{\beta} s_2 \xrightarrow{\perp})^m s_1$ . Let  $m > 0$ . If a path  $\pi \in \langle \hat{\pi}_m \rangle \subseteq \text{FPaths}^{\mathcal{M}}$  is observed,  $\beta$  has been chosen at least once. Recall from Example 4.2 that in this case,  $\sigma$  uniformly chooses  $\beta$  every time  $s_1$  is visited, i.e.,  $\alpha$  will never be performed. Let  $T_{s_0}(\pi)$  denote the sojourn time of  $\pi \in \langle \hat{\pi}_m \rangle$  in state  $s_0$ . We have

$$\begin{aligned} \text{ta}(\sigma)(\hat{\pi}_m, \alpha) &= \int_{\pi \in \langle \hat{\pi}_m \rangle} \sigma(\pi, \alpha) \, d\Pr_\sigma^{\mathcal{M}}(\pi \mid \langle \hat{\pi}_m \rangle) \\ &= \int_{\substack{\pi \in \langle \hat{\pi}_m \rangle \\ T_{s_0}(\pi) \leq 1}} \sigma(\pi, \alpha) \underbrace{d\Pr_\sigma^{\mathcal{M}}(\pi \mid \langle \hat{\pi}_m \rangle)}_{=0 \text{ (as } \beta \text{ occurs in } \pi)} + \int_{\substack{\pi \in \langle \hat{\pi}_m \rangle \\ T_{s_0}(\pi) > 1}} \underbrace{\sigma(\pi, \alpha)}_{=0} \, d\Pr_\sigma^{\mathcal{M}}(\pi \mid \langle \hat{\pi}_m \rangle) \\ &= 0 . \end{aligned}$$

Hence, the probability to eventually perform  $\alpha$  under scheduler  $\text{ta}(\sigma)$  is the probability to perform  $\alpha$  at the first visit of  $s_1$ , i.e.,  $1 - e^{-1}$ . With our observations from Example 4.2 it follows that

$$\Pr_\sigma^{\mathcal{M}}(\diamond s_3) = \Pr_{\text{ta}(\sigma)}^{\mathcal{M}\mathcal{D}}(\diamond s_3) = 1 - e^{-1} . \quad \blacksquare$$

In the example above, we have seen that the considered scheduler and its time-abstraction induce the same reachability probability. We now generalize this observation. First, it is shown that  $\sigma \in \text{GM}^{\mathcal{M}}$  and  $\text{ta}(\sigma) \in \text{TA}$  induce the same probabilities

for a given finite time-abstract path. In a second step, this result is lifted to unbounded until probabilities which yields the connection between the achievable points of  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$ .

**Lemma 4.7**

For MA  $\mathcal{M}$ , scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ , and time-abstract path  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  it holds that

$$\Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) = \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}). \quad \blacksquare$$

*Proof.* The proof is by induction over the length of the considered path  $|\hat{\pi}| = n$ . Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_{\ell}\})$  and  $\mathcal{M}_{\mathcal{D}} = (S, \text{Act}, \mathbf{P}, s_0, \{\rho_1^{\mathcal{D}}, \dots, \rho_{\ell}^{\mathcal{D}}\})$ . If  $n = 0$ , then  $\{\hat{\pi}\} = \langle \hat{\pi} \rangle = \{s_0\}$ . Hence,  $\Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) = 1 = \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi})$ . In the induction step, we assume that the lemma holds for a fixed path  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  with length  $|\hat{\pi}| = n$  and  $\text{last}(\hat{\pi}) = s$ . Consider the path  $\hat{\pi} \xrightarrow{\alpha} s' \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$ .

**Case  $s \in \text{PS}$ :** It follows that

$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \xrightarrow{\alpha} s' \rangle) &= \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \cdot \mathbf{P}(s, \alpha, s') \, d\Pr_{\sigma}^{\mathcal{M}}(\pi) \\ &= \mathbf{P}(s, \alpha, s') \cdot \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \, d\Pr_{\sigma}^{\mathcal{M}}(\{\pi\} \cap \langle \hat{\pi} \rangle) \\ &= \mathbf{P}(s, \alpha, s') \cdot \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \, d[\Pr_{\sigma}^{\mathcal{M}}(\pi \mid \langle \hat{\pi} \rangle) \cdot \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle)] \\ &= \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \cdot \mathbf{P}(s, \alpha, s') \cdot \int_{\pi \in \langle \hat{\pi} \rangle} \sigma(\pi, \alpha) \, d\Pr_{\sigma}^{\mathcal{M}}(\pi \mid \langle \hat{\pi} \rangle) \\ &= \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \cdot \mathbf{P}(s, \alpha, s') \cdot \text{ta}(\sigma)(\hat{\pi}, \alpha) \\ &\stackrel{IH}{=} \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \cdot \mathbf{P}(s, \alpha, s') \cdot \text{ta}(\sigma)(\hat{\pi}, \alpha) \\ &= \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi} \xrightarrow{\alpha} s'). \end{aligned}$$

**Case  $s \in \text{MS}$ :** According to our restrictions for paths we have  $\alpha = \perp$  and it follows

$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \xrightarrow{\perp} s' \rangle) &= \int_{\pi \in \langle \hat{\pi} \rangle} \int_0^{\infty} \mathbf{E}(s) \cdot e^{-\mathbf{E}(s)t} \cdot \mathbf{P}(s, \perp, s') \, dt \, d\Pr_{\sigma}^{\mathcal{M}}(\pi) \\ &= \mathbf{P}(s, \perp, s') \cdot \int_{\pi \in \langle \hat{\pi} \rangle} \int_0^{\infty} \mathbf{E}(s) \cdot e^{-\mathbf{E}(s)t} \, dt \, d\Pr_{\sigma}^{\mathcal{M}}(\pi) \\ &= \mathbf{P}(s, \perp, s') \cdot \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \\ &\stackrel{IH}{=} \mathbf{P}(s, \perp, s') \cdot \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \\ &= \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi} \xrightarrow{\perp} s'). \quad \square \end{aligned}$$

Lemma 4.7 can be employed to show that  $\sigma$  and  $\text{ta}(\sigma)$  induce the same unbounded until probabilities.

**Proposition 4.8**

For MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  with  $H, G \subseteq S$  and scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  it holds that

$$\Pr_{\sigma}^{\mathcal{M}}(H \mathcal{U} G) = \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(H \mathcal{U} G). \quad \blacksquare$$

*Proof.* Let  $\Pi$  be the set of finite time-abstract paths of  $\mathcal{M}_{\mathcal{D}}$  that end at the first visit of a state in  $G$  and for which the remaining states are in  $H$ , i.e.,

$$\Pi = \{s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{n-1}} s_n \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}} \mid s_n \in G \text{ and } \forall i < n: s_i \in H \setminus G\}.$$

Every path  $\pi \in H \mathcal{U} G \subseteq \text{IPaths}^{\mathcal{M}}$  has a unique prefix  $\pi'$  with  $\text{ta}(\pi') \in \Pi$ . Recall the definition of the cylinder of a set of finite paths (cf. Definition 2.21 on page 21). We have

$$H \mathcal{U} G = \bigcup_{\hat{\pi} \in \Pi} \text{Cyl}(\langle \hat{\pi} \rangle).$$

The claim follows since

$$\Pr_{\sigma}^{\mathcal{M}}(H \mathcal{U} G) = \sum_{\hat{\pi} \in \Pi} \Pr_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \stackrel{\text{Lem. 4.7}}{=} \sum_{\hat{\pi} \in \Pi} \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) = \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(H \mathcal{U} G). \quad \square$$

As the definition of  $\text{ta}(\sigma)$  is independent of the considered objective  $\mathbb{P}(H \mathcal{U} G)$ , Proposition 4.8 can be lifted to lists of unbounded until objectives. This is the basis for the proof of the main result of this section.

*Proof of Theorem 4.1.* Let  $\mathbb{O} = (\mathbb{P}(H_1 \mathcal{U} G_1), \dots, \mathbb{P}(H_d \mathcal{U} G_d))$  be the considered list of objectives with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . The following equivalences hold for any  $\sigma \in \text{GM}^{\mathcal{M}}$  and  $\mathbf{p} \in \mathbb{R}^d$ .

$$\begin{aligned} \mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} &\iff \forall i: \mathcal{M}, \sigma \models \mathbb{P}(H_i \mathcal{U} G_i) \triangleright_i p_i \\ &\iff \forall i: \Pr_{\sigma}^{\mathcal{M}}(H_i \mathcal{U} G_i) \triangleright_i p_i \\ &\stackrel{\text{Prop. 4.8}}{\iff} \forall i: \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(H_i \mathcal{U} G_i) \triangleright_i p_i \\ &\iff \forall i: \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{P}(H_i \mathcal{U} G_i) \triangleright_i p_i \\ &\iff \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O} \triangleright \mathbf{p}, \end{aligned}$$

For Claim 1, assume that  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  holds, i.e., there is a  $\sigma \in \text{GM}^{\mathcal{M}}$  such that  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p}$ . It follows that  $\mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O} \triangleright \mathbf{p}$  which means that  $\text{achieve}^{\mathcal{M}_{\mathcal{D}}}(\mathbb{O} \triangleright \mathbf{p})$  holds as well. For the other direction assume  $\text{achieve}^{\mathcal{M}_{\mathcal{D}}}(\mathbb{O} \triangleright \mathbf{p})$ , i.e.,  $\mathcal{M}_{\mathcal{D}}, \sigma \models \mathbb{O} \triangleright \mathbf{p}$  for some time-abstract scheduler  $\sigma \in \text{TA}$ . We have  $\text{ta}(\sigma) = \sigma$ . It follows that  $\mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O} \triangleright \mathbf{p}$ . Applying the equivalences above yields  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p}$  and thus  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$ .

Claim 2 follows immediately with the equivalences above and the observation  $\text{ta}(\sigma) = \sigma$  for any  $\sigma \in \text{TA}$ .  $\square$

## 4.2 Expected Reachability Reward Objectives

Next, we focus on expected reachability reward objectives. The results are very similar to unbounded until objectives since we also obtain that an analysis of the underlying MDP suffices. More precisely, we show the following extension of Theorem 4.1 to expected reachability reward objectives.

### Theorem 4.9

Let  $\mathcal{M}$  be an MA and let  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  be a list of unbounded until and expected reachability reward objectives with threshold relations  $\triangleright$ . For every point  $\mathbf{p} \in \mathbb{R}^d$  it holds that

$$\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p}) \iff \text{achieve}^{\mathcal{M}_{\mathcal{D}}}(\mathbb{O} \triangleright \mathbf{p})$$

and for any  $\sigma \in \text{TA}$  we have

$$\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} \iff \mathcal{M}_{\mathcal{D}}, \sigma \models \mathbb{O} \triangleright \mathbf{p} . \quad \blacksquare$$

For the proof of the theorem, we show that a scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  and its time-abstraction  $\text{ta}(\sigma) \in \text{TA}$  induce the same expected reachability rewards on MA  $\mathcal{M}$  and MDP  $\mathcal{M}_{\mathcal{D}}$ , respectively. Although this claim is very similar to Proposition 4.8 for unbounded until probabilities, the proof can not be adapted straightforwardly. In particular, the analogon to Lemma 4.7 does not hold: Let  $\rho$  be a reward function for  $\mathcal{M}$  and let  $\rho^{\mathcal{D}}$  be the corresponding counterpart for  $\mathcal{M}_{\mathcal{D}}$ . The expected reward that is collected along a single time-abstract path  $\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}}$  does in general not coincide for  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$ , i.e.,

$$\int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\rho, \pi) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \neq \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \hat{\pi}) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}).$$

The following example illustrates why this is the case.

### Example 4.10

We extend the MA from Figure 4.1 on page 41 to the MA  $\mathcal{M}$  depicted in Figure 4.2(a) by adding a reward function  $\rho$  such that  $\rho(s_0) = 1$  and all other rewards are zero. For the corresponding reward function  $\rho^{\mathcal{D}}$  of the underlying MDP  $\mathcal{M}_{\mathcal{D}}$  (depicted in Figure 4.2(b)) it follows that  $\rho^{\mathcal{D}}(s_0, \perp) = 1/E(s_0) = 1$ . Furthermore, reconsider the scheduler  $\sigma$  from Example 4.2 and its time-abstraction  $\text{ta}(\sigma)$  discussed in Example 4.6 which satisfy

$$\sigma(s_0 \xrightarrow{t} s_1, \alpha) = \begin{cases} 1 & \text{if } t \leq 1 \\ 0 & \text{if } t > 1 \end{cases} \quad \text{and} \quad \text{ta}(\sigma)(s_0 \xrightarrow{\perp} s_1, \alpha) = 1 - e^{-1} .$$

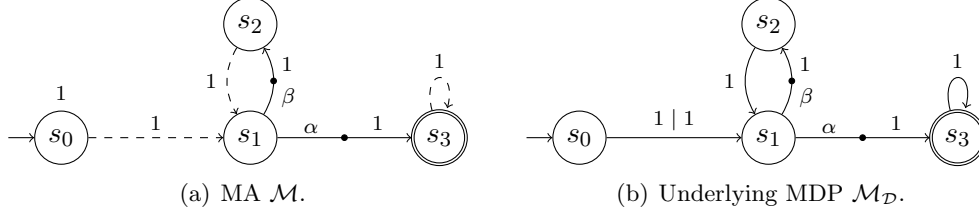


Figure 4.2: MA  $\mathcal{M}$  with reward function  $\rho$  and underlying MDP  $\mathcal{M}_{\mathcal{D}}$  (cf. Example 4.10).

Let  $\hat{\pi}_{\alpha} = s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3$ . The probability  $\Pr_{\sigma}^{\mathcal{M}}(\{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \in \langle \hat{\pi}_{\alpha} \rangle \mid t > 1\})$  is zero since  $\sigma$  chooses  $\beta$  on such paths. For the remaining paths in  $\langle \hat{\pi}_{\alpha} \rangle$ , action  $\alpha$  is chosen with probability one. Hence, the expected reward collected in  $\mathcal{M}$  along  $\hat{\pi}_{\alpha}$  is given by

$$\begin{aligned} \int_{\pi \in \langle \hat{\pi}_{\alpha} \rangle} \text{rew}^{\mathcal{M}}(\rho, \pi) d\Pr_{\sigma}^{\mathcal{M}}(\pi) &= \int_0^1 \rho(s_0) \cdot t \cdot \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} dt \\ &= \frac{\rho(s_0)}{\mathbf{E}(s_0)} \cdot \left(1 - (\mathbf{E}(s_0) + 1) \cdot e^{-\mathbf{E}(s_0)}\right) = 1 - 2e^{-1}. \end{aligned}$$

The expected reward collected in  $\mathcal{M}_{\mathcal{D}}$  along  $\hat{\pi}_{\alpha}$  differs from this value as it is given by

$$\text{rew}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \hat{\pi}_{\alpha}) \cdot \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}_{\alpha}) = \rho^{\mathcal{D}}(s_0, \perp) \cdot \text{ta}(\sigma)(s_0 \xrightarrow{\perp} s_1, \alpha) = 1 - e^{-1}.$$

The intuition is as follows: If some path  $s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_2$  of  $\mathcal{M}$  under  $\sigma$  is observed, we infer that  $t \leq 1$  (since  $\sigma$  chose  $\alpha$ ). Hence, the reward collected from paths in  $\langle \hat{\pi}_{\alpha} \rangle$  is at most one. We conclude that there is a dependency between the choice of the scheduler at  $s_1$  and the collected reward at  $s_0$ . Note that such a conclusion can not be made for  $\mathcal{M}_{\mathcal{D}}$  as the collected reward at some state is independent of the performed action at subsequent states.

Now consider the other time-abstract path of length two, i.e.,  $\hat{\pi}_{\beta} = s_0 \xrightarrow{\perp} s_1 \xrightarrow{\beta} s_2$ . We obtain the expected reward collected along  $\hat{\pi}_{\beta}$  with a similar calculation as for  $\hat{\pi}_{\alpha}$ , yielding  $2e^{-1}$  for  $\mathcal{M}$  and  $e^{-1}$  for  $\mathcal{M}_{\mathcal{D}}$ . Since the rewards for  $\hat{\pi}_{\alpha}$  and  $\hat{\pi}_{\beta}$  sum up to one in both cases, the expected reward collected along all paths of length two coincides for  $\mathcal{M}$  and  $\mathcal{M}_{\mathcal{D}}$ . ■

In the example, the set of all paths of length  $n = 2$  induce the same expected reward for the considered MA and its underlying MDP. We show that this observation can be generalized to arbitrary models and path lengths.

Let  $n \geq 0$  and  $G \subseteq S$ . The set of time abstract paths that end after  $n$  steps or at the first visit of a state in  $G$  is denoted by

$$\begin{aligned} \Pi_G^n = \{s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{m-1}} s_m \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}} \mid (m = n \text{ or } s_m \in G) \text{ and} \\ s_i \notin G \text{ for all } 0 \leq i < m\}. \end{aligned}$$

For  $\mathcal{M}$  under  $\sigma \in \text{GM}^{\mathcal{M}}$  and  $\mathcal{M}_{\mathcal{D}}$  under  $\text{ta}(\sigma) \in \text{TA}$ , we define the expected reward collected along the paths of  $\Pi_G^n$  as

$$\begin{aligned} \text{eR}_{\sigma}^{\mathcal{M}}(\rho, \Pi_G^n) &= \sum_{\hat{\pi} \in \Pi_G^n} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\rho, \pi) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) \text{ and} \\ \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \Pi_G^n) &= \sum_{\hat{\pi} \in \Pi_G^n} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \hat{\pi}) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}), \end{aligned}$$

respectively. Intuitively,  $\text{eR}_{\sigma}^{\mathcal{M}}(\rho, \Pi_G^n)$  corresponds to the expected reachability reward  $\text{eR}_{\sigma}^{\mathcal{M}}(\rho, G)$  assuming that no more reward is collected after the  $n$ -th transition. It follows that the value  $\text{eR}_{\sigma}^{\mathcal{M}}(\rho, \Pi_G^n)$  approaches  $\text{eR}_{\sigma}^{\mathcal{M}}(\rho, G)$  for large  $n$ . Similar notions hold for  $\text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \Pi_G^n)$ . This observation is formalized by the following lemma.

**Lemma 4.11**

For MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_{\ell}\})$  with  $G \subseteq S$ ,  $\sigma \in \text{GM}$ , and reward function  $\rho$  it holds that

$$\lim_{n \rightarrow \infty} \text{eR}_{\sigma}^{\mathcal{M}}(\rho, \Pi_G^n) = \text{eR}_{\sigma}^{\mathcal{M}}(\rho, G). \quad (\text{Claim 3})$$

Furthermore, any reward function  $\rho^{\mathcal{D}}$  for  $\mathcal{M}_{\mathcal{D}}$  satisfies

$$\lim_{n \rightarrow \infty} \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \Pi_G^n) = \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, G). \quad (\text{Claim 4})$$

*Proof.* We show Claim 3. Claim 4 follows analogously. For each  $n \geq 0$ , consider the function  $f_n: \text{IPaths}^{\mathcal{M}} \rightarrow \mathbb{R}_{\geq 0}$  given by

$$f_n(\pi) = \begin{cases} \text{rew}^{\mathcal{M}}(\rho, \text{pref}(\pi, m)) & \text{if } m = \min \{i \in \{0, \dots, n\} \mid s_i \in G\} \\ \text{rew}^{\mathcal{M}}(\rho, \text{pref}(\pi, n)) & \text{if } s_i \notin G \text{ for all } i \leq n. \end{cases}$$

for every path  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots \in \text{IPaths}^{\mathcal{M}}$ . Intuitively,  $f_n(\pi)$  is the reward collected on  $\pi$  within the first  $n$  steps and only up to the first visit of  $G$ . This allows us to express the expected reward collected along the paths of  $\Pi_G^n$  as follows.

$$\text{eR}_{\sigma}^{\mathcal{M}}(\Pi_G^n) = \sum_{\hat{\pi} \in \Pi_G^n} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\rho, \pi) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) = \int_{\pi \in \text{IPaths}^{\mathcal{M}}} f_n(\pi) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) \quad (4.1)$$

It holds that  $\lim_{n \rightarrow \infty} f_n(\pi) = \text{rew}^{\mathcal{M}}(\rho, \pi, G)$  which is a direct consequence from the definition of the reward of  $\pi$  up to  $G$  (cf. Definition 2.30 on page 28). Furthermore, note that the sequence of functions  $f_0, f_1, \dots$  is non-decreasing, i.e., we have  $f_n(\pi) \leq f_{n+1}(\pi)$  for all  $n \geq 0$  and  $\pi \in \text{IPaths}^{\mathcal{M}}$ . By applying the *monotone convergence*

theorem [ADD00] we obtain

$$\begin{aligned}
\lim_{n \rightarrow \infty} \text{eR}_\sigma^{\mathcal{M}}(\Pi_G^n) &\stackrel{4.1}{=} \lim_{n \rightarrow \infty} \int_{\pi \in \text{IPaths}^{\mathcal{M}}} f_n(\pi) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) \\
&= \int_{\pi \in \text{IPaths}^{\mathcal{M}}} \lim_{n \rightarrow \infty} f_n(\pi) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) \\
&= \int_{\pi \in \text{IPaths}^{\mathcal{M}}} \text{rew}^{\mathcal{M}}(\rho, \pi, G) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) = \text{eR}_\sigma^{\mathcal{M}}(\rho, G). \quad \square
\end{aligned}$$

The next step is to show that the expected reward collected along the paths of  $\Pi_G^n$  coincides for  $\mathcal{M}$  under  $\sigma$  and  $\mathcal{M}_{\mathcal{D}}$  under  $\text{ta}(\sigma)$ .

**Lemma 4.12**

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $G \subseteq S$  and  $\sigma \in \text{GM}$ . Furthermore, let  $\rho$  be some reward function of  $\mathcal{M}$  and let  $\rho^{\mathcal{D}}$  be its counterpart for  $\mathcal{M}_{\mathcal{D}}$ . For all  $n \geq 0$  it holds that

$$\text{eR}_\sigma^{\mathcal{M}}(\rho, \Pi_G^n) = \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \Pi_G^n). \quad \blacksquare$$

*Proof.* The proof is by induction over the maximal path length  $n$ . To simplify the notation, we often omit the reward functions  $\rho$  and  $\rho^{\mathcal{D}}$  and write, e.g.,  $\text{rew}^{\mathcal{M}_{\mathcal{D}}}(\pi)$  instead of  $\text{rew}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \pi)$  or  $\text{eR}_\sigma^{\mathcal{M}}(\Pi_G^n)$  instead of  $\text{eR}_\sigma^{\mathcal{M}}(\rho, \Pi_G^n)$ .

If  $n = 0$ , then  $\Pi_G^n = \{s_0\}$ . The claim holds since  $\text{rew}^{\mathcal{M}}(s_0) = \text{rew}^{\mathcal{M}_{\mathcal{D}}}(s_0) = 0$ .

In the induction step, we assume that the lemma is true for some fixed  $n \geq 0$ . We split the term  $\text{eR}_\sigma^{\mathcal{M}}(\Pi_G^{n+1})$  into the reward that is obtained by paths which reach  $G$  within  $n$  steps and the reward obtained by paths of length  $n + 1$ . In a second step, we consider the sum of the reward collected within the first  $n$  steps and the reward obtained in the  $(n + 1)$ -th step:

$$\begin{aligned}
\text{eR}_\sigma^{\mathcal{M}}(\Pi_G^{n+1}) &= \sum_{\substack{\hat{\pi} \in \Pi_G^{n+1} \\ |\hat{\pi}| \leq n}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\pi) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) \\
&\quad + \sum_{\substack{\hat{\pi} \in \Pi_G^{n+1} \\ |\hat{\pi}| = n+1}} \int_{\substack{\pi = \pi' \xrightarrow{\kappa} s' \in \langle \hat{\pi} \rangle \\ \text{last}(\pi') = s}} \text{rew}^{\mathcal{M}}(\pi') + \rho(s) \cdot t(\kappa) + \rho(s, \alpha(\kappa)) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) \\
&= \sum_{\hat{\pi} \in \Pi_G^{n+1}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\text{pref}(\pi, n)) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi) \tag{4.2}
\end{aligned}$$

$$\begin{aligned}
&\quad + \sum_{\substack{\hat{\pi} \in \Pi_G^{n+1} \\ |\hat{\pi}| = n+1}} \int_{\substack{\pi = \pi' \xrightarrow{\kappa} s' \in \langle \hat{\pi} \rangle \\ \text{last}(\pi') = s}} \rho(s) \cdot t(\kappa) + \rho(s, \alpha(\kappa)) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi), \tag{4.3}
\end{aligned}$$

where we define  $\text{pref}(\pi, n)$  for paths with  $|\pi| \leq n$  such that  $\text{pref}(\pi, n) = \pi$ . The two terms 4.2 and 4.3 above are treated separately.

**Term 4.2:** Let  $\Lambda_G^{\leq n} = \{\hat{\pi} \in \Pi_G^{n+1} \mid |\hat{\pi}| \leq n\}$  be the paths in  $\Pi_G^{n+1}$  of length at most  $n$ . We have  $\Lambda_G^{\leq n} \subseteq \Pi_G^n$  and every path in  $\Lambda_G^{\leq n}$  visits a state in  $G$ . Correspondingly,  $\Lambda_{-G}^{\leq n} = \Pi_G^n \setminus \Lambda_G^{\leq n}$  is the set of time-abstract paths of length  $n$  that do not visit a state in  $G$ . Hence, the paths in  $\Pi_G^{n+1}$  with length  $n+1$  have a prefix in  $\Lambda_{-G}^{\leq n}$ . The set  $\Pi_G^{n+1}$  is partitioned such that

$$\begin{aligned} \Pi_G^{n+1} &= \Lambda_G^{\leq n} \cup \{\hat{\pi} \in \Pi_G^{n+1} \mid |\hat{\pi}| = n+1\} \\ &= \Lambda_G^{\leq n} \cup \{\hat{\pi} = \hat{\pi}' \xrightarrow{\alpha} s' \in FPaths^{\mathcal{M}\mathcal{D}} \mid \hat{\pi}' \in \Lambda_{-G}^{\leq n}\}. \end{aligned}$$

The reward obtained within the first  $n$  steps is independent of the  $(n+1)$ -th transition. To show this formally, we fix a path  $\hat{\pi}' \in \Lambda_{-G}^{\leq n}$  with  $\text{last}(\hat{\pi}') = s$  and derive

$$\begin{aligned} & \sum_{\hat{\pi}' \xrightarrow{\alpha} s' \in FPaths^{\mathcal{M}\mathcal{D}}} \int_{\pi \in \langle \hat{\pi}' \xrightarrow{\alpha} s' \rangle} \text{rew}^{\mathcal{M}}(\text{pref}(\pi, n)) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \\ &= \begin{cases} \int_{\pi' \in \langle \hat{\pi}' \rangle} \text{rew}^{\mathcal{M}}(\pi') \cdot \sum_{(\alpha, s') \in \text{Act} \times S} \sigma(\pi', \alpha) \cdot \mathbf{P}(s, \alpha, s') \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi') & \text{if } s \in \text{PS} \\ \int_{\pi' \in \langle \hat{\pi}' \rangle} \text{rew}^{\mathcal{M}}(\pi') \cdot \sum_{s' \in S} \mathbf{P}(s, \perp, s') \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi') & \text{if } s \in \text{MS} \end{cases} \\ &= \int_{\pi' \in \langle \hat{\pi}' \rangle} \text{rew}^{\mathcal{M}}(\pi') \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi'). \end{aligned} \tag{4.4}$$

With the above-mentioned partition of the set  $\Pi_G^{n+1}$ , it follows that the expected reward obtained within the first  $n$  steps is given by

$$\begin{aligned} & \sum_{\hat{\pi} \in \Pi_G^{n+1}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\text{pref}(\pi, n)) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \\ &= \sum_{\hat{\pi} \in \Lambda_G^{\leq n}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\pi) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \\ & \quad + \sum_{\hat{\pi}' \in \Lambda_{-G}^{\leq n}} \sum_{\hat{\pi}' \xrightarrow{\alpha} s' \in FPaths^{\mathcal{M}\mathcal{D}}} \int_{\pi \in \langle \hat{\pi}' \xrightarrow{\alpha} s' \rangle} \text{rew}^{\mathcal{M}}(\text{pref}(\pi, n)) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \\ &\stackrel{4.4}{=} \sum_{\hat{\pi} \in \Lambda_G^{\leq n}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\pi) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) + \sum_{\hat{\pi}' \in \Lambda_{-G}^{\leq n}} \int_{\pi \in \langle \hat{\pi}' \rangle} \text{rew}^{\mathcal{M}}(\pi) \, d\text{Pr}_{\sigma}^{\mathcal{M}}(\pi) \\ &= \text{eR}_{\sigma}^{\mathcal{M}}(\Pi_G^n) \end{aligned}$$

$$\begin{aligned}
& \stackrel{IH}{=} \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\Pi_G^n) \\
&= \sum_{\hat{\pi} \in \Lambda_G^{\leq n}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) + \sum_{\hat{\pi} \in \Lambda_{-G}^{\leq n}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \\
&= \sum_{\hat{\pi} \in \Lambda_G^{\leq n}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) + \sum_{\hat{\pi}' \in \Lambda_{-G}^{\leq n}} \sum_{\substack{\hat{\pi} \in \text{FPaths}^{\mathcal{M}_{\mathcal{D}}} \\ \hat{\pi} = \hat{\pi}' \xrightarrow{\alpha} s'}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\text{pref}(\hat{\pi}, n)) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \\
&= \sum_{\hat{\pi} \in \Pi_G^{n+1}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\text{pref}(\hat{\pi}, n)) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}). \tag{4.5}
\end{aligned}$$

**Term 4.3:** For the expected reward obtained in step  $n + 1$ , consider a path  $\hat{\pi} = \hat{\pi}' \xrightarrow{\alpha} s' \in \Pi_G^{n+1}$  such that  $|\hat{\pi}'| = n$  and  $\text{last}(\hat{\pi}') = s$ .

- If  $s \in \text{MS}$ , we have  $\hat{\pi} = \hat{\pi}' \xrightarrow{\perp} s'$ . It follows that

$$\begin{aligned}
& \int_{\pi = \pi' \xrightarrow{t} s' \in \langle \hat{\pi} \rangle} \rho(s) \cdot t + \rho(s, \perp) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) \\
&= \int_{\pi = \pi' \xrightarrow{t} s' \in \langle \hat{\pi} \rangle} \rho(s) \cdot t \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) + \int_{\pi \in \langle \hat{\pi} \rangle} \rho(s, \perp) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) \\
&= \rho(s) \cdot \int_{\pi' \in \langle \hat{\pi}' \rangle} \int_0^{\infty} t \cdot \mathbf{E}(s) \cdot e^{-\mathbf{E}(s)t} \cdot \mathbf{P}(s, \perp, s') \, \text{d}t \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi') + \rho(s, \perp) \cdot \text{Pr}_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \\
&= \frac{\rho(s)}{\mathbf{E}(s)} \cdot \text{Pr}_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) + \rho(s, \perp) \cdot \text{Pr}_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \\
&= \rho^{\mathcal{D}}(s, \perp) \cdot \text{Pr}_{\sigma}^{\mathcal{M}}(\langle \hat{\pi} \rangle) \stackrel{\text{Lem. 4.7}}{=} \rho^{\mathcal{D}}(s, \perp) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}). \tag{4.6}
\end{aligned}$$

- If  $s \in \text{PS}$ , then  $\int_{\pi = \pi' \xrightarrow{\alpha} s' \in \langle \hat{\pi} \rangle} \rho(s, \alpha) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) = \rho^{\mathcal{D}}(s, \alpha) \cdot \text{Pr}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi})$  follows similarly.

Combining the two results yields

$$\begin{aligned}
& \text{eR}_{\sigma}^{\mathcal{M}}(\Pi_G^{n+1}) \stackrel{4.2, 4.3}{=} \sum_{\hat{\pi} \in \Pi_G^{n+1}} \int_{\pi \in \langle \hat{\pi} \rangle} \text{rew}^{\mathcal{M}}(\text{pref}(\pi, n)) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi) \\
& \quad + \sum_{\substack{\hat{\pi} \in \Pi_G^{n+1} \\ |\hat{\pi}| = n+1}} \int_{\substack{\pi = \pi' \xrightarrow{\kappa} s' \in \langle \hat{\pi} \rangle \\ \text{last}(\pi') = s}} \rho(s) \cdot t(\kappa) + \rho(s, \alpha(\kappa)) \, \text{dPr}_{\sigma}^{\mathcal{M}}(\pi)
\end{aligned}$$

$$\begin{aligned}
& \stackrel{4.5, 4.6}{=} \sum_{\hat{\pi} \in \Pi_G^{n+1}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\text{pref}(\hat{\pi}, n)) \cdot \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \\
& + \sum_{\substack{\hat{\pi} = \hat{\pi}' \xrightarrow{\alpha} s' \in \Pi_G^{n+1} \\ |\hat{\pi}| = n+1}} \rho^{\mathcal{D}}(\text{last}(\hat{\pi}'), \alpha) \cdot \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \\
& = \sum_{\hat{\pi} \in \Pi_G^{n+1}} \text{rew}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) \cdot \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\hat{\pi}) = \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\Pi_G^{n+1}). \quad \square
\end{aligned}$$

We can lift the lemma to expected reachability rewards by letting the considered path length  $n$  approach infinity. The result is similar to Proposition 4.8 for unbounded until objectives.

**Proposition 4.13**

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $G \subseteq S$  and  $\sigma \in \text{GM}$ . Furthermore, let  $\rho$  be some reward function of  $\mathcal{M}$  and let  $\rho^{\mathcal{D}}$  be its counterpart for  $\mathcal{M}_{\mathcal{D}}$ . It holds that

$$\text{eR}_{\sigma}^{\mathcal{M}}(\rho, G) = \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, G). \quad \blacksquare$$

*Proof.* The proposition is a direct consequence of Lemma 4.11 and Lemma 4.12 as

$$\begin{aligned}
\text{eR}_{\sigma}^{\mathcal{M}}(\rho, G) &= \lim_{n \rightarrow \infty} \text{eR}_{\sigma}^{\mathcal{M}}(\rho, \Pi_G^n) \\
&= \lim_{n \rightarrow \infty} \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, \Pi_G^n) = \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, G). \quad \square
\end{aligned}$$

Proposition 4.13 is applied in the proof of Theorem 4.9.

*Proof of Theorem 4.9.* Let  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  be the considered list of unbounded until and expected reward objectives with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . The following equivalences hold for any  $\sigma \in \text{GM}^{\mathcal{M}}$  and  $\mathbf{p} \in \mathbb{R}^d$ .

$$\begin{aligned}
\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p} &\iff \forall i: \mathcal{M}, \sigma \models \mathbb{O}_i \triangleright_i p_i \\
&\stackrel{*}{\iff} \forall i: \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O}_i \triangleright_i p_i \iff \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O} \triangleright \mathbf{p},
\end{aligned}$$

where for the equivalence marked with  $*$  we consider two cases: If  $\mathbb{O}_i$  is of the form  $\mathbb{P}(HUG)$ , Proposition 4.8 yields

$$\begin{aligned}
\mathcal{M}, \sigma \models \mathbb{O}_i \triangleright_i p_i &\iff \Pr_{\sigma}^{\mathcal{M}}(HUG) \triangleright_i p_i \\
&\iff \Pr_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(HUG) \triangleright_i p_i \iff \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O}_i \triangleright_i p_i.
\end{aligned}$$

Otherwise,  $\mathbb{O}_i$  is of the form  $\mathbb{E}(\#j, G)$  and with Proposition 4.13 it follows that

$$\begin{aligned}
\mathcal{M}, \sigma \models \mathbb{O}_i \triangleright_i p_i &\iff \text{eR}_{\sigma}^{\mathcal{M}}(\rho_j, G) \triangleright_i p_i \\
&\iff \text{eR}_{\text{ta}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\rho_j^{\mathcal{D}}, G) \triangleright_i p_i \iff \mathcal{M}_{\mathcal{D}}, \text{ta}(\sigma) \models \mathbb{O}_i \triangleright_i p_i.
\end{aligned}$$

The remaining steps of the proof are completely analogous to the proof of Theorem 4.1 conducted on page 44.  $\square$

We conclude that multi-objective queries for an MA  $\mathcal{M}$  considering combinations of unbounded until and expected reachability reward objectives can be answered by analyzing the underlying MDP.

### 4.3 Bounded Until Objectives

The analysis of bounded until objectives can not be conducted on the underlying MDP of the MA  $\mathcal{M}$ . The main reason is that the transition probability function  $\mathbf{P}$  of  $\mathcal{M}_{\mathcal{D}}$  carries no information of the time spent at the states of the model. The following example illustrates why this is an issue when computing bounded until probabilities.

**Example 4.14**

Consider the MA  $\mathcal{M}$  as shown in Figure 4.3(a) as well as the bounded until objective  $\mathbb{P}(\{s_0, s_1\} \mathcal{U}^{\leq 2} \{s_3\})$ . Note that  $s_1$  is the only state of  $\mathcal{M}$  where the choice of a scheduler is not unique. We define  $\sigma \in \text{GM}^{\mathcal{M}}$  that satisfies

$$\sigma(s_0 \xrightarrow{t} s_1, \alpha) = \begin{cases} 1 & \text{if } t \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

We have

$$\{s_0, s_1\} \mathcal{U}^{\leq 2} \{s_3\} = \{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \mid 0 \leq t \leq 2\}.$$

Thus, the probability  $\text{Pr}_{\sigma}^{\mathcal{M}}(\{s_0, s_1\} \mathcal{U}^{\leq 2} \{s_3\})$  comprises

- the probability to leave  $s_0$  between time point 1 and 2 (to assert that  $\sigma$  chooses  $\alpha$  and that  $s_3$  is reached in time) and
- the probability to move from  $s_1$  to  $s_3$  when action  $\alpha$  is performed.

Hence, we obtain

$$\text{Pr}_{\sigma}^{\mathcal{M}}(\{s_0, s_1\} \mathcal{U}^{\leq 2} \{s_3\}) = e^{-\lambda} \cdot (1 - e^{-\lambda}) \cdot 0.5 = \frac{e^{-\lambda} - e^{-2\lambda}}{2}.$$

Note that this value depends on the rate  $\lambda$  which does not occur in the underlying MDP  $\mathcal{M}_{\mathcal{D}}$  of  $\mathcal{M}$  depicted in Fig 4.3(b).  $\blacksquare$

If only a single bounded until objective is considered, [HH12, GHH<sup>+</sup>13, GHH<sup>+</sup>14] show how a *digitization technique* can be employed to obtain a sound and arbitrary close approximation of the maximal (or minimal) value for this objective. Our goal is to lift this idea to multiple objectives.

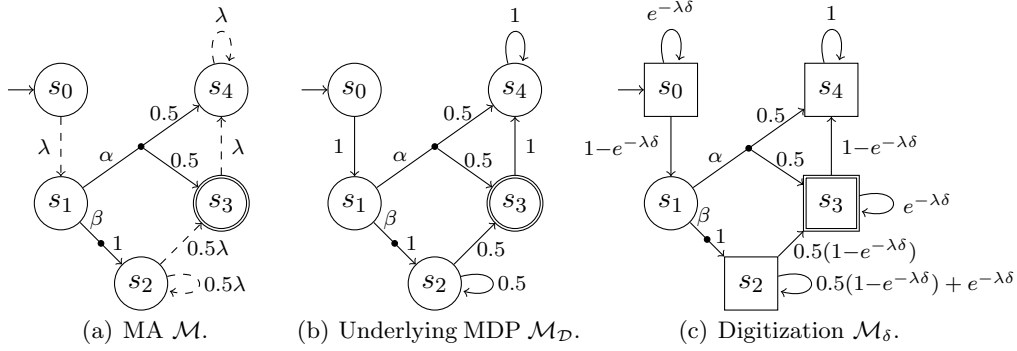


Figure 4.3: MA  $\mathcal{M}$  with underlying MDP  $\mathcal{M}_{\mathcal{D}}$  and digitization  $\mathcal{M}_{\delta}$  (cf. Example 4.14 and Example 4.15).

First, we present how a bounded until probability  $\Pr_{\sigma}^{\mathcal{M}}(HU^I G)$  can be approximated for an arbitrary scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$  using the above-mentioned digitization technique. Our results hold for any scheduler of the MA  $\mathcal{M}$  while existing works (such as [HH12, GHH<sup>+</sup>14]) only consider schedulers that induce a maximal or minimal probability. Our more general results require entirely new proofs as the existing ones are tailored to the optimization of a single objective. An MDP  $\mathcal{M}_{\delta}$  (called the digitization of  $\mathcal{M}$ ) and a scheduler  $\text{di}(\sigma)$  of  $\mathcal{M}_{\delta}$  (called the digitization of  $\sigma$ ) are defined with respect to some *digitization constant*  $\delta \in \mathbb{R}_{>0}$ . A *lower* and an *upper bound* for  $\Pr_{\sigma}^{\mathcal{M}}(HU^I G)$  can be established by an analysis of  $\mathcal{M}_{\delta}$  under  $\text{di}(\sigma)$ . The bounds can be tightened arbitrarily by considering smaller digitization constants  $\tilde{\delta} < \delta$ .

In a second step, we lift the presented notions to the case where multiple bounded until objectives are considered. Here, we make use of the fact that arbitrary schedulers can be considered (not just schedulers that are optimal for a single objective). We show that a multi-objective analysis of  $\mathcal{M}_{\delta}$  yields an approximation of the set of achievable points of the original MA  $\mathcal{M}$ . For  $d = 2$  objectives, Figure 4.4 illustrates this connection in more detail. Let the gray area in Figure 4.4(a) denote the set  $A$  of achievable points in  $\mathcal{M}_{\delta}$ . Our aim is to infer an *under-approximation*  $A^{-}$  and an *over-approximation*  $A^{+}$  from the set  $A$  such that

1. Every point  $\mathbf{p} \in A^{-}$  is achievable in  $\mathcal{M}$  (the green area in Figure 4.4(b)) and
2. Every point  $\mathbf{q} \in \mathbb{R}^d \setminus A^{+}$  is not achievable in  $\mathcal{M}$  (the red area in Figure 4.4(b)).

These approximation can be refined arbitrarily by picking a smaller digitization constant  $\tilde{\delta} < \delta$  as illustrated in Figure 4.4(c).

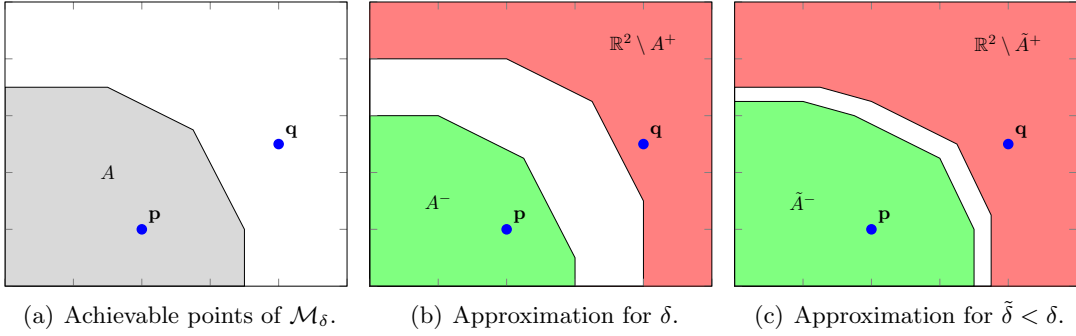


Figure 4.4: Illustration of the approximation of achievable points in  $\mathcal{M}$ .

### 4.3.1 Digitization Approach

We formally introduce the *digitization*  $\mathcal{M}_\delta$  of  $\mathcal{M}$  with respect to some digitization constant  $\delta \in \mathbb{R}_{>0}$  and detail the connection between paths of  $\mathcal{M}_\delta$  and  $\mathcal{M}$ . The definition of  $\mathcal{M}_\delta$  is similar to the definition of the underlying MDP (cf. Definition 2.15 on page 17). The main difference between  $\mathcal{M}_\mathcal{D}$  and  $\mathcal{M}_\delta$  is that the latter also introduces *self-loops*<sup>1</sup> which describe the probability to stay in a Markovian state for  $\delta$  time units. More precisely, the outgoing transitions of states  $s \in \text{MS}$  in  $\mathcal{M}_\delta$  represent that either

- some transition  $s \xrightarrow{\lambda} s'$  is taken within  $\delta$  time units (happens with probability  $\mathbf{P}(s, \perp, s') \cdot (1 - e^{-\mathbf{E}(s)\delta})$ ), or
- no transition is taken within  $\delta$  time units (happens with probability  $e^{-\mathbf{E}(s)\delta}$ ).

#### Example 4.15

The digitization  $\mathcal{M}_\delta$  of the MA  $\mathcal{M}$  from Figure 4.3(a) is shown in Figure 4.3(c). States that correspond to Markovian states of the original MA are depicted with rectangles. ■

The idea of the digitization approach is to count the number of taken transitions in  $\mathcal{M}_\delta$  that emerge from Markovian states. (we refer to this number as *digitization steps*). Since one digitization step represents the elapsing of at most  $\delta$  time units, we obtain an estimation of the actual sojourn times by multiplying the digitization steps with  $\delta$ . This allows us to approximate (time-)bounded until probabilities for  $\mathcal{M}$  by considering digitization step bounded until probabilities for  $\mathcal{M}_\delta$  instead.

#### Definition 4.16 (Digitization of an MA)

For an MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  with transition probability function  $\mathbf{P}$  and a digitization constant  $\delta \in \mathbb{R}_{>0}$ , the digitization of  $\mathcal{M}$  w.r.t.  $\delta$  is given by the

<sup>1</sup>A self-loop refers to a transition at some state  $s \in S$  that leads back to  $s$ .

MDP  $\mathcal{M}_\delta = (S, Act, \mathbf{P}_\delta, s_0, \{\rho_1^\delta, \dots, \rho_\ell^\delta\})$ , where  $\mathbf{P}_\delta$  satisfies

$$\mathbf{P}_\delta(s, \alpha, s') = \begin{cases} \mathbf{P}(s, \alpha, s') & \text{if } s \in \text{PS} \\ \mathbf{P}(s, \perp, s') \cdot (1 - e^{-E(s)\delta}) & \text{if } s \in \text{MS}, \alpha = \perp, s \neq s' \\ \mathbf{P}(s, \perp, s') \cdot (1 - e^{-E(s)\delta}) + e^{-E(s)\delta} & \text{if } s \in \text{MS}, \alpha = \perp, s = s' \\ 0 & \text{otherwise} \end{cases}$$

and for each  $i \in \{1, \dots, \ell\}$

$$\rho_i^\delta(s, \alpha) = \begin{cases} \rho_i(s, \alpha) & \text{if } s \in \text{PS} \\ (\rho_i(s, \perp) + 1/E(s) \cdot \rho_i(s)) \cdot (1 - e^{-E(s)\delta}) & \text{if } s \in \text{MS and } \alpha = \perp \\ 0 & \text{otherwise.} \end{cases} \quad \blacksquare$$

Let  $\delta \in \mathbb{R}_{>0}$  be a digitization constant and let  $\mathcal{M}_\delta = (S, Act, \mathbf{P}_\delta, s_0, \{\rho_1^\delta, \dots, \rho_\ell^\delta\})$  be the digitization of  $\mathcal{M}$  with respect to  $\delta$ . The reward functions  $\rho_1^\delta, \dots, \rho_\ell^\delta$  are relevant when trade-offs between bounded until and expected reachability reward objectives are analyzed. This scenario is discussed in Section 4.4.

The outgoing transitions of states in PS are the same for  $\mathcal{M}$  and  $\mathcal{M}_\delta$ . As discussed above, each state  $s \in \text{MS}$  receives a self-loop which represents the possibility to stay at  $s$  for  $\delta$  time units. Applying this notion, we can transform every path  $\pi$  of  $\mathcal{M}$  to a path  $\bar{\pi}$  of  $\mathcal{M}_\delta$ . This transformation is called the *digitization* of  $\pi$ .

**Definition 4.17 (Digitization of a Path)**

Let  $\pi = s_0 \xrightarrow{\kappa_0} s_1 \xrightarrow{\kappa_1} \dots$  be a finite or infinite path of an MA  $\mathcal{M}$  and  $\delta \in \mathbb{R}_{>0}$  be a digitization constant. Further, let  $m_i = \max\{m \in \mathbb{N} \mid m\delta \leq t(\kappa_i)\}$  for each  $i \geq 0$ . The digitization  $\text{di}(\pi)$  of  $\pi$  is a path of  $\mathcal{M}_\delta$  given by

$$\text{di}(\pi) = (s_0 \xrightarrow{\alpha(\kappa_0)} s_0)^{m_0} s_0 \xrightarrow{\alpha(\kappa_0)} (s_1 \xrightarrow{\alpha(\kappa_1)} s_1)^{m_1} s_1 \xrightarrow{\alpha(\kappa_1)} \dots \quad \blacksquare$$

The  $m_i$  in the definition above can be interpreted as a digitization of the time-stamps  $t(\kappa_i)$  such that  $m_i\delta \leq t(\kappa_i) < (m_i + 1)\delta$ . These digitized time-stamps are incorporated into the digitization of a path by taking the self-loop at state  $s_i \in \text{MS}$   $m_i$  times. We also refer to the paths of  $\mathcal{M}_\delta$  as *digital paths* of  $\mathcal{M}$ .

**Example 4.18**

Consider the MA  $\mathcal{M}$  and its digitization  $\mathcal{M}_\delta$  from Figure 4.3. We assume the digitization constant  $\delta = 0.1$ . Consider the following paths of  $\mathcal{M}$  and their digitization.

- For  $\pi_1 = s_0 \xrightarrow{0.25} s_1 \xrightarrow{\alpha} s_3$  we obtain  $m_0 = \max\{m \in \mathbb{N} \mid m \cdot 0.1 \leq 0.25\} = 2$  and  $m_1 = \max\{m \in \mathbb{N} \mid m \cdot 0.1 \leq 0\} = 0$ . Hence, the digitization of  $\pi_1$  is given by

$$\text{di}(\pi_1) = (s_0 \xrightarrow{\perp})^2 s_0 \xrightarrow{\perp} (s_1 \xrightarrow{\alpha})^0 s_1 \xrightarrow{\alpha} s_3 = s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3 .$$

- The digitization of  $\pi_2 = s_0 \xrightarrow{2.5} s_1 \xrightarrow{\alpha} s_4 \xrightarrow{1} s_4 \xrightarrow{1} \dots$  is given by

$$\text{di}(\pi_2) = (s_0 \xrightarrow{\perp} s_0)^{25} s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_4 \xrightarrow{\perp} s_4 \xrightarrow{\perp} \dots$$

- The digitization of  $\pi_3 = s_0 \xrightarrow{0.01} s_1$  is given by  $\text{di}(\pi_3) = s_0 \xrightarrow{\perp} s_1$ . Note that the digitization of the time-stamp 0.01 is not very accurate in this case since, e.g.,  $\pi_4 = s_0 \xrightarrow{0.09} s_1$  yields the same digital path. ■

A scheduler  $\sigma \in \text{TA}^{\mathcal{M}_\delta}$  for  $\mathcal{M}_\delta$  can be transformed to a scheduler  $\sigma' \in \text{GM}^{\mathcal{M}}$  for  $\mathcal{M}$ : Let  $\pi \in \text{FPaths}^{\mathcal{M}}$  and  $\alpha \in \text{Act}$ . We set  $\sigma'(\pi, \alpha) = \sigma(\text{di}(\pi), \alpha)$ . For simplicity, this transformation is made implicit, i.e., we assume that  $\sigma$  is also a scheduler for  $\mathcal{M}$ .

Let  $\bar{\pi} = s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{n-1}} s_n \in \text{FPaths}^{\mathcal{M}_\delta}$  be a finite digital path. We denote by  $|\bar{\pi}|_{\text{ds}}$  the number of *digitization steps* of  $\bar{\pi}$  which is the number of depicted transitions emerging from Markovian states, i.e.,  $|\bar{\pi}|_{\text{ds}} = |\{i \in \{0, \dots, n-1\} \mid s_i \in \text{MS}\}|$ . The notation is extended to paths of  $\mathcal{M}$ , where  $|\pi|_{\text{ds}} = |\text{di}(\pi)|_{\text{ds}}$  for any  $\pi \in \text{FPaths}^{\mathcal{M}}$ . Intuitively, one digitization step represents the elapsing of at most  $\delta$  time units. Consequently, the duration of a path with  $k$  digitization steps is at most  $k\delta$ . This is formalized by the following lemma.

**Lemma 4.19**

For a path  $\pi \in \text{FPaths}^{\mathcal{M}}$  and digitization constant  $\delta$  it holds that

$$T(\pi) \leq |\pi|_{\text{ds}} \cdot \delta . \quad \blacksquare$$

*Proof.* Let  $\pi = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n$  and let  $m_i = \max\{m \in \mathbb{N} \mid m\delta \leq t(\kappa_i)\}$  for each  $i \in \{0, \dots, n-1\}$  (as in Definition 4.17). The number  $|\pi|_{\text{ds}}$  is given by  $\sum_{0 \leq i < n, s_i \in \text{MS}} (m_i + 1)$ . With  $t(\kappa_i) \leq (m_i + 1)\delta$  it follows that

$$T(\pi) = \sum_{\substack{0 \leq i < n \\ s_i \in \text{MS}}} t(\kappa_i) \leq \sum_{\substack{0 \leq i < n \\ s_i \in \text{MS}}} (m_i + 1)\delta = |\pi|_{\text{ds}} \cdot \delta . \quad \square$$

**Example 4.20**

Consider again the path  $\pi_1 = s_0 \xrightarrow{0.25} s_1 \xrightarrow{\alpha} s_3$  from Example 4.18 with digitization  $\text{di}(\pi_1) = s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3$ . The number of digitization steps of  $\pi_1$  is given by  $|\pi_1|_{\text{ds}} = |\text{di}(\pi_1)|_{\text{ds}} = 3$ . With  $\delta = 0.1$  it follows that  $T(\pi_1) = 0.25 \leq 0.3 = |\pi_1|_{\text{ds}} \cdot \delta$ . ■

Multiple paths of  $\mathcal{M}$  can have the same digitization. Correspondingly, some digital path  $\bar{\pi} \in \text{FPaths}^{\mathcal{M}_\delta}$  can be interpreted as a representation of the set of paths of  $\mathcal{M}$  whose digitization is  $\bar{\pi}$ .

**Definition 4.21 (Induced Paths of a Digital Path)**

For a digital path  $\bar{\pi} \in FPaths^{\mathcal{M}_\delta}$  of MA  $\mathcal{M}$ , the set of *induced paths* of  $\bar{\pi}$  is given by

$$[\bar{\pi}] = \text{di}^{-1}(\bar{\pi}) = \{\pi \in FPaths^{\mathcal{M}} \mid \text{di}(\pi) = \bar{\pi}\}. \quad \blacksquare$$

**Example 4.22**

Let  $\mathcal{M}$  be the MA with digitization  $\mathcal{M}_\delta$  as shown in Figure 4.3. We assume  $\delta = 0.1$ .

Consider the digital path  $\bar{\pi}_1 = s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3$ . The set  $[\bar{\pi}_1]$  contains, e.g., the path  $\pi_1 = s_0 \xrightarrow{0.25} s_1 \xrightarrow{\alpha} s_3$  from Example 4.18 since  $\text{di}(\pi_1) = \bar{\pi}_1$ . More generally, the set of paths induced by  $\bar{\pi}_1$  is given by

$$[\bar{\pi}_1] = \{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \in FPaths^{\mathcal{M}} \mid 0.2 \leq t < 0.3\}.$$

Next, consider the digital path  $\bar{\pi}_2 = s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_0$ . Note that there is no path  $\pi \in FPaths^{\mathcal{M}}$  with  $\text{di}(\pi) = \bar{\pi}_2$ , implying  $[\bar{\pi}_2] = \emptyset$ . The intuitive reason is that  $\bar{\pi}_2$  depicts a sojourn time at  $\text{last}(\bar{\pi}_2)$  but finite paths of MAs do not depict sojourn times at their last state.  $\blacksquare$

As seen in Example 4.22,  $[\bar{\pi}] = \emptyset$  may hold although  $\bar{\pi}$  intuitively represents feasible system runs. The issue arises whenever the considered digital path ends with a self-loop at a Markovian state (which represents a sojourn time at  $\text{last}(\bar{\pi})$ ). To avoid this problematic, we also consider the infinite paths of  $\mathcal{M}$  that are represented by a finite digital path.

**Definition 4.23 (Induced Cylinder of a Digital Path)**

Given a digital path  $\bar{\pi} \in FPaths^{\mathcal{M}_\delta}$  of MA  $\mathcal{M}$ , the *induced cylinder* of  $\bar{\pi}$  is given by

$$[\bar{\pi}]_{cyl} = \{\pi \in IPaths^{\mathcal{M}} \mid \bar{\pi} \text{ is a prefix of } \text{di}(\pi)\}. \quad \blacksquare$$

$[\bar{\pi}]_{cyl} = Cyl([\bar{\pi}])$  holds whenever  $\bar{\pi}$  does not end with a self-loop at a Markovian state. Here,  $Cyl([\bar{\pi}])$  refers to the cylinder of the set  $[\bar{\pi}]$  (cf. Definition 2.21 on page 21).

**Example 4.24**

Let  $\mathcal{M}$ ,  $\mathcal{M}_\delta$ ,  $\delta$ ,  $\bar{\pi}_1$ , and  $\bar{\pi}_2$  be as in Example 4.22. The set  $[\bar{\pi}_1]_{cyl}$  contains each infinite path whose digitization has the prefix  $\bar{\pi}_1$ , i.e.,

$$[\bar{\pi}_1]_{cyl} = \{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \xrightarrow{\kappa} \dots \in IPaths^{\mathcal{M}} \mid 0.2 \leq t < 0.3\}.$$

We observe that these are exactly the paths that have a prefix in  $[\bar{\pi}_1]$ . Put differently, we have  $[\bar{\pi}_1]_{cyl} = Cyl([\bar{\pi}_1])$ .

The induced cylinder of  $\bar{\pi}_2$  contains all paths that sojourn at least  $2\delta$  time units at  $s_0$ , i.e.,

$$[\bar{\pi}_2]_{cyl} = \{s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa} \dots \in IPaths^{\mathcal{M}} \mid t \geq 0.2\}. \quad \blacksquare$$

### 4.3.2 A Lower Bound for $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$

Assume sets of states  $H, G \subseteq S$ , a time interval  $I$ , and a scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ . We show how a lower bound for the probability  $\Pr_\sigma^{\mathcal{M}}(HU^I G)$  can be established by an analysis of the digitization of  $\mathcal{M}$  with respect to some digitization constant. At first, we restrict ourselves to the case where  $I = [0, b]$  for an upper time bound  $b > 0$ . An extension of our results to non-zero lower time bounds is discussed in Section 4.3.4.

Let  $\delta \in \mathbb{R}_{>0}$  be a digitization constant such that  $b/\delta = k$  for some  $k \in \mathbb{N}$  and let  $\mathcal{M}_\delta$  be the digitization of  $\mathcal{M}$  with respect to  $\delta$ . The idea is to give a lower bound for  $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$  by computing the probability of the paths that reach  $G$  via  $H$  with at most  $k$  digitization steps.

#### Definition 4.25 (Digitization Step Bounded Paths)

Let  $\mathcal{M}$  be an MA with digitization  $\mathcal{M}_\delta$  and subsets of states  $H$  and  $G$ . Further, let  $J \subseteq \mathbb{N}$  be a (finite or infinite) set of consecutive natural numbers. The set of infinite digital paths that reach  $G$  via  $H$  within the digitization step bounds  $J$  is given by

$$HU_{\text{ds}}^J G = \{\bar{\pi} = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots \in \text{IPaths}^{\mathcal{M}_\delta} \mid \exists n \geq 0: s_n \in G, \forall i < n: s_i \in H, \text{ and } [\text{pref}(\bar{\pi}, n)]_{\text{ds}} \in J\}. \quad \blacksquare$$

We simplify the notation for the digitization step bounds  $J$  by writing, e.g.,  $HU_{\text{ds}}^{\leq k} G$  for  $J = \{0, \dots, k\}$  or  $HU_{\text{ds}}^{(j,k]} G$  for  $J = \{j+1, \dots, k\}$ . The set of infinite paths of  $\mathcal{M}$  whose digitization is in  $HU_{\text{ds}}^J G$  is denoted by

$$[HU_{\text{ds}}^J G] = \{\pi \in \text{IPaths}^{\mathcal{M}} \mid \text{di}(\pi) \in HU_{\text{ds}}^J G\}.$$

#### Example 4.26

Consider the MA  $\mathcal{M}$  and its digitization  $\mathcal{M}_\delta$  from Figure 4.3. The set of digital paths that reach  $s_3$  via  $s_0$  and  $s_1$  within at most 20 digitization steps is given by

$$\{s_0, s_1\} \mathcal{U}_{\text{ds}}^{\leq 20} \{s_3\} = \{(s_0 \xrightarrow{\perp} s_0)^m s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_3 \xrightarrow{\perp} \dots \in \text{IPaths}^{\mathcal{M}_\delta} \mid 0 \leq m < 20\}.$$

We assume  $\delta = 0.1$ . The corresponding paths of  $\mathcal{M}$  are given by the set

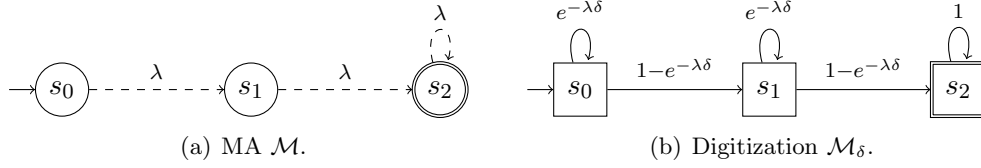
$$[\{s_0, s_1\} \mathcal{U}_{\text{ds}}^{\leq 20} \{s_3\}] = \{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} s_3 \xrightarrow{\kappa} \dots \in \text{IPaths}^{\mathcal{M}} \mid t < 2\}. \quad \blacksquare$$

The following lemma expresses that the *digitization step bounded until probability*  $\Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G])$  is a lower bound for  $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$ .

#### Lemma 4.27

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$  and  $\sigma \in \text{GM}$ . For  $b, \delta \in \mathbb{R}_{>0}$  such that  $b/\delta = k \in \mathbb{N}$  it holds that

$$\Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G]) \leq \Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G). \quad \blacksquare$$

Figure 4.5: MA  $\mathcal{M}$  with digitization  $\mathcal{M}_\delta$  (cf. Example 4.28).

*Proof.* We show that  $[H\mathcal{U}_{\text{ds}}^{\leq k} G] \subseteq H\mathcal{U}^{\leq b} G$ . Let  $\pi \in [H\mathcal{U}_{\text{ds}}^{\leq k} G]$  and let  $\pi'$  be the smallest prefix of  $\pi$  with  $\text{last}(\pi') \in G$ . It follows that  $\text{di}(\pi')$  is also the smallest prefix of  $\text{di}(\pi)$  with  $\text{last}(\text{di}(\pi')) \in G$ . Since  $\text{di}(\pi) \in H\mathcal{U}_{\text{ds}}^{\leq k} G$ , it follows that  $|\pi'|_{\text{ds}} = |\text{di}(\pi')|_{\text{ds}} \leq k$ . From Lemma 4.19 we obtain

$$T(\pi') \leq |\pi'|_{\text{ds}} \cdot \delta = |\text{di}(\pi')|_{\text{ds}} \cdot \delta \leq k\delta = b.$$

Hence, the prefix  $\pi'$  reaches  $G$  via  $H$  within  $b$  time units, implying  $\pi \in H\mathcal{U}^{\leq b} G$ .  $\square$

#### Example 4.28

Consider the MA  $\mathcal{M}$  with digitization  $\mathcal{M}_\delta$  as illustrated in Figure 4.5. We consider the probability  $\Pr^{\mathcal{M}}(S\mathcal{U}^{\leq 0.8} \{s_2\})$ , where the (unique) scheduler of  $\mathcal{M}$  is omitted from the notation. We choose the digitization constant  $\delta = 0.4$  which yields the upper digitization step bound  $0.8/0.4 = 2 \in \mathbb{N}$ . Notice that

$$[S\mathcal{U}_{\text{ds}}^{\leq 2} \{s_2\}] = \{s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \dots \in \text{IPaths}^{\mathcal{M}} \mid t_0 < 0.4 \text{ and } t_1 < 0.4\}$$

is a subset of

$$\{s_0, s_1\} \mathcal{U}^{\leq 0.8} \{s_2\} = \{s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \dots \in \text{IPaths}^{\mathcal{M}} \mid t_0 + t_1 \leq 0.8\}$$

which yields  $\Pr^{\mathcal{M}}([S\mathcal{U}_{\text{ds}}^{\leq 2} \{s_2\}]) \leq \Pr^{\mathcal{M}}(S\mathcal{U}^{\leq 0.8} \{s_2\})$ , as claimed in Lemma 4.27. We emphasize that the equality  $\Pr^{\mathcal{M}}([S\mathcal{U}_{\text{ds}}^{\leq 2} \{s_2\}]) = \Pr^{\mathcal{M}}(S\mathcal{U}^{\leq 0.8} \{s_2\})$  does *not* hold: There is a positive probability to reach  $s_2$  with more than 2 digitization steps but within 0.8 time units, e.g., by sojourning at  $s_0$  for  $t_0 \in [0.4, 0.8]$  time units and leaving  $s_1$  within  $t_1 \leq 0.8 - t_0$  time.  $\blacksquare$

The next step is to express the probability  $\Pr_\sigma^{\mathcal{M}}([H\mathcal{U}_{\text{ds}}^{\leq k} G])$  as a probability over paths of  $\mathcal{M}_\delta$ . To this end, we construct a scheduler  $\text{di}(\sigma)$  for  $\mathcal{M}_\delta$  that mimics the choices of  $\sigma$ . Our aim is to define  $\text{di}(\sigma)$  such that  $\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(H\mathcal{U}_{\text{ds}}^{\leq k} G) = \Pr_\sigma^{\mathcal{M}}([H\mathcal{U}_{\text{ds}}^{\leq k} G])$  holds.

#### Definition 4.29 (Digitization of a Scheduler)

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with digitization  $\mathcal{M}_\delta$ . The *digitization of scheduler*  $\sigma \in \text{GM}^{\mathcal{M}}$  is defined as  $\text{di}(\sigma) \in \text{TA}^{\mathcal{M}_\delta}$  such that

$$\text{di}(\sigma)(\bar{\pi}, \alpha) = \int_{\pi \in [\bar{\pi}]} \sigma(\pi, \alpha) \text{dPr}_\sigma^{\mathcal{M}}(\pi \mid [\bar{\pi}])$$

for each  $\bar{\pi} \in \text{FPaths}^{\mathcal{M}_\delta}$  with  $\text{last}(\bar{\pi}) \in \text{PS}$  and  $\alpha \in \text{Act}$ .  $\blacksquare$

Intuitively,  $\text{di}(\sigma)(\bar{\pi}, \alpha)$  considers  $\sigma(\pi, \alpha)$  for each  $\pi \in [\bar{\pi}]$ , weighted with the probability that the time-stamps of a path in  $[\bar{\pi}]$  are as given by  $\pi$ . The restriction  $\text{last}(\bar{\pi}) \in \text{PS}$  asserts that  $\bar{\pi}$  does not end with a self-loop on a Markovian state, implying  $[\bar{\pi}] \neq \emptyset$ . Note that  $\text{di}(\sigma) = \sigma$  if  $\sigma \in \text{TA}^{\mathcal{M}^\delta}$ . We emphasize the similarity of  $\text{di}(\sigma)$  to the time-abstraction  $\text{ta}(\sigma)$  from Definition 4.5 on page 41. Both schedulers get a path with restricted timing information as an input and mimic the choice of  $\sigma$ . However, while  $\text{ta}(\sigma)$  receives no information regarding sojourn times,  $\text{di}(\sigma)$  receives the digital estimate.

**Example 4.30**

Consider the MA  $\mathcal{M}$  and the digitization  $\mathcal{M}_\delta$  from Figure 4.3 on page 53. Further, let  $\sigma \in \text{GM}^{\mathcal{M}}$  be the scheduler of  $\mathcal{M}$  from Example 4.14, satisfying

$$\sigma(s_0 \xrightarrow{t} s_1, \alpha) = \begin{cases} 1 & \text{if } t \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

- Assume  $\delta = 0.1$ . Notice that the sojourn time of a path  $\pi$  of  $\mathcal{M}$  in  $s_0$  is at least 1 iff  $\text{di}(\pi)$  takes the self-loop at  $s_0$  at least  $1/\delta = 10$  times. We obtain the digitization  $\text{di}(\sigma) \in \text{TA}^{\mathcal{M}^\delta}$  which satisfies

$$\text{di}(\sigma)((s_0 \xrightarrow{\perp})^m s_0 \xrightarrow{\perp} s_1, \alpha) = \begin{cases} 1 & \text{if } m \geq 10 \\ 0 & \text{otherwise.} \end{cases}$$

The probability  $\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\{s_0, s_1\} \mathcal{U}_{\text{ds}}^{\leq 20} \{s_3\})$  comprises

- the probability to take the self-loop at  $s_0$  at least 10 and at most 19 times (to assert that  $\text{di}(\sigma)$  chooses  $\alpha$  and that  $s_3$  is reached within 20 digitization steps) and
- the probability to move from  $s_1$  to  $s_3$  when action  $\alpha$  is performed.

Hence, we obtain

$$\begin{aligned} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\{s_0, s_1\} \mathcal{U}_{\text{ds}}^{\leq 20} \{s_3\}) &= \sum_{m=10}^{19} (e^{-\lambda \delta m} \cdot (1 - e^{-\lambda \delta})) \cdot 0.5 \\ &= (e^{-\lambda \delta \cdot 10} - e^{-\lambda \delta \cdot 20}) \cdot 0.5 = \frac{e^{-\lambda} - e^{-2\lambda}}{2}. \end{aligned}$$

Notice that this value coincides with the probability  $\Pr_\sigma^{\mathcal{M}}(\{s_0, s_1\} \mathcal{U}^{\leq 20 \cdot \delta} \{s_3\})$  which has been computed in Example 4.14.

- Assume  $\delta = 0.4$  and consider the digital path  $\bar{\pi} = (s_0 \xrightarrow{\perp} s_0)^m s_0 \xrightarrow{\perp} s_1$ .
  - If  $m < 2$ , then each path  $\pi \in [\bar{\pi}]$  sojourns in  $s_0$  for less than  $2\delta = 0.8$  time units. On such paths,  $\sigma$  chooses  $\alpha$  with probability 0. It follows that  $\text{di}(\sigma)(\bar{\pi}, \alpha) = 0$ .
  - If  $m \geq 3$ , then any  $\pi \in [\bar{\pi}]$  sojourns in  $s_0$  for at least  $3\delta = 1.2$  time units and we have  $\text{di}(\sigma)(\bar{\pi}, \alpha) = 1$ .
  - For the case  $m = 2$ ,  $\text{di}(\sigma)$  needs to choose probabilistically between  $\alpha$  and  $\beta$ . We have

$$\begin{aligned}
 \text{di}(\sigma)(\bar{\pi}, \alpha) &= \int_{s_0 \xrightarrow{t} s_1 \in [\bar{\pi}]} \sigma(\pi, \alpha) \, \text{dPr}_\sigma^{\mathcal{M}}(\pi \mid [\bar{\pi}]) \\
 &= \frac{\int_{1.0}^{1.2} \lambda e^{-\lambda t} \, dt}{\text{Pr}_\sigma^{\mathcal{M}}([\bar{\pi}])} \\
 &= \frac{e^{-1.0\lambda} - e^{-1.2\lambda}}{e^{-0.8\lambda} - e^{-1.2\lambda}}. \quad \blacksquare
 \end{aligned}$$

The schedulers  $\sigma$  and  $\text{di}(\sigma)$  induce the same probabilities for a given digital path. This is formalized by the following lemma. Note that a similar statement for  $\text{ta}(\sigma)$  and time-abstract paths was shown in Lemma 4.7.

**Lemma 4.31**

Let  $\mathcal{M}$  be an MA with scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ , digitization  $\mathcal{M}_\delta$ , and digital path  $\bar{\pi} \in \text{FPaths}^{\mathcal{M}_\delta}$ . It holds that

$$\text{Pr}_\sigma^{\mathcal{M}}([\bar{\pi}]_{\text{cyl}}) = \text{Pr}_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\bar{\pi}). \quad \blacksquare$$

*Proof.* The proof is by induction over the length of the considered path  $|\bar{\pi}| = n$ . Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_\ell\})$  and  $\mathcal{M}_\delta = (S, \text{Act}, \mathbf{P}_\delta, s_0, \{\rho_1^\delta, \dots, \rho_\ell^\delta\})$ . If  $n = 0$ , then  $\bar{\pi} = s_0$  and  $[\bar{\pi}]_{\text{cyl}} = \text{IPaths}^{\mathcal{M}}$ . Hence,  $\text{Pr}_\sigma^{\mathcal{M}}([s_0]_{\text{cyl}}) = 1 = \text{Pr}_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(s_0)$ . In the induction step it is assumed that the lemma holds for a fixed path  $\bar{\pi} \in \text{FPaths}^{\mathcal{M}_\delta}$  with  $|\bar{\pi}| = n$  and  $\text{last}(\bar{\pi}) = s$ . Consider a path  $\bar{\pi} \xrightarrow{\alpha} s' \in \text{FPaths}^{\mathcal{M}_\delta}$ . We distinguish the following cases.

**Case  $s \in \text{PS}$ :** It follows that  $[\bar{\pi} \xrightarrow{\alpha} s']_{cyl} = \text{Cyl}([\bar{\pi} \xrightarrow{\alpha} s'])$  since  $\bar{\pi} \xrightarrow{\alpha} s'$  ends with a probabilistic transition. Hence,

$$\begin{aligned}
\Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\alpha} s']_{cyl}) &= \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\alpha} s']) \\
&= \int_{\pi \in [\bar{\pi}]} \sigma(\pi, \alpha) \cdot \mathbf{P}(s, \alpha, s') \, d\Pr_{\sigma}^{\mathcal{M}}(\pi) \\
&= \int_{\pi \in [\bar{\pi}]} \sigma(\pi, \alpha) \cdot \mathbf{P}(s, \alpha, s') \, d\Pr_{\sigma}^{\mathcal{M}}(\{\pi\} \cap [\bar{\pi}]) \\
&= \int_{\pi \in [\bar{\pi}]} \sigma(\pi, \alpha) \cdot \mathbf{P}(s, \alpha, s') \, d[\Pr_{\sigma}^{\mathcal{M}}(\pi \mid [\bar{\pi}]) \cdot \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}])] \\
&= \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]) \cdot \mathbf{P}(s, \alpha, s') \cdot \int_{\pi \in [\bar{\pi}]} \sigma(\pi, \alpha) \, d\Pr_{\sigma}^{\mathcal{M}}(\pi \mid [\bar{\pi}]) \\
&= \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]) \cdot \mathbf{P}(s, \alpha, s') \cdot \text{di}(\sigma)(\bar{\pi}, \alpha) \\
&\stackrel{IH}{=} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\bar{\pi}) \cdot \mathbf{P}(s, \alpha, s') \cdot \text{di}(\sigma)(\bar{\pi}, \alpha) \\
&= \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\mathcal{D}}}(\bar{\pi} \xrightarrow{\alpha} s').
\end{aligned}$$

**Case  $s \in \text{MS}$ :** According to our definition of paths it follows that  $\alpha = \perp$ . Further, we have

$$\begin{aligned}
\Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\perp} s']_{cyl}) &= \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]_{cyl} \cap [\bar{\pi} \xrightarrow{\perp} s']_{cyl}) \\
&= \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]_{cyl}) \cdot \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\perp} s']_{cyl} \mid [\bar{\pi}]_{cyl}). \tag{4.7}
\end{aligned}$$

Assume that a path  $\pi \in [\bar{\pi}]_{cyl}$  has been observed, i.e.,  $\text{pref}(\text{di}(\pi), m) = \bar{\pi}$  holds for some  $m \geq 0$ . The term  $\Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\perp} s']_{cyl} \mid [\bar{\pi}]_{cyl})$  coincides with the probability that also  $\text{pref}(\text{di}(\pi), m+1) = \bar{\pi} \xrightarrow{\perp} s'$  holds. We have either

- $s \neq s'$  which means that the transition from  $s$  to  $s'$  has to be taken during a period of  $\delta$  time units or
- $s = s'$  where we additionally have to consider the case that no transition is taken at  $s$  for  $\delta$  time units.

It follows that

$$\begin{aligned}
\Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\perp} s']_{cyl} \mid [\bar{\pi}]_{cyl}) &= \begin{cases} \mathbf{P}(s, \perp, s')(1 - e^{-\mathbf{E}(s)\delta}) & \text{if } s \neq s' \\ \mathbf{P}(s, \perp, s')(1 - e^{-\mathbf{E}(s)\delta}) + e^{-\mathbf{E}(s)\delta} & \text{if } s = s' \end{cases} \\
&= \mathbf{P}_{\delta}(s, \perp, s'). \tag{4.8}
\end{aligned}$$

We conclude that

$$\begin{aligned}
\Pr_{\sigma}^{\mathcal{M}}([\bar{\pi} \xrightarrow{\perp} s']_{cyl}) &\stackrel{4.7, 4.8}{=} \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]_{cyl}) \cdot \mathbf{P}_{\delta}(s, \perp, s') \\
&\stackrel{IH}{=} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(\bar{\pi}) \cdot \mathbf{P}_{\delta}(s, \perp, s') = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(\bar{\pi} \xrightarrow{\perp} s'). \quad \square
\end{aligned}$$

The lemma can be lifted to digitization step bounded until probabilities.

**Lemma 4.32**

Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$  and  $\sigma \in \text{GM}$ . Further, let  $\delta \in \mathbb{R}_{>0}$  and  $k \in \mathbb{N}$ . It holds that

$$\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\leq k} G) = \Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G]) \quad \blacksquare$$

*Proof.* Let  $\Pi^{\leq k}$  be the set of finite digital paths with at most  $k$  digitization steps that end at the first visit of a state in  $G$  while the remaining states are in  $H$ , i.e.,

$$\Pi^{\leq k} = \{\bar{\pi} = s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{n-1}} s_n \in FPaths^{\mathcal{M}_\delta} \mid |\bar{\pi}|_{\text{ds}} \leq k, s_n \in G, \text{ and } \forall i < n: s_i \in H \setminus G\}.$$

Every path in  $HU_{\text{ds}}^{\leq k} G$  has a unique prefix in  $\Pi^{\leq k}$ , yielding

$$HU_{\text{ds}}^{\leq k} G = \bigcup_{\bar{\pi} \in \Pi^{\leq k}} \text{Cyl}(\{\bar{\pi}\}).$$

For the corresponding paths of  $\mathcal{M}$  we obtain

$$[HU_{\text{ds}}^{\leq k} G] = \{\pi \in IPaths^{\mathcal{M}} \mid \text{di}(\pi) \text{ has a unique prefix in } \Pi^{\leq k}\} = \bigcup_{\bar{\pi} \in \Pi^{\leq k}} [\bar{\pi}]_{\text{cyl}}.$$

The claim follows since

$$\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\leq k} G) = \sum_{\bar{\pi} \in \Pi^{\leq k}} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\bar{\pi}) \stackrel{\text{Lem. 4.31}}{=} \sum_{\bar{\pi} \in \Pi^{\leq k}} \Pr_\sigma^{\mathcal{M}}([\bar{\pi}]_{\text{cyl}}) = \Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G]). \quad \square$$

The following proposition provides a lower bound for  $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$ .

**Proposition 4.33**

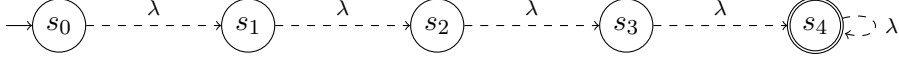
Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$  and  $\sigma \in \text{GM}$ . Further, let  $b, \delta \in \mathbb{R}_{>0}$  such that  $b/\delta = k \in \mathbb{N}$ . It holds that

$$\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\leq k} G) \leq \Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G). \quad \blacksquare$$

*Proof.* The Proposition is a direct consequence of Lemma 4.27 and Lemma 4.32.  $\square$

### 4.3.3 An Upper Bound for $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$

In the previous section, we established a lower bound for  $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$  by considering the paths in  $HU^{\leq b} G$  that require at most  $k = b/\delta$  digitization steps to reach  $G$ .

Figure 4.6: MA  $\mathcal{M}$  (cf. Example 4.34).

To establish an upper bound, we also consider the paths that require more than  $k$  digitization steps. We have

$$\Pr_{\sigma}^{\mathcal{M}}(HU^{\leq b} G) = \Pr_{\sigma}^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G]) + \Pr_{\sigma}^{\mathcal{M}}(HU^{\leq b} G \setminus [HU_{\text{ds}}^{\leq k} G]). \quad (4.9)$$

Lemma 4.32 yields  $\Pr_{\sigma}^{\mathcal{M}}([HU_{\text{ds}}^{\leq k} G]) = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(HU_{\text{ds}}^{\leq k} G)$ . The idea is to find an upper bound  $\varepsilon \in \mathbb{R}_{\geq 0}$  for the second term such that  $\Pr_{\sigma}^{\mathcal{M}}(HU^{\leq b} G \setminus [HU_{\text{ds}}^{\leq k} G]) \leq \varepsilon$ . Then, it follows that

$$\Pr_{\sigma}^{\mathcal{M}}(HU^{\leq b} G) \leq \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(HU_{\text{ds}}^{\leq k} G) + \varepsilon.$$

**Example 4.34**

Consider the MA  $\mathcal{M}$  from Figure 4.6 and the probability  $\Pr^{\mathcal{M}}(SU^{\leq 0.3} \{s_4\})$ . We assume the digitization constant  $\delta = 0.1$ , yielding the digitization step bound  $0.3/\delta = 3$ . However, it is not possible to reach  $s_4$  with less than four digitization steps, i.e.,  $\Pr^{\mathcal{M}}([SU_{\text{ds}}^{\leq 3} \{s_4\}]) = 0$ . The above-mentioned upper bound  $\varepsilon$  has to comprise the probability to take all four Markovian transitions leading to  $s_4$  within  $3\delta = 0.3$  time units. ■

Consider a path  $\pi \in HU^{\leq b} G \setminus [HU_{\text{ds}}^{\leq k} G]$ . Note that  $\pi$  reaches  $G$  via  $H$  within  $b$  time units but with more than  $k$  digitization steps. Hence, the prefix of  $\pi$  up to time point  $b$  certainly has more than  $k$  digitization steps, i.e.,  $\pi$  satisfies  $|pref_T(\pi, b)|_{\text{ds}} > k$ .

**Definition 4.35 (Paths with Digitization Step Bounded Prefixes)**

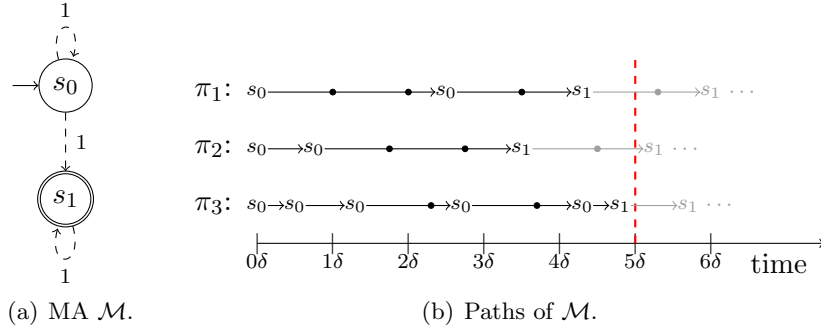
Assume an MA  $\mathcal{M}$  and a digitization constant  $\delta \in \mathbb{R}_{>0}$ . For some  $t \in \mathbb{R}_{\geq 0}$ ,  $j \in \mathbb{N}$ , and  $\triangleright \in \{<, \leq, >, \geq\}$ , we define the set of paths whose prefix up to time point  $t$  has  $\triangleright j$  digitization steps as

$$\#[t]^{\triangleright j} = \{\pi \in IPaths^{\mathcal{M}} \mid |pref_T(\pi, t)|_{\text{ds}} \triangleright j\}. \quad \blacksquare$$

Our observations above yield  $HU^{\leq b} G \setminus [HU_{\text{ds}}^{\leq k} G] \subseteq \#[b]^{\triangleright k}$  and therefore

$$\Pr_{\sigma}^{\mathcal{M}}(HU^{\leq b} G \setminus [HU_{\text{ds}}^{\leq k} G]) \leq \Pr_{\sigma}^{\mathcal{M}}(\#[b]^{\triangleright k}) = 1 - \Pr_{\sigma}^{\mathcal{M}}(\#[b]^{\leq k}). \quad (4.10)$$

Instead of an upper bound for  $1 - \Pr_{\sigma}^{\mathcal{M}}(\#[b]^{\leq k})$ , we establish a lower bound for  $\Pr_{\sigma}^{\mathcal{M}}(\#[k\delta]^{\leq k})$  (recall that we picked  $k$  and  $\delta$  such that  $b = k\delta$ ). The set  $\#[k\delta]^{\leq k}$  contains the paths with at most  $k$  digitization steps up to the first  $k\delta$  time units. Intuitively, these are the paths for which the number of digitization steps gives an accurate estimation for the actual sojourn times.

Figure 4.7: MA  $\mathcal{M}$  and illustration of paths of  $\mathcal{M}$  (cf. Example 4.36).**Example 4.36**

Let  $\mathcal{M}$  be the MA given in Figure 4.7(a). Assume  $k = 5$ , i.e., we consider the set  $\#[k\delta]^{\leq k} = \#[5\delta]^{\leq 5}$ . The digitization constant  $\delta$  remains unspecified in this example. Figure 4.7(b) illustrates paths  $\pi_1$ ,  $\pi_2$ , and  $\pi_3$  of  $\mathcal{M}$ . We depict sojourn times by arrow length (instead of arrow labels). For instance, the path  $\pi_1$  corresponds to  $s_0 \xrightarrow{2.5\delta} s_0 \xrightarrow{1.8\delta} s_0 \xrightarrow{1.7\delta} s_1 \xrightarrow{0.5\delta} s_1 \xrightarrow{0.5\delta} s_1 \dots \in IPaths^{\mathcal{M}}$ . Digitization steps that are “earned” by sojourning at some state for a multiple of  $\delta$  time units are indicated by black dots. Transitions of  $\pi_i$  (where  $i \in \{1, 2, 3\}$ ) that do not belong to  $pref_T(\pi_i, 5\delta)$  are depicted in gray. We obtain

$$\begin{aligned} pref_T(\pi_1, 5\delta) = 5 &\implies \pi_1 \in \#[5\delta]^{\leq 5} \\ pref_T(\pi_2, 5\delta) = 4 &\implies \pi_2 \in \#[5\delta]^{\leq 5} \\ pref_T(\pi_3, 5\delta) = 7 &\implies \pi_3 \notin \#[5\delta]^{\leq 5}. \end{aligned}$$

Note that only the digitization steps of the prefix up to time point  $5\delta$  are considered. For example, the step of  $\pi_2$  at time point  $4.5\delta$  is not considered since the corresponding transition is not part of  $pref_T(\pi_2, 5\delta)$ . However, we have  $|pref_T(\pi_2, 5.5\delta)|_{ds} = 6$ , implying  $\pi_2 \notin \#[5.5\delta]^{\leq 5}$ .

Although all considered paths reach  $G = \{s_1\}$  within  $5\delta$  time units, we remark that  $\pi_3 \notin \#[5\delta]^{\leq 5}$  requires more than  $k = 5$  digitization steps. ■

**Proposition 4.37**

Let  $\mathcal{M}$  be an MA,  $\sigma \in GM$ , and  $\lambda = \max\{E(s) \mid s \in MS\}$ . For each  $\delta \in \mathbb{R}_{>0}$  and  $k \in \mathbb{N}$  it holds that

$$\Pr_{\sigma}^{\mathcal{M}}(\#[k\delta]^{\leq k}) \geq (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k}. \quad \blacksquare$$

For the proof of the proposition we employ the following lemma.

**Lemma 4.38**

Let  $\mathcal{M}$  be an MA with  $\sigma \in GM$  and maximum rate  $\lambda = \max\{E(s) \mid s \in MS\}$ . For

each  $\delta \in \mathbb{R}_{>0}$ ,  $k \in \mathbb{N}$ , and  $t \in \mathbb{R}_{\geq 0}$  it holds that

$$\Pr_{\sigma}^{\mathcal{M}}(\#[k\delta + t]^{\leq k}) \geq \Pr_{\sigma}^{\mathcal{M}}(\#[k\delta]^{\leq k}) \cdot e^{-\lambda t}. \quad \blacksquare$$

*Proof.* First, we show that the set  $\#[k\delta + t]^{\leq k}$  corresponds to the paths of  $\#[k\delta]^{\leq k}$  with the additional requirement that no transition is taken between the time points  $k\delta$  and  $k\delta + t$ , i.e.,

$$\#[k\delta + t]^{\leq k} = \{\pi \in \#[k\delta]^{\leq k} \mid \text{there is no prefix } \pi' \text{ of } \pi \text{ with } k\delta < T(\pi') \leq k\delta + t\}.$$

“ $\subseteq$ ”: If  $\pi \in \#[k\delta + t]^{\leq k}$ , then  $\pi \in \#[k\delta]^{\leq k}$  follows immediately. Furthermore, assume towards a contradiction that there is a prefix  $\pi'$  of  $\pi$  with  $k\delta < T(\pi') \leq k\delta + t$ . Then,  $k < T(\pi')/\delta \leq |\pi'|_{\text{ds}}$  (cf. Lemma 4.19 on page 56). As  $T(\pi') \leq k\delta + t$ , this means that  $|pref_T(\pi, k\delta + t)|_{\text{ds}} \geq |\pi'|_{\text{ds}} > k$  which contradicts  $\pi \in \#[k\delta + t]^{\leq k}$ .

“ $\supseteq$ ”: For  $\pi \in \#[k\delta]^{\leq k}$  with no prefix  $\pi'$  such that  $k\delta < T(\pi') \leq k\delta + t$ , it holds that  $pref_T(\pi, k\delta + t) = pref_T(\pi, k\delta)$ . Hence,  $|pref_T(\pi, k\delta + t)|_{\text{ds}} = |pref_T(\pi, k\delta)|_{\text{ds}} \leq k$  and it follows that  $\pi \in \#[k\delta + t]^{\leq k}$ .

The probability for no transition to be taken between  $k\delta$  and  $k\delta + t$  only depends on the current state at time point  $k\delta$ . More precisely, for some state  $s \in \text{MS}$  assume the set of paths  $\{\pi \in \#[k\delta]^{\leq k} \mid last(pref_T(\pi, k\delta)) = s\}$ . The probability that a path in this set takes no transition between time points  $k\delta$  and  $k\delta + t$  is given by  $e^{-E(s)t}$ . With  $\lambda \geq E(s)$  for all  $s \in \text{MS}$  it follows that

$$\begin{aligned} & \Pr_{\sigma}^{\mathcal{M}}(\#[k\delta + t]^{\leq k}) \\ &= \Pr_{\sigma}^{\mathcal{M}}(\{\pi \in \#[k\delta]^{\leq k} \mid \text{there is no prefix } \pi' \text{ of } \pi \text{ with } k\delta < T(\pi') \leq k\delta + t\}) \\ &= \sum_{s \in \text{MS}} \Pr_{\sigma}^{\mathcal{M}}(\{\pi \in \#[k\delta]^{\leq k} \mid last(pref_T(\pi, k\delta)) = s\}) \cdot e^{-E(s)t} \\ &\geq \sum_{s \in \text{MS}} \Pr_{\sigma}^{\mathcal{M}}(\{\pi \in \#[k\delta]^{\leq k} \mid last(pref_T(\pi, k\delta)) = s\}) \cdot e^{-\lambda t} \\ &= \Pr_{\sigma}^{\mathcal{M}}(\#[k\delta]^{\leq k}) \cdot e^{-\lambda t}. \quad \square \end{aligned}$$

*Proof of Proposition 4.37.* Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_{\ell}\})$ . The proof is by induction over  $k$ . For  $k = 0$ , we have  $\pi \in \#[0 \cdot \delta]^{\leq 0}$  iff  $\pi$  takes no Markovian transition at time point zero. As this happens with probability one, it follows that

$$\Pr_{\sigma}^{\mathcal{M}}(\#[0 \cdot \delta]^{\leq 0}) = 1 = (1 + \lambda\delta)^0 \cdot e^{-\lambda\delta \cdot 0}.$$

We assume in the induction step that the proposition holds for some fixed  $k$ . We distinguish between two cases for the initial state  $s_0$  of  $\mathcal{M}$ .

Case  $s_0 \in \text{MS}$ : We partition the set  $\#[k\delta + \delta]^{\leq k+1} = \Lambda^{\geq \delta} \cup \Lambda^{< \delta}$  with

$$\begin{aligned}\Lambda^{\geq \delta} &= \{s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa_1} \dots \in \#[k\delta + \delta]^{\leq k+1} \mid t \geq \delta\} \text{ and} \\ \Lambda^{< \delta} &= \{s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa_1} \dots \in \#[k\delta + \delta]^{\leq k+1} \mid t < \delta\}.\end{aligned}$$

Hence,  $\Lambda^{\geq \delta}$  contains the paths where we wait at least  $\delta$  time units at  $s_0$  and  $\Lambda^{< \delta}$  contains the paths where the first transition is taken within  $t < \delta$  time units. It follows that  $\Pr_{\sigma}^{\mathcal{M}}(\#[k\delta + \delta]^{\leq k+1}) = \Pr_{\sigma}^{\mathcal{M}}(\Lambda^{\geq \delta}) + \Pr_{\sigma}^{\mathcal{M}}(\Lambda^{< \delta})$ . We consider the probabilities for  $\Lambda^{\geq \delta}$  and  $\Lambda^{< \delta}$  separately.

- $\Pr_{\sigma}^{\mathcal{M}}(\Lambda^{\geq \delta})$ : For a path  $s_0 \xrightarrow{t+\delta} s_1 \xrightarrow{\kappa_1} \dots \in \Lambda^{\geq \delta}$ , after the first  $\delta$  time units there are at most  $k$  digitization steps within the next  $k\delta$  time units, i.e.,

$$s_0 \xrightarrow{t+\delta} s_1 \xrightarrow{\kappa_1} \dots \in \Lambda^{\geq \delta} \iff s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa_1} \dots \in \#[k\delta]^{\leq k}.$$

The probability for  $\Lambda^{\geq \delta}$  can therefore be derived from the probability to wait at  $s_0$  for at least  $\delta$  time units and the probability for  $\#[k\delta]^{\leq k}$ . In order to apply this, we need to modify the considered scheduler as it might depend on the sojourn time in  $s_0$ . Let  $\sigma_{\delta}$  be the scheduler for  $\mathcal{M}$  that mimics  $\sigma$  on paths where the first transition is delayed by  $\delta$ , i.e.,  $\sigma_{\delta}$  satisfies

$$\sigma_{\delta}(s_0 \xrightarrow{t} \dots \xrightarrow{\kappa_{n-1}} s_n, \alpha) = \sigma(s_0 \xrightarrow{t+\delta} \dots \xrightarrow{\kappa_{n-1}} s_n, \alpha).$$

for all  $s_0 \xrightarrow{t} \dots \xrightarrow{\kappa_{n-1}} s_n \in \text{FPaths}^{\mathcal{M}}$  and  $\alpha \in \text{Act}$ . It holds that

$$\begin{aligned}\Pr_{\sigma}^{\mathcal{M}}(\Lambda^{\geq \delta}) &= e^{-\mathbb{E}(s_0)\delta} \cdot \Pr_{\sigma_{\delta}}^{\mathcal{M}}(\#[k\delta]^{\leq k}) \\ &\stackrel{IH}{\geq} e^{-\mathbb{E}(s_0)\delta} \cdot (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k} \\ &= e^{-\mathbb{E}(s_0)\delta} \cdot (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k} \cdot e^{-\lambda\delta} \cdot e^{\lambda\delta} \\ &= (1 + \lambda\delta)^k \cdot e^{-\lambda\delta(k+1)} \cdot e^{(\lambda - \mathbb{E}(s_0))\delta}.\end{aligned}\tag{4.11}$$

- $\Pr_{\sigma}^{\mathcal{M}}(\Lambda^{< \delta})$ : For a path  $s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa_1} \dots \in \Lambda^{< \delta}$ , the first digitization step happens at less than  $\delta$  time units, i.e.,  $0 \leq t < \delta$ . It follows that there are at most  $k$  digitization steps in the remaining  $k\delta + \delta - t$  time units, i.e.,

$$s_0 \xrightarrow{t} s_1 \xrightarrow{\kappa_1} s_2 \xrightarrow{\kappa_2} \dots \in \Lambda^{< \delta} \iff s_1 \xrightarrow{\kappa_1} s_2 \xrightarrow{\kappa_2} \dots \in \#^{s_1}[k\delta + \delta - t]^{\leq k},$$

where  $\#^{s_1}[k\delta + \delta - t]^{\leq k}$  refers to the paths  $\#[k\delta + \delta - t]^{\leq k}$  of  $\mathcal{M}^{s_1}$ , the MA obtained from  $\mathcal{M}$  by changing the initial state to  $s_1$ . Hence, the probability for  $\Lambda^{< \delta}$  can be derived from the probability to take a transition from  $s_0$  to some state  $s$  within  $t < \delta$  time units and the probability for  $\#^s[k\delta + \delta - t]^{\leq k}$ . Again, we need to adapt the considered scheduler. Let  $\pi \in \text{FPaths}^{\mathcal{M}}$  with  $\text{last}(\pi) = s$ .

The scheduler  $\sigma[\pi]$  for  $\mathcal{M}^s$  mimics the scheduler  $\sigma$  for  $\mathcal{M}$ , where  $\pi$  is prepended to the given path, i.e., we set

$$\sigma[\pi](s \xrightarrow{\kappa_j} \dots \xrightarrow{\kappa_{n-1}} s_n, \alpha) = \sigma(\pi \xrightarrow{\kappa_j} \dots \xrightarrow{\kappa_{n-1}} s_n, \alpha)$$

for all  $s \xrightarrow{\kappa_j} \dots \xrightarrow{\kappa_{n-1}} s_n \in FPaths^{\mathcal{M}^s}$  and  $\alpha \in Act$ . With Lemma 4.38 it follows that

$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\Lambda^{<\delta}) &= \int_0^{\delta} \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} \cdot \left( \sum_{s \in S} \mathbf{P}(s_0, \perp, s) \cdot \Pr_{\sigma[\pi]}^{\mathcal{M}^s}(\#^s[k\delta + \delta - t] \leq k) \right) dt \\ &\geq \int_0^{\delta} \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} \cdot \left( \sum_{s \in S} \mathbf{P}(s_0, \perp, s) \cdot \Pr_{\sigma[\pi]}^{\mathcal{M}^s}(\#^s[k\delta] \leq k) \cdot e^{-\lambda(\delta-t)} \right) dt \\ &\stackrel{IH}{\geq} \int_0^{\delta} \mathbf{E}(s_0) \cdot e^{-\mathbf{E}(s_0)t} \cdot \left( \sum_{s \in S} \mathbf{P}(s_0, \perp, s) \cdot (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k} \cdot e^{-\lambda(\delta-t)} \right) dt \\ &= (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k} \cdot \mathbf{E}(s_0) \cdot \int_0^{\delta} e^{-\mathbf{E}(s_0)t} \cdot e^{-\lambda(\delta-t)} \cdot \left( \sum_{s \in S} \mathbf{P}(s_0, \perp, s) \right) dt \\ &= (1 + \lambda\delta)^k \cdot e^{-\lambda\delta k} \cdot \mathbf{E}(s_0) \cdot \int_0^{\delta} e^{-\mathbf{E}(s_0)t} \cdot e^{-\lambda\delta} \cdot e^{\lambda t} dt \\ &= (1 + \lambda\delta)^k \cdot e^{-\lambda\delta(k+1)} \cdot \mathbf{E}(s_0) \cdot \int_0^{\delta} e^{(\lambda - \mathbf{E}(s_0))t} dt. \end{aligned} \quad (4.12)$$

Combining the two results for  $\Lambda^{\geq\delta}$  and  $\Lambda^{<\delta}$  (i.e., Equations 4.11 and 4.12), we obtain

$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\#[k\delta + \delta] \leq k+1) &= \Pr_{\sigma}^{\mathcal{M}}(\Lambda^{\geq\delta}) + \Pr_{\sigma}^{\mathcal{M}}(\Lambda^{<\delta}) \\ &\geq (1 + \lambda\delta)^k \cdot e^{-\lambda\delta(k+1)} \cdot \left( e^{(\lambda - \mathbf{E}(s_0))\delta} + \mathbf{E}(s_0) \cdot \int_0^{\delta} e^{(\lambda - \mathbf{E}(s_0))t} dt \right) \\ &\stackrel{*}{\geq} (1 + \lambda\delta)^k \cdot e^{-\lambda\delta(k+1)} \cdot (1 + \lambda\delta) \\ &= (1 + \lambda\delta)^{k+1} \cdot e^{-\lambda\delta(k+1)}, \end{aligned}$$

where the inequality marked with \* is due to

$$\begin{aligned}
& e^{(\lambda - \mathbb{E}(s_0))\delta} + \mathbb{E}(s_0) \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt \\
&= e^{(\lambda - \mathbb{E}(s_0))\delta} + (\mathbb{E}(s_0) - \lambda + \lambda) \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt \\
&= e^{(\lambda - \mathbb{E}(s_0))\delta} - (\lambda - \mathbb{E}(s_0)) \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt + \lambda \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt \\
&= \begin{cases} 1 - 0 + \lambda \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt & \text{if } \mathbb{E}(s_0) = \lambda \\ e^{(\lambda - \mathbb{E}(s_0))\delta} - (e^{(\lambda - \mathbb{E}(s_0))\delta} - 1) + \lambda \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt & \text{if } \mathbb{E}(s_0) < \lambda \end{cases} \\
&= 1 + \lambda \cdot \int_0^\delta e^{(\lambda - \mathbb{E}(s_0))t} dt \geq 1 + \lambda \cdot \int_0^\delta 1 dt = 1 + \lambda\delta .
\end{aligned}$$

**Case  $s_0 \in \text{PS}$ :** Since  $\mathcal{M}$  is non-zeno, a state  $s \in \text{MS}$  is reached from  $s_0$  within zero time almost surely (i.e., with probability one). From the previous case, it already follows that the Proposition holds for  $\mathcal{M}^s$  with  $s \in \text{MS}$  and the set  $\#^s[k\delta + \delta]^{\leq k+1}$ . With  $\Pi_{\text{MS}} = \{s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n \in \text{FPaths}^{\mathcal{M}} \mid s_n \in \text{MS} \text{ and } \forall i < n: s_i \in \text{PS}\}$  we obtain

$$\begin{aligned}
\Pr_\sigma^{\mathcal{M}}(\#[k\delta + \delta]^{\leq k+1}) &= \int_{\substack{\pi \in \Pi_{\text{MS}} \\ \text{last}(\pi) = s}} \Pr_{\sigma[\pi]}^{\mathcal{M}^s}(\#[k\delta + \delta]^{\leq k+1}) d\Pr_\sigma^{\mathcal{M}}(\pi) \\
&\geq \int_{\substack{\pi \in \Pi_{\text{MS}} \\ \text{last}(\pi) = s}} (1 + \lambda\delta)^{k+1} \cdot e^{-\lambda\delta(k+1)} d\Pr_\sigma^{\mathcal{M}}(\pi) \\
&= (1 + \lambda\delta)^{k+1} \cdot e^{-\lambda\delta(k+1)} \cdot \Pr_\sigma^{\mathcal{M}}(\Pi_{\text{MS}}) \\
&= (1 + \lambda\delta)^{k+1} \cdot e^{-\lambda\delta(k+1)} . \quad \square
\end{aligned}$$

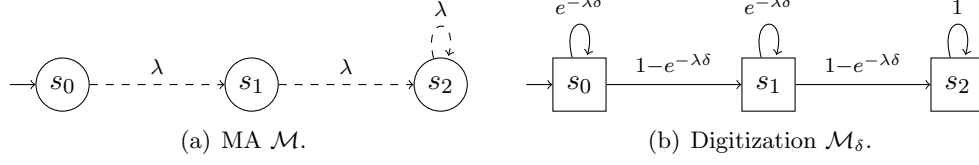
We conclude our results for the approximation of  $\Pr_\sigma^{\mathcal{M}}(H\mathcal{U}^{\leq b}G)$  with the following proposition.

**Proposition 4.39**

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$ ,  $\sigma \in \text{GM}$ , and maximum rate  $\lambda = \max\{\mathbb{E}(s) \mid s \in \text{MS}\}$ . Further, let  $b, \delta \in \mathbb{R}_{>0}$  such that  $b/\delta = k \in \mathbb{N}$ . It holds that

$$\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(H\mathcal{U}_{\text{ds}}^{\leq k}G) \leq \Pr_\sigma^{\mathcal{M}}(H\mathcal{U}^{\leq b}G) \leq \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(H\mathcal{U}_{\text{ds}}^{\leq k}G) + 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b} . \quad \blacksquare$$

*Proof.* The first inequality is due to Proposition 4.33. The second inequality is a result of Proposition 4.37 and Equations 4.9 and 4.10 from the beginning of this section.  $\square$

Figure 4.8: MA  $\mathcal{M}$  with digitization  $\mathcal{M}_\delta$  (cf. Example 4.40).

The term  $1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b}$  decreases when a smaller digitization constant  $\tilde{\delta} \in \mathbb{R}_{>0}$  with  $\tilde{\delta} < \delta$  and  $b/\tilde{\delta} = \tilde{k} \in \mathbb{N}$  is considered, i.e.,

$$1 - (1 + \lambda\tilde{\delta})^{\tilde{k}} \cdot e^{-\lambda b} < 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b}.$$

Hence, the value  $\Pr_\sigma^{\mathcal{M}}(HU^{\leq b} G)$  can be approximated with arbitrary precision by computing  $\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\leq k} G)$  for a sufficiently small  $\delta \in \mathbb{R}_{>0}$ . The following example indicates that the bounds given in Proposition 4.39 are tight.

**Example 4.40**

Let  $\mathcal{M}$  be the MA with digitization  $\mathcal{M}_\delta$  as shown in Figure 4.8. We consider the time-bounded reachability probabilities  $\Pr_\sigma^{\mathcal{M}}(\diamond^{\leq b}\{s_1\})$  and  $\Pr_\sigma^{\mathcal{M}}(\diamond^{\leq b}\{s_2\})$ , where  $\sigma$  is the (unique) scheduler of  $\mathcal{M}$ . We set  $\delta = b$  which yields the digitization step-bound  $k = b/\delta = 1$  for both probabilities. It holds that

$$\Pr_\sigma^{\mathcal{M}}(\diamond^{\leq b}\{s_1\}) = 1 - e^{-\lambda b} = 1 - e^{-\lambda\delta} = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\diamond_{\text{ds}}^{\leq k}\{s_1\}).$$

Furthermore, we have

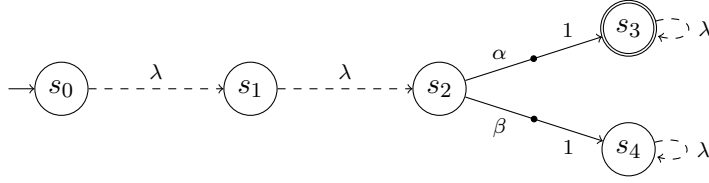
$$\Pr_\sigma^{\mathcal{M}}(\diamond^{\leq b}\{s_2\}) = 1 - (1 + \lambda b) \cdot e^{-\lambda b} = \underbrace{\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(\diamond_{\text{ds}}^{\leq k}\{s_2\})}_{=0} + 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b}. \blacksquare$$

**Remark 4.41**

[GHH<sup>+</sup>14, Theorem 5.3] states that

$$\begin{aligned} & \sup_{\sigma \in \text{TA}^{\mathcal{M}_\delta}} \Pr_\sigma^{\mathcal{M}_\delta}(\diamond_{\text{ds}}^{\leq b} G) \\ & \leq \sup_{\sigma \in \text{GM}^{\mathcal{M}}} \Pr_\sigma^{\mathcal{M}}(\diamond^{\leq b} G) \\ & \leq \sup_{\sigma \in \text{TA}^{\mathcal{M}_\delta}} \Pr_\sigma^{\mathcal{M}_\delta}(\diamond_{\text{ds}}^{\leq b} G) + 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b} \end{aligned}$$

holds. Besides the extension to bounded until probabilities, Proposition 4.39 generalizes this result by explicitly referring to the schedulers  $\sigma \in \text{GM}^{\mathcal{M}}$  and  $\text{di}(\sigma) \in \text{TA}^{\mathcal{M}_\delta}$  under which the claim holds. This extension is necessary to conduct the analysis of trade-offs between multiple bounded until objectives for  $\mathcal{M}$  on the digitization  $\mathcal{M}_\delta$ . Further details are given in Section 4.3.5.

Figure 4.9: MA  $\mathcal{M}$  (cf. Remark 4.41).

We also remark that the proof given by the authors of [GHH<sup>+</sup>14, Theorem 5.3] can not be adapted to show Proposition 4.39. The main reason is that the proof relies on an auxiliary lemma which claims that<sup>2</sup>

$$\Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} G \mid \#[\delta]^{<2}) \leq \Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} G) \quad (4.13)$$

holds for *all* schedulers  $\sigma \in \text{GM}^{\mathcal{M}}$ . We show that this claim does *not* hold. The intuition is as follows. Assume we observe that at most one Markovian transition is taken in  $\mathcal{M}$  within the first  $\delta$  time units (i.e., we observe a path in  $\#[\delta]^{<2}$ ). The lemma claims that under this observation the probability to reach  $G$  within  $b$  time units does not increase. We give a counterexample to illustrate that there are schedulers for which this is not true. Consider the MA  $\mathcal{M}$  from Figure 4.9 and let  $\sigma$  be the scheduler for  $\mathcal{M}$  satisfying

$$\sigma(s_0 \xrightarrow{t_1} s_1 \xrightarrow{t_2} s_2, \alpha) = \begin{cases} 1 & \text{if } t_1 + t_2 > \delta \\ 0 & \text{otherwise.} \end{cases}$$

Hence,  $\sigma$  chooses  $\alpha$  iff there are less than two digitization steps within the first  $\delta$  time units. It follows that the probability to reach  $G = \{s_3\}$  on a path in  $\#[\delta]^{\geq 2}$  is zero. We conclude that

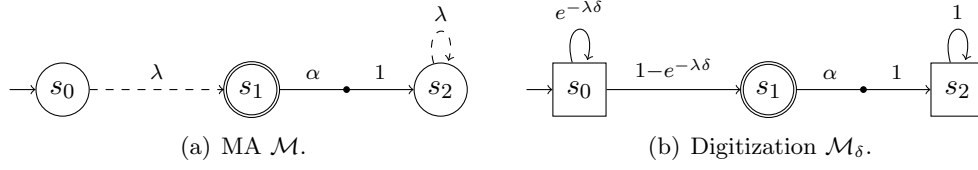
$$\begin{aligned} \Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} \{s_3\}) &= \Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} \{s_3\} \cap \#[\delta]^{<2}) + \underbrace{\Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} \{s_3\} \cap \#[\delta]^{\geq 2})}_{=0} \\ &= \Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} \{s_3\} \mid \#[\delta]^{<2}) \cdot \underbrace{\Pr_{\sigma}^{\mathcal{M}}(\#[\delta]^{<2})}_{<1} < \Pr_{\sigma}^{\mathcal{M}}(\diamond^{\leq b} \{s_3\} \mid \#[\delta]^{<2}) \end{aligned}$$

which contradicts Equation 4.13. ■

#### 4.3.4 Until Probabilities with Lower Time-bounds

We extend the results of Section 4.3.2 and Section 4.3.3 to bounded until probabilities  $\Pr_{\sigma}^{\mathcal{M}}(H\mathcal{U}^I G)$  where the time interval  $I$  considers a non-zero lower time bound, i.e., either  $I = [a, b]$  or  $I = [a, \infty)$  for  $0 < a < b$ .

<sup>2</sup>We adapt [GHH<sup>+</sup>14, Lemma G.2] to our notations.

Figure 4.10: MA  $\mathcal{M}$  with digitization  $\mathcal{M}_\delta$  (cf. Example 4.42).**Example 4.42**

Let  $\mathcal{M}$  be the MA with digitization  $\mathcal{M}_\delta$  as illustrated in Figure 4.10. With  $\delta = 0.1$  we obtain

$$\begin{aligned} \Pr^{\mathcal{M}}(\{s_0\} \mathcal{U}^{[0.2, 0.3]} \{s_1\}) &= \Pr^{\mathcal{M}}(\{s_0 \xrightarrow{t} s_1 \xrightarrow{\alpha} \dots \in \text{IPaths}^{\mathcal{M}} \mid 0.2 \leq t \leq 0.3\}) \\ &= \Pr^{\mathcal{M}}([s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_0 \xrightarrow{\perp} s_1]) \\ &= \Pr^{\mathcal{M}}([\{s_0\} \mathcal{U}_{\text{ds}}^{(2,3)} \{s_1\}]) = \Pr^{\mathcal{M}_\delta}(\{s_0\} \mathcal{U}_{\text{ds}}^{(2,3)} \{s_1\}), \end{aligned}$$

where we omit the unique scheduler for  $\mathcal{M}$  and  $\mathcal{M}_\delta$  from the notation.  $\blacksquare$

In the example, the given time-bounded until probability is expressed as a digitization step bounded until probability. In general, the digitization approach only yields an estimate of the actual time-bounded until probability. The following proposition provides a lower and an upper bound for probabilities of the form  $\Pr_\sigma^{\mathcal{M}}(HU^{[a,b]} G)$ .

**Proposition 4.43**

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$ ,  $\sigma \in \text{GM}$ , and maximum rate  $\lambda = \max\{E(s) \mid s \in \text{MS}\}$ . Further, let  $[a, b]$  be a time interval with  $0 < a < b$  and let  $\delta \in \mathbb{R}_{>0}$  such that  $a/\delta = j \in \mathbb{N}$  and  $b/\delta = k \in \mathbb{N}$ . It holds that

$$\begin{aligned} &\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{(j,k)} G) - 1 + (1 + \lambda\delta)^j \cdot e^{-\lambda a} \\ &\leq \Pr_\sigma^{\mathcal{M}}(HU^{[a,b]} G) \\ &\leq \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{(j,k)} G) + 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b} + 1 - e^{-\lambda\delta}. \end{aligned} \quad \blacksquare$$

*Proof.* We have

$$HU^{[a,b]} G \cup ([HU_{\text{ds}}^{(j,k)} G] \setminus HU^{[a,b]} G) = [HU_{\text{ds}}^{(j,k)} G] \cup (HU^{[a,b]} G \setminus [HU_{\text{ds}}^{(j,k)} G]).$$

It follows that the probability for  $HU^{[a,b]} G$  is given by

$$\begin{aligned} \Pr_\sigma^{\mathcal{M}}(HU^{[a,b]} G) &= \Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{(j,k)} G]) + \Pr_\sigma^{\mathcal{M}}(HU^{[a,b]} G \setminus [HU_{\text{ds}}^{(j,k)} G]) \\ &\quad - \Pr_\sigma^{\mathcal{M}}([HU_{\text{ds}}^{(j,k)} G] \setminus HU^{[a,b]} G). \end{aligned} \quad (4.14)$$

The rest of the proof consists of showing the (in)equalities

$$\Pr_{\sigma}^{\mathcal{M}}([H\mathcal{U}_{\text{ds}}^{(j,k)} G]) = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(H\mathcal{U}_{\text{ds}}^{(j,k)} G), \quad (4.15)$$

$$\Pr_{\sigma}^{\mathcal{M}}([H\mathcal{U}_{\text{ds}}^{(j,k)} G] \setminus H\mathcal{U}^{[a,b]} G) \leq 1 - (1 + \lambda\delta)^j \cdot e^{-\lambda a}, \text{ and} \quad (4.16)$$

$$\Pr_{\sigma}^{\mathcal{M}}(H\mathcal{U}^{[a,b]} G \setminus [H\mathcal{U}_{\text{ds}}^{(j,k)} G]) \leq 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b} + 1 - e^{-\lambda\delta}. \quad (4.17)$$

The proposition follows by considering Equation 4.14 and Equation 4.15 together with either Inequality 4.16 (for the lower bound) or Inequality 4.17 (for the upper bound).

**Equation 4.15:** We proceed similar to the proof of Lemma 4.32 on page 63. Consider the set  $\Pi^{(j,k)}$  of finite digital paths given by

$$\begin{aligned} \Pi^{(j,k)} = \{ & \bar{\pi} = s_0 \xrightarrow{\alpha_0} \dots \xrightarrow{\alpha_{n-1}} s_n \in FPaths^{\mathcal{M}_{\delta}} \mid j < |\bar{\pi}|_{\text{ds}} \leq k, s_n \in G, \\ & \forall i < n: s_i \in H, \text{ and } \forall i < n: s_i \in G \text{ implies } |pref(\bar{\pi}, i)|_{\text{ds}} \leq j \}. \end{aligned}$$

Every path in  $H\mathcal{U}_{\text{ds}}^{(j,k)} G$  has a unique prefix in  $\Pi^{(j,k)}$ , yielding

$$H\mathcal{U}_{\text{ds}}^{(j,k)} G = \bigcup_{\bar{\pi} \in \Pi^{(j,k)}} Cyl(\{\bar{\pi}\}) \quad \text{and} \quad [H\mathcal{U}_{\text{ds}}^{(j,k)} G] = \bigcup_{\bar{\pi} \in \Pi^{(j,k)}} [\bar{\pi}]_{\text{cyl}}.$$

Equation 4.15 follows with Lemma 4.31 since

$$\Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(H\mathcal{U}_{\text{ds}}^{(j,k)} G) = \sum_{\bar{\pi} \in \Pi^{(j,k)}} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(\bar{\pi}) = \sum_{\bar{\pi} \in \Pi^{(j,k)}} \Pr_{\sigma}^{\mathcal{M}}([\bar{\pi}]_{\text{cyl}}) = \Pr_{\sigma}^{\mathcal{M}}([H\mathcal{U}_{\text{ds}}^{(j,k)} G]).$$

**Inequality 4.16:** We show that

$$[H\mathcal{U}_{\text{ds}}^{(j,k)} G] \setminus H\mathcal{U}^{[a,b]} G \subseteq \#[a]^{>j}.$$

The claim follows directly with  $\Pr_{\sigma}^{\mathcal{M}}(\#[a]^{>j}) = 1 - \Pr_{\sigma}^{\mathcal{M}}(\#[j\delta]^{\leq j})$  and Proposition 4.37. Consider a path  $\pi \in [H\mathcal{U}_{\text{ds}}^{(j,k)} G] \setminus H\mathcal{U}^{[a,b]} G$ . Since  $\pi \in [H\mathcal{U}_{\text{ds}}^{(j,k)} G]$ , a state in  $G$  is reached via  $H$  with at most  $k$  digitization steps and therefore within at most  $b$  time units, i.e.,  $\pi \in H\mathcal{U}^{\leq b} G$ . As  $\pi \notin H\mathcal{U}^{[a,b]} G$ , we conclude that  $\pi$  has to reach (and leave)  $G$  within less than  $a$  time units. Let  $\bar{\pi} \in \Pi^{(j,k)}$  be the unique digital path with  $\pi \in [\bar{\pi}]_{\text{cyl}}$ . Our observations yield that  $\pi$  leaves  $last(\bar{\pi}) \in G$  before time point  $a$ . Put differently,  $\bar{\pi}$  is a prefix of  $\text{di}(pref_T(\pi, a))$ . Hence,  $|pref_T(\pi, a)|_{\text{ds}} \geq |\bar{\pi}|_{\text{ds}} > j$  which implies  $\pi \in \#[a]^{>j}$ .

**Inequality 4.17:** Consider some path  $\pi \in H\mathcal{U}^{[a,b]} G \setminus [H\mathcal{U}_{\text{ds}}^{(j,k)} G]$  and let  $\pi' = s_0 \xrightarrow{\kappa_0} \dots \xrightarrow{\kappa_{n-1}} s_n$  be the largest prefix of  $\pi$  such that  $s_n \in G$ ,  $s_i \in H$  for all  $i < n$ , and  $T(\pi') \leq b$ . Such a prefix exists due to  $\pi \in H\mathcal{U}^{[a,b]} G$ . We distinguish two cases.

- If  $|\pi'|_{\text{ds}} > k$ , then  $\pi \in \#[b]^{>k}$  since  $|\text{pref}_T(\pi, b)|_{\text{ds}} \geq |\pi'|_{\text{ds}} > k$ .
- If  $|\pi'|_{\text{ds}} \leq k$ , then  $|\pi'|_{\text{ds}} \leq j$  holds due to  $\pi \notin [H\mathcal{U}_{\text{ds}}^{(j,k)} G]$ . Let  $\pi' \xrightarrow{\kappa} s$  be the prefix of  $\pi$  of length  $|\pi'|+1$ . We show by contradiction that  $a \leq T(\pi' \xrightarrow{\kappa} s) < a+\delta$  holds:
  - If  $T(\pi' \xrightarrow{\kappa} s) < a$ , then  $\text{last}(\pi') \in G$  is left before time point  $a$  which contradicts  $\pi \in H\mathcal{U}^{[a,b]} G$ .
  - Further, assume that  $T(\pi' \xrightarrow{\kappa} s) \geq a + \delta$ . With Lemma 4.19 on page 56 we obtain

$$\begin{aligned} t(\kappa) &\geq a + \delta - T(\pi') \\ &\geq a + \delta - |\pi'|_{\text{ds}} \cdot \delta \\ &\geq (j + 1 - \underbrace{|\pi'|_{\text{ds}}}_{\leq j}) \cdot \delta > 0 . \end{aligned}$$

Hence,  $\pi$  stays at  $\text{last}(\pi')$  for at least  $(j+1-|\pi'|_{\text{ds}}) \cdot \delta$  time units which means that  $\text{di}(\pi')(\xrightarrow{\perp} \text{last}(\pi'))^{j+1-|\pi'|_{\text{ds}}} = \bar{\pi}$  is a prefix of  $\text{di}(\pi)$ . Since  $|\bar{\pi}|_{\text{ds}} = j+1$ , it follows that  $\bar{\pi} \in \Pi^{(j,k)}$ , contradicting  $\pi \notin [H\mathcal{U}_{\text{ds}}^{(j,k)} G]$ .

We infer that  $\pi$  takes at least one transition in the time interval  $[a, a + \delta)$ , i.e.,  $\text{pref}_T(\pi, a) \neq \text{pref}_T(\pi, a + \delta)$ .

Since at least one of the two cases above is true for every path in the set  $H\mathcal{U}^{[a,b]} G \setminus [H\mathcal{U}_{\text{ds}}^{(j,k)} G]$ , it holds that

$$H\mathcal{U}^{[a,b]} G \setminus [H\mathcal{U}_{\text{ds}}^{(j,k)} G] \subseteq \#[b]^{>k} \cup \{\pi \in \text{IPaths}^{\mathcal{M}} \mid \text{pref}_T(\pi, a) \neq \text{pref}_T(\pi, a + \delta)\}.$$

The probability  $\Pr_{\sigma}^{\mathcal{M}}(\#[b]^{>k})$  is at most  $1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b}$  due to Proposition 4.37. The probability for a path to take at least one transition in the time interval  $[a, a + \delta)$  can be upper bounded by  $1 - e^{-\lambda\delta}$ . Inequality 4.17 follows by considering the sum of these values.  $\square$

A similar result holds for (time-)bounded until probabilities that only consider a lower time bound.

#### Proposition 4.44

Let  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1 \dots, \rho_\ell\})$  be an MA with  $H, G \subseteq S$ ,  $\sigma \in \text{GM}$ , and maximum rate  $\lambda = \max\{E(s) \mid s \in \text{MS}\}$ . Further, let  $a, \delta \in \mathbb{R}_{>0}$  such that  $a/\delta = j \in \mathbb{N}$ . It holds that

$$\begin{aligned} &\Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(H\mathcal{U}_{\text{ds}}^{>j} G) - 1 + (1 + \lambda\delta)^j \cdot e^{-\lambda a} \\ &\leq \Pr_{\sigma}^{\mathcal{M}}(H\mathcal{U}^{\geq a} G) \\ &\leq \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(H\mathcal{U}_{\text{ds}}^{>j} G) + 1 - e^{-\lambda\delta} . \end{aligned} \quad \blacksquare$$

We omit a formal proof as it is similar to the proof of Proposition 4.43.

### 4.3.5 Lifting to Multiple Bounded Until Objectives

Proposition 4.39, Proposition 4.43, and Proposition 4.44 provide the basis for the analysis of bounded until objectives. To lift these results to multiple objectives, we have to incorporate that the digitization approach only yields an estimation of the desired values. Given a time interval  $I$  and a digitization constant  $\delta \in \mathbb{R}_{>0}$ , let  $\text{di}(I)$  denote the resulting digitization step bounds, i.e.,

$$\text{di}(I) = \begin{cases} \{0, 1, \dots, k\} & \text{if } I = [0, b] \text{ and } k = b/\delta \in \mathbb{N} \\ \{j + 1, j + 2, \dots\} & \text{if } I = [a, \infty) \text{ and } j = a/\delta \in \mathbb{N} \\ \{j + 1, j + 2, \dots, k\} & \text{if } I = [a, b], k = b/\delta \in \mathbb{N}, \text{ and } j = a/\delta \in \mathbb{N}. \end{cases}$$

We define values  $\varepsilon_i^\downarrow$  and  $\varepsilon_i^\uparrow$  for a given objective  $\mathbb{O}_i = \mathbb{P}(HU^I G)$  such that for any scheduler  $\sigma \in \text{GM}$  we have

$$\Pr_\sigma^{\mathcal{M}}(HU^I G) \in [p_i - \varepsilon_i^\downarrow, p_i + \varepsilon_i^\uparrow], \quad (4.18)$$

where  $p_i = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\text{di}(I)} G)$ . Let  $\lambda = \max\{E(s) \mid s \in \text{MS}\}$  be the maximum rate occurring in the MA  $\mathcal{M}$ .

- If  $\mathbb{O}_i = \mathbb{P}(HU^{\leq b} G)$ , then Proposition 4.39 yields

$$\varepsilon_i^\downarrow = 0 \text{ and } \varepsilon_i^\uparrow = 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b}.$$

- If  $\mathbb{O}_i = \mathbb{P}(HU^I G)$ , then Proposition 4.43 yields

$$\begin{aligned} \varepsilon_i^\downarrow &= 1 - (1 + \lambda\delta)^j \cdot e^{-\lambda a} \text{ and} \\ \varepsilon_i^\uparrow &= 1 - (1 + \lambda\delta)^k \cdot e^{-\lambda b} + 1 - e^{-\lambda\delta}. \end{aligned}$$

- If  $\mathbb{O}_i = \mathbb{P}(HU^{[a,b]} G)$ , then Proposition 4.44 yields

$$\varepsilon_i^\downarrow = 1 - (1 + \lambda\delta)^j \cdot e^{-\lambda a} \text{ and } \varepsilon_i^\uparrow = 1 - e^{-\lambda\delta}.$$

The notation is lifted to multiple objectives: For a list of bounded until objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  and some point  $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{R}^d$  we consider the set

$$\varepsilon(\mathbb{O}, \mathbf{p}) = \bigtimes_{1 \leq i \leq d} [p_i - \varepsilon_i^\downarrow, p_i + \varepsilon_i^\uparrow] \subseteq \mathbb{R}^d.$$

**Example 4.45**

Let  $\mathbb{O} = (\mathbb{O}_1, \mathbb{O}_2)$  be two bounded until objectives such that  $\varepsilon_1^\downarrow = \varepsilon_2^\downarrow = 0.15$  and  $\varepsilon_1^\uparrow = \varepsilon_2^\uparrow = 0.25$ . The blue rectangle in Figure 4.11 illustrates the set  $\varepsilon(\mathbb{O}, \mathbf{p})$  for  $\mathbf{p} = (0.5, 0.5)$ . ■

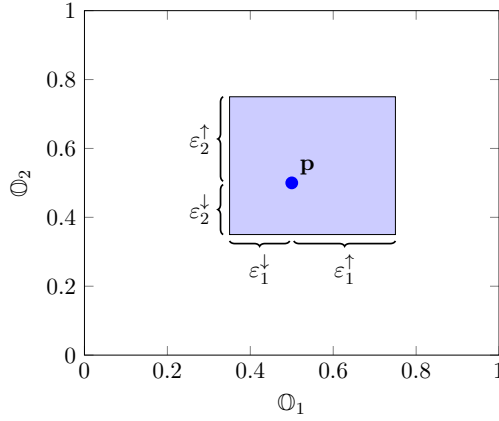


Figure 4.11: Illustration of the set  $\varepsilon(\mathbb{O}, \mathbf{p})$ . (cf. Example 4.45).

We define objectives that refer to the probability of digitization step bounded paths as in Definition 4.25 on page 58.

**Definition 4.46 (ds-bounded Until Objective)**

Let  $\mathcal{M}_\delta$  be a digitization of an MA with scheduler  $\sigma \in \text{TA}^{\mathcal{M}_\delta}$  and subsets of states  $H$  and  $G$ . Further, let  $J \subseteq \mathbb{N}$  denote digitization step bounds. A *ds-bounded until objective* has the form  $\mathbb{P}(HU_{\text{ds}}^J G)$  and satisfaction w.r.t. a threshold  $\triangleright_i p_i$  is defined by

$$\mathcal{M}_\delta, \sigma \models \mathbb{P}(HU_{\text{ds}}^J G) \triangleright_i p_i \iff \text{Pr}_\sigma^{\mathcal{M}_\delta}(HU_{\text{ds}}^J G) \triangleright_i p_i. \quad \blacksquare$$

For the sake of uniformity, we say that a digitization  $\mathcal{M}_\delta$  satisfies a (time-)bounded until objective  $\mathbb{P}(HU^I G)$  iff the corresponding ds-bounded until objective  $\mathbb{P}(HU_{\text{ds}}^{\text{di}(I)} G)$  is satisfied, i.e.,

$$\mathcal{M}_\delta, \sigma \models \mathbb{P}(HU^I G) \triangleright_i p_i \iff \mathcal{M}_\delta, \sigma \models \mathbb{P}(HU_{\text{ds}}^{\text{di}(I)} G) \triangleright_i p_i.$$

Here we assume that  $\text{di}(I)$  is well-defined, i.e., the digitization constant  $\delta$  induces natural numbers for the digitization step bounds.

The following example illustrates how the set of achievable points of  $\mathcal{M}$  can be approximated when given the set of achievable points of  $\mathcal{M}_\delta$ .

**Example 4.47**

Let  $\mathcal{M}$  be an MA with digitization  $\mathcal{M}_\delta$  and consider two bounded until objectives  $\mathbb{O} = (\mathbb{O}_1, \mathbb{O}_2)$  with threshold relations  $\triangleright = \{\geq, \geq\}$ . The gray area<sup>3</sup> in Figure 4.12(a) denotes the set of achievable points of  $\mathcal{M}_\delta$  given by  $A = \{\mathbf{p} \in \mathbb{R}^2 \mid \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\}$ .

For the point  $\mathbf{p} = (0.5, 0.5) \in A$ , the blue rectangle illustrates the set  $\varepsilon(\mathbb{O}, \mathbf{p})$ . From Equation 4.18 we infer that  $\varepsilon(\mathbb{O}, \mathbf{p})$  contains at least one point  $\mathbf{p}'$  that is achievable for

<sup>3</sup>In the figure,  $A^-$  partly overlaps  $A$ , i.e., the green area also belongs to  $A$ .

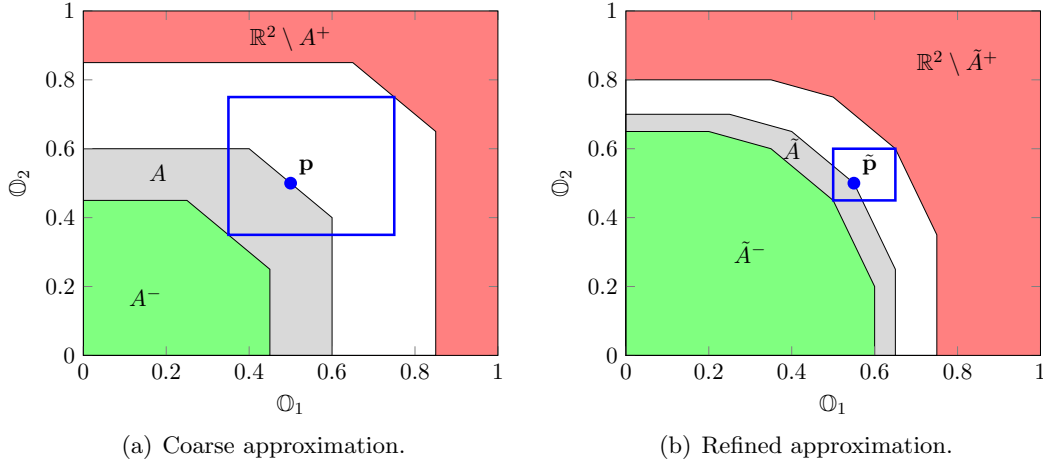


Figure 4.12: Illustration of the approximation of achievable points (cf. Example 4.47).

the MA  $\mathcal{M}$ . Hence, the bottom left corner point of the rectangle certainly is achievable for  $\mathcal{M}$ . As this observation holds for any  $\mathbf{q} \in A$  and the corresponding set  $\varepsilon(\mathbb{O}, \mathbf{q})$ , we obtain that any point in  $A^-$  (depicted by the green area) is achievable.

On the other hand, an achievable point of  $\mathcal{M}$  has to be contained in a set  $\varepsilon(\mathbb{O}, \mathbf{q})$  for at least one  $\mathbf{q} \in A$ . The red area depicts the set of points  $\mathbb{R}^d \setminus A^+$  for which this is not the case. It follows that  $A^-$  is an under-approximation and  $A^+$  is an over-approximation of the set of achievable points of  $\mathcal{M}$ .

The digitization constant  $\delta$  controls the accuracy of the resulting approximation. Figure 4.12(b) depicts a possible result when a smaller digitization constant  $\tilde{\delta} < \delta$  is considered. ■

The observations from the example above are formalized in the following theorem which lifts Proposition 4.39, Proposition 4.43, and Proposition 4.44 to multiple objectives. The theorem considers two sets  $A^-$  and  $A^+$  (as in Example 4.47) and states that the points in  $A^-$  are achievable (for  $\mathcal{M}$ ) while the points in  $\mathbb{R}^d \setminus A^+$  are not. We refer to  $A^-$  as *under-approximation* and to  $A^+$  as *over-approximation* of the set of achievable points of  $\mathcal{M}$ .

#### Theorem 4.48

Let  $\mathcal{M}$  be an MA with digitization  $\mathcal{M}_\delta$ . Furthermore, let  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  be a list of bounded until objectives with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . For the sets

$$A^- = \{\mathbf{p}' \in \mathbb{R}^d \mid \forall \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ implies } \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\} \text{ and}$$

$$A^+ = \{\mathbf{p}' \in \mathbb{R}^d \mid \exists \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ and } \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\}$$

it holds that

$$A^- \subseteq \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})\} \subseteq A^+ . \quad \blacksquare$$

*Proof.* For simplicity, we assume that only the threshold relation  $\geq$  is considered, i.e.,  $\triangleright = (\geq, \dots, \geq)$ . The remaining cases are treated analogously.

First assume a point  $\mathbf{p}' = (p_1', \dots, p_d') \in A^-$ . Consider the point  $\mathbf{p} = (p_1, \dots, p_d)$  satisfying  $p_i' = p_i - \varepsilon_i^\downarrow$  for each index  $i$ . It follows that  $\mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p})$  and thus  $\mathcal{M}_\delta, \sigma \models \mathbb{O} \triangleright \mathbf{p}'$  for some scheduler  $\sigma \in \text{TA}^{\mathcal{M}_\delta}$ . For an index  $i$  let  $\mathbb{O}_i$  be the objective  $\mathbb{P}(HU^I G)$  with corresponding ds-bounded objective  $\mathbb{P}(HU_{\text{ds}}^{\text{di}(I)} G)$ . It follows that

$$\mathcal{M}_\delta, \sigma \models \mathbb{O}_i \geq p_i \iff \mathcal{M}_\delta, \text{di}(\sigma) \models \mathbb{O}_i \geq p_i \iff \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\text{di}(I)} G) \geq p_i,$$

where the first equivalence is due to the observation  $\sigma = \text{di}(\sigma)$  for any  $\sigma \in \text{TA}^{\mathcal{M}_\delta}$ . With Equation 4.18 on page 75 it follows that

$$p_i' = p_i - \varepsilon_i^\downarrow \leq \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\text{di}(I)} G) - \varepsilon_i^\downarrow \stackrel{4.18}{\leq} \Pr_\sigma^{\mathcal{M}}(HU^I G).$$

As this observation holds for all objectives in  $\mathbb{O}$ , it follows that  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p}'$ , implying  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p}')$ .

The proof of the second inclusion is similar. Assume that  $\mathcal{M}, \sigma \models \mathbb{O} \triangleright \mathbf{p}'$  holds for a point  $\mathbf{p}' = (p_1', \dots, p_d') \in \mathbb{R}^d$  and a scheduler  $\sigma \in \text{GM}^{\mathcal{M}}$ . For some index  $i$ , consider  $\mathbb{O}_i = \mathbb{P}(HU^I G)$  with ds-bounded objective  $\mathbb{P}(HU_{\text{ds}}^{\text{di}(I)} G)$ . It follows that  $\Pr_\sigma^{\mathcal{M}}(HU^I G) \geq p_i'$ . With Equation 4.18 we obtain

$$p_i' - \varepsilon_i^\uparrow \leq \Pr_\sigma^{\mathcal{M}}(HU^I G) - \varepsilon_i^\uparrow \stackrel{4.18}{\leq} \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HU_{\text{ds}}^{\text{di}(I)} G).$$

Applying this for all objectives in  $\mathbb{O}$  yields  $\mathcal{M}_\delta, \text{di}(\sigma) \models \mathbb{O} \triangleright \mathbf{p}$ , where the point  $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{R}^d$  satisfies  $p_i = p_i' - \varepsilon_i^\uparrow$  or, equivalently,  $p_i' = p_i + \varepsilon_i^\uparrow$  for each index  $i$ . Note that  $\mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p})$  which implies  $\mathbf{p}' \in A^+$ .  $\square$

Theorem 4.48 can be applied to answer a multi-objective query consisting of bounded until objectives. This is detailed in the next section where we also extend the theorem to unbounded until and expected reachability reward objectives.

## 4.4 Combinations of Different Types of Objectives

So far, we discussed how to analyze (combinations of) unbounded until and expected reachability reward objectives for MAs by employing the underlying MDP (Section 4.1 and Section 4.2). Further, lists of bounded until objectives have been treated in Section 4.3 by considering a digitization approach. We now focus on combinations of all three objective types. To this end, consider an MA  $\mathcal{M} = (S, \text{Act}, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_\ell\})$  and a list of objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . Each objective  $\mathbb{O}_i$  can be either

- an unbounded until objective  $\mathbb{P}(H_i \mathcal{U} G_i)$ ,
- an expected reachability reward objective  $\mathbb{E}(\#j_i, G_i)$ , or
- a bounded until objective  $\mathbb{P}(H_i \mathcal{U}^{l_i} G_i)$ .

If  $\circledast$  does not contain a bounded until objective, Theorem 4.9 on page 45 can be applied to conduct the multi-objective analysis on the underlying MDP of  $\mathcal{M}$ . Otherwise, we employ the digitization  $\mathcal{M}_\delta$  of  $\mathcal{M}$  w.r.t. some digitization constant  $\delta \in \mathbb{R}_{>0}$  as presented in the previous section. In the latter case, unbounded until and expected reachability reward objectives need to be checked on  $\mathcal{M}_\delta$  (instead of  $\mathcal{M}_\mathcal{D}$ ).

Recall the definitions of the underlying MDP  $\mathcal{M}_\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1^\mathcal{D}, \dots, \rho_\ell^\mathcal{D}\})$  and the digitization  $\mathcal{M}_\delta = (S, Act, \mathbf{P}_\delta, s_0, \{\rho_1^\delta, \dots, \rho_\ell^\delta\})$  as given in Definition 2.15 on page 17 and Definition 4.16 on page 54, respectively. We discuss the similarities between the two MDPs. First note that we have  $\mathbf{P}(s, \alpha, s') = \mathbf{P}_\delta(s, \alpha, s')$  and  $\rho_i^\mathcal{D}(s, \alpha) = \rho_i^\delta(s, \alpha)$  for each  $s \in \text{PS}$ ,  $\alpha \in Act$ ,  $s' \in S$ , and  $i \in \{1, \dots, \ell\}$ . Put differently,  $\mathcal{M}_\mathcal{D}$  and  $\mathcal{M}_\delta$  coincide at the probabilistic states of  $\mathcal{M}$ . For Markovian states, the introduced self-loops in  $\mathcal{M}_\delta$  can be eliminated by rescaling the probabilities and rewards for the remaining outgoing transitions accordingly. Then, the two MDPs also coincide at Markovian states. We claim that  $\mathcal{M}_\delta$  and  $\mathcal{M}_\mathcal{D}$  yield the same unbounded until probabilities and expected reachability rewards.

**Example 4.49**

Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho\})$  be the MA depicted in Figure 4.13(a). We set  $\lambda = \mathbb{E}(s_0) = \lambda_1 + \lambda_2$ . The underlying MDP  $\mathcal{M}_\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho^\mathcal{D}\})$  and the digitization  $\mathcal{M}_\delta = (S, Act, \mathbf{P}_\delta, s_0, \{\rho^\delta\})$  of  $\mathcal{M}$  are illustrated in Figure 4.13(b) and Figure 4.13(c), respectively. We omit the (unique) scheduler  $\sigma$  in notations like  $\text{Pr}_\sigma^{\mathcal{M}_\delta}$  or  $\text{eR}_\sigma^{\mathcal{M}_\mathcal{D}}$ .

We compute the reachability probability  $\text{Pr}^{\mathcal{M}_\delta}(\diamond\{s_1\})$  which is the probability of the paths  $\{(s_0 \xrightarrow{\perp} s_0)^m s_0 \xrightarrow{\perp} s_1 \mid m \geq 0\}$ , yielding

$$\begin{aligned} \text{Pr}^{\mathcal{M}_\delta}(\diamond s_1) &= \sum_{m=0}^{\infty} (e^{-\lambda\delta})^m \cdot \frac{\lambda_1}{\lambda} \cdot (1 - e^{-\lambda\delta}) \\ &= \frac{1}{(1 - e^{-\lambda\delta})} \cdot \frac{\lambda_1}{\lambda} \cdot (1 - e^{-\lambda\delta}) = \frac{\lambda_1}{\lambda} = \text{Pr}^{\mathcal{M}_\mathcal{D}}(\diamond s_1) \stackrel{\text{Prop. 4.8}}{=} \text{Pr}^{\mathcal{M}}(\diamond s_1). \end{aligned}$$

Here, we considered the closed form of the geometric series  $\sum_{m=0}^{\infty} (e^{-\lambda\delta})^m = \frac{1}{(1 - e^{-\lambda\delta})}$  which holds since  $|e^{-\lambda\delta}| < 1$ .

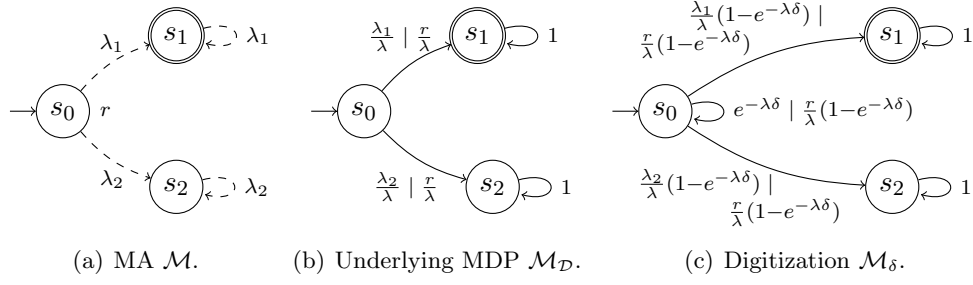


Figure 4.13: MA  $\mathcal{M}$  with underlying MDP  $\mathcal{M}_{\mathcal{D}}$  and digitization  $\mathcal{M}_{\delta}$  (cf. Example 4.49).

The expected reachability reward  $eR^{\mathcal{M}_{\delta}}(\rho^{\delta}, G)$  for  $G = \{s_1, s_2\}$  is given by

$$\begin{aligned}
 eR^{\mathcal{M}_{\delta}}(\rho^{\delta}, G) &= \sum_{m=0}^{\infty} \underbrace{(m+1) \cdot \frac{r}{\lambda} \cdot (1-e^{-\lambda\delta})}_{\text{reward}} \cdot \underbrace{(e^{-\lambda\delta})^m \cdot \frac{\lambda_1 + \lambda_2}{\lambda} \cdot (1-e^{-\lambda\delta})}_{\text{probability}} \\
 &= \frac{r}{\lambda} \cdot (1-e^{-\lambda\delta})^2 \cdot \sum_{m=0}^{\infty} (m+1) \cdot (e^{-\lambda\delta})^m \\
 &= \frac{r}{\lambda} \cdot \frac{(1-e^{-\lambda\delta})^2}{e^{-\lambda\delta}} \cdot \sum_{m=1}^{\infty} m \cdot (e^{-\lambda\delta})^m \\
 &= \frac{r}{\lambda} \cdot \frac{(1-e^{-\lambda\delta})^2 \cdot e^{-\lambda\delta}}{e^{-\lambda\delta} \cdot (1-e^{-\lambda\delta})^2} = \frac{r}{\lambda} = eR^{\mathcal{M}_{\mathcal{D}}}(\rho^{\mathcal{D}}, G) \stackrel{\text{Prop. 4.13}}{=} eR^{\mathcal{M}}(\rho, G). \quad \blacksquare
 \end{aligned}$$

Proposition 4.8 on page 44 and Proposition 4.13 on page 51 can be adapted to the digitization approach.

#### Proposition 4.50

Let  $\mathcal{M} = (S, Act, \rightarrow, \dashrightarrow, s_0, \{\rho_1, \dots, \rho_{\ell}\})$  be an MA with scheduler  $\sigma \in \text{GM}$  and digitization  $\mathcal{M}_{\delta} = (S, Act, \mathbf{P}_{\delta}, s_0, \{\rho_1^{\delta}, \dots, \rho_{\ell}^{\delta}\})$ . For each  $H, G \subseteq S$  and  $i \in \{1, \dots, \ell\}$  it holds that

$$\Pr_{\sigma}^{\mathcal{M}}(HU G) = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(HU G) \quad \text{and} \quad eR_{\sigma}^{\mathcal{M}}(\rho_i, G) = eR_{\text{di}(\sigma)}^{\mathcal{M}_{\delta}}(\rho_i^{\delta}, G). \quad \blacksquare$$

We omit the proof of the proposition as it can be conducted in a similar way as the proofs of Proposition 4.8 and Proposition 4.13. The next step is to extend Theorem 4.48 on page 77 to arbitrary types of objectives. To this end, let  $\mathbb{O}_i$  be either an unbounded until or an expected reachability reward objective. We set

$$\varepsilon_i^{\downarrow} = \varepsilon_i^{\uparrow} = 0.$$

Assume  $\mathbb{O}_i = \mathbb{P}(HUG)$  and let  $p_i = \Pr_{\text{di}(\sigma)}^{\mathcal{M}_\delta}(HUG)$ . Proposition 4.50 yields

$$\Pr_\sigma^{\mathcal{M}}(HUG) \in [p_i - \varepsilon_i^\downarrow, p_i + \varepsilon_i^\uparrow].$$

A similar observation can be made for expected reachability reward objectives. Theorem 4.48 can be generalized to arbitrary types of objectives by following our explanations in Section 4.3.5.

### Theorem 4.51

Let  $\mathcal{M}$  be an MA with digitization  $\mathcal{M}_\delta$ . Furthermore, let  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  be a list of unbounded until, expected reachability reward, and bounded until objectives with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . For the sets

$$\begin{aligned} A^- &= \{\mathbf{p}' \in \mathbb{R}^d \mid \forall \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ implies } \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\} \text{ and} \\ A^+ &= \{\mathbf{p}' \in \mathbb{R}^d \mid \exists \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ and } \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\} \end{aligned}$$

it holds that

$$A^- \subseteq \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})\} \subseteq A^+ . \quad \blacksquare$$

The proof is similar to the proof of Theorem 4.48 presented on page 78. In the following, we discuss how multi-objective model checking of MAs can be performed by applying Theorem 4.51. For  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  and  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$  we assume that the set of achievable points of  $\mathcal{M}_\delta$  given by  $A = \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\}$  has already been computed<sup>4</sup>. Let  $A^-$  and  $A^+$  be as in the theorem above.

**Achievability queries.** Consider the query  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})$  for some  $\mathbf{p} \in \mathbb{R}^d$ . If  $\mathbf{p} \in A^-$ , the answer for the query is *true*. On the other hand, if  $\mathbf{p} \notin A^+$ , the answer is *false*. If none of the cases above holds (i.e.,  $\mathbf{p} \in A^+ \setminus A^-$ ), the approximation can be refined by decreasing the digitization constant and repeating the computation until a conclusive result is obtained. We remark that this procedure might not terminate if the given point  $\mathbf{p}$  is Pareto optimal (for  $\mathcal{M}$ ). The reason is that, in general, the approximation error only approaches zero. A similar issue arises in the single-objective case as presented in, e.g., [GHH<sup>+</sup>14].

**Quantitative queries.** Assume  $\triangleright = (\geq, \triangleright_2, \dots, \triangleright_d)$  and consider the quantitative query  $\text{quantitative}^{\mathcal{M}}(\max \mathbb{O}_1, (\mathbb{O}_2, \dots, \mathbb{O}_d) (\triangleright_2, \dots, \triangleright_d) (p_2, \dots, p_d))$ . Employing the digitization approach, the query can not be answered precisely if it considers bounded until objectives. Instead, a lower bound  $p_1^\ell$  and an upper bound  $p_1^u$  for the optimal value represented by  $\mathbb{O}_1$  is obtained from the sets  $A^-$  and  $A^+$ . It is assumed that there is at least one  $p_1 \in \mathbb{R}$  for which  $\text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright (p_1, \dots, p_d))$  holds (otherwise, the query is *false*). The result for the query is given by

$$p_1^\ell = \sup\{p_1 \in \mathbb{R} \mid (p_1, \dots, p_d) \in A^-\} \quad \text{and} \quad p_1^u = \sup\{p_1 \in \mathbb{R} \mid (p_1, \dots, p_d) \in A^+\}.$$

<sup>4</sup>Chapter 5 details how  $A$  can be obtained algorithmically.

The gap between the obtained lower and upper bound (i.e.,  $p_1^u - p_1^\ell$ ) can be reduced by choosing a smaller digitization constant. Queries with optimization direction min are treated similarly.

**Pareto queries.** Assume optimization directions  $\mathbf{opt} = (\text{opt}_1, \dots, \text{opt}_d)$  such that the threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$  satisfy

$$\triangleright_i = \begin{cases} \geq & \text{if } \text{opt}_i = \max \\ \leq & \text{if } \text{opt}_i = \min. \end{cases}$$

For a query  $\text{pareto}^{\mathcal{M}}(\mathbf{opt} \circledast)$ , the sets  $A^-$  and  $A^+$  can be returned with the interpretation that the Pareto curve (for  $\mathcal{M}$ ) is contained in

$$\{\mathbf{q} \in A^+ \mid \text{there is no } \mathbf{p} \in A^- \text{ with } \mathbf{p} \neq \mathbf{q} \text{ and } \mathbf{p} \triangleright \mathbf{q}\}.$$

Intuitively, this set corresponds to  $A^+ \setminus A^-$  unified with the set of points that lie at the border of  $A^-$ . Again,  $\delta$  can be chosen in advance such that the approximation error is sufficiently small.

**Scheduler synthesis.** Schedulers synthesized during the analysis of  $\mathcal{M}_\delta$  can also be applied to  $\mathcal{M}$ : Consider some point  $\mathbf{p}' = (p_1', \dots, p_d') \in A^-$  for which a scheduler  $\sigma$  should be obtained such that  $\mathcal{M}, \sigma \models \circledast \triangleright \mathbf{p}'$ .

Assume that  $\triangleright = (\geq, \dots, \geq)$  (other cases are treated similarly) and let  $\mathbf{p} = (p_1, \dots, p_d)$  be the point satisfying  $p_i' = p_i - \varepsilon_i^\downarrow$  for each index  $i$ . Following the implications as given in the first part of the proof of Theorem 4.48 presented on page 78, we obtain that  $\mathbf{p}$  is achievable in  $\mathcal{M}_\delta$ . Further, if a scheduler  $\sigma \in \text{TA}^{\mathcal{M}_\delta}$  with  $\mathcal{M}_\delta, \sigma \models \circledast \triangleright \mathbf{p}$  has been synthesized, we can also apply  $\sigma$  to  $\mathcal{M}$  which yields  $\mathcal{M}, \sigma \models \circledast \triangleright \mathbf{p}'$ .

## Chapter 5

# Approximation of the Set of Achievable Points

This chapter presents the approach of [FKP12] which we call the *Pareto curve approximation algorithm* for multi-objective model checking of MDPs. We extend the original work as follows:

1. ds-bounded until objectives (cf. Definition 4.46 on page 76) are incorporated to be able to check time-bounded until objectives of MAs as presented in Section 4.3. This generalizes the results of [FKP12] for step-bounded reachability objectives.
2. Expected reachability reward objectives are considered, while [FKP12] is restricted to expected total reward objectives.
3. [FKP12] imposes the requirement that the given expected reward objectives are either all maximizing or all minimizing. This restriction is rather unfavorable as we usually want to minimize expected costs and maximize expected rewards. We avoid such a requirement by performing additional preprocessing steps based on the elimination of *end components*.

Multi-objective model checking of MAs is enabled by employing the procedure to the results of Chapter 4. We emphasize that this can also be accomplished by considering other multi-objective MDP model checking techniques such as the LP-based approach of [FKN<sup>+</sup>11]. However, the authors of [FKP12] claim that “the performance and scalability of [their] approach is significantly better” compared to the LP-based approach and provide practical experiments to confirm this. Furthermore, the algorithm is based on value iteration, a model checking technique known from single-objective MDP analysis [Put94]. This allows us to employ existing notions for ds-bounded until objectives [HH12] that are also based on value iteration. The following example illustrates the rough idea of the procedure.

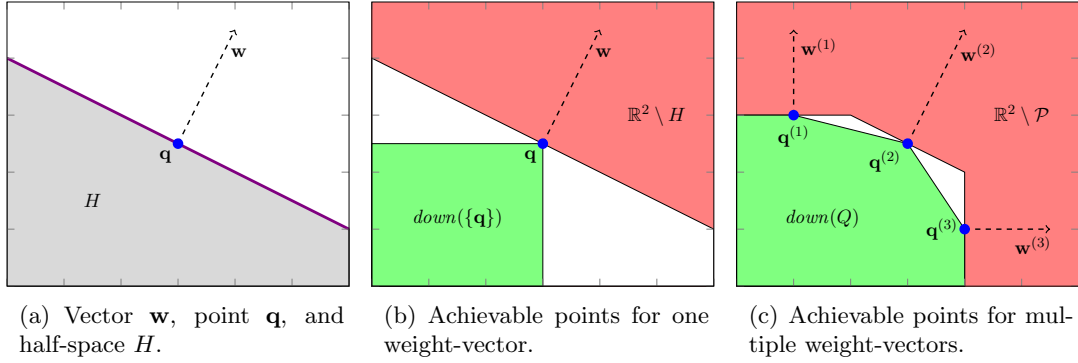


Figure 5.1: Illustration of the Pareto curve approximation algorithm (cf. Example 5.1).

### Example 5.1

Consider an MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  with two objectives  $\mathbb{O} = (\mathbb{O}_1, \mathbb{O}_2)$  and relations  $\triangleright = \{\geq, \leq\}$ . The Pareto curve approximation algorithm iteratively considers *weight-vectors*  $\mathbf{w} = (w_1, w_2) \in \mathbb{R}^2$  with  $\mathbf{w} \neq (0, 0)$  that assign a weight  $w_i \geq 0$  to each objective  $\mathbb{O}_i$ . Optimizing a combination of the (weighted) objectives<sup>1</sup> yields a point  $\mathbf{q} \in \mathbb{R}^2$ . Figure 5.1(a) illustrates the vector  $\mathbf{w}$ , the point  $\mathbf{q}$ , and the so-called *half-space* (cf. Definition 5.2)  $H = \{\mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} \cdot \mathbf{w} \leq \mathbf{q} \cdot \mathbf{w}\}$ . It can be shown that

- $\mathbf{q}$  is achievable and
- all achievable points of  $\mathcal{D}$  are contained in  $H$ .

Since  $\mathbf{q}$  is achievable, any point in the set  $down(\{\mathbf{q}\}) = \{\mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} \leq \mathbf{q}\}$  depicted by the green area of Figure 5.1(b) is achievable as well. On the contrary, there is no achievable point in  $\mathbb{R}^2 \setminus H$ , illustrated by the red area. For the points in the white area, it is still unknown whether they are achievable or not. The set of achievable points of  $\mathcal{D}$  is explored by combining the results for multiple weight-vectors as indicated in Figure 5.1(c). ■

The chapter is structured as follows. Section 5.1 presents preliminary notions for the representation of subsets of  $\mathbb{R}^d$  using geometric objects (in particular, half-spaces and polyhedra). In Section 5.2, we describe the Pareto curve approximation algorithm for the exploration of the set of achievable points of an MDP. Here, we assume that the input of the algorithm is of a certain form. Section 5.3 details the treatment of more general inputs. The analysis of ds-bounded until objectives is discussed separately in Section 5.4.

We emphasize that the notions presented in this chapter are strongly inspired by [FKP12] and the previous works [FKN<sup>+</sup>11, EKVY08].

<sup>1</sup>Details are given in [FKP12] as well as Section 5.2.

## 5.1 Geometric Set Representations

We briefly present the required background for geometric set representations. Further information can be found in, e.g., [Zie95]. Consider a natural number  $d > 0$  and the  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ . The elements of  $\mathbb{R}^d$  are depicted by bold uncapitalized letters (e.g.,  $\mathbf{w}$ ,  $\mathbf{p}$ ,  $\mathbf{q}$ , ...) and we refer to them as either *vectors* or *points* (depending on the geometric interpretation). For each  $\mathbf{p} \in \mathbb{R}^d$  and  $i \in \{1, \dots, d\}$ , we write  $p_i$  to denote the  $i$ -th entry of  $\mathbf{p}$ .  $\mathbf{0} = (0, \dots, 0) \in \mathbb{R}^d$  refers to the zero point (or zero vector). Assume  $\mathbf{w}, \mathbf{p} \in \mathbb{R}^d$ . We denote the *dot product* of  $\mathbf{w}$  and  $\mathbf{p}$  by  $\mathbf{w} \cdot \mathbf{p} = \sum_{i=1}^d w_i \cdot p_i$ . Moreover, for  $\triangleright \in \{<, \leq, >, \geq\}$  we write  $\mathbf{w} \triangleright \mathbf{p}$  iff the relation holds element-wise, i.e.,  $w_i \triangleright p_i$  holds for each  $i$ . We call  $\mathbf{w}$  a *weight-vector* iff  $\mathbf{w} \geq \mathbf{0}$  and  $\mathbf{w} \neq \mathbf{0}$ . Put differently, weight-vectors are the elements of  $\mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$ .

### Definition 5.2 (Half-space)

A ( $d$ -dimensional) *half-space* is a set  $H = H(\mathbf{w}, o) = \{\mathbf{p} \in \mathbb{R}^d \mid \mathbf{w} \cdot \mathbf{p} \leq o\}$  with

- *direction vector*  $\mathbf{w} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$  and
- *offset*  $o \in \mathbb{R}$ . ■

For  $\mathbf{w}, \mathbf{p} \in \mathbb{R}^d$  we also write  $H(\mathbf{w}, \mathbf{p})$  as an abbreviation for  $H(\mathbf{w}, \mathbf{w} \cdot \mathbf{p})$ . Note that the direction vector and the offset of a given half-space is not unique. For instance, we have  $H((2, 4), 8) = H((1, 2), 4)$ .  $\|\cdot\|$  denotes the Euclidean norm of the given vector, i.e.,  $\|\mathbf{w}\| = \sqrt{\mathbf{w} \cdot \mathbf{w}}$ . The *distance* between a half-space  $H$  and a point  $\mathbf{q}$  is given by  $\text{distance}(H, \mathbf{q}) = \min_{\mathbf{p} \in H} (\|\mathbf{q} - \mathbf{p}\|)$ . For  $H = H(\mathbf{w}, o)$  we can show

$$\text{distance}(H, \mathbf{q}) = \begin{cases} 0 & \text{if } \mathbf{q} \in H \\ \frac{|\mathbf{q} \cdot \mathbf{w} - o|}{\|\mathbf{w}\|} & \text{if } \mathbf{q} \notin H. \end{cases}$$

The following example illustrates the geometric interpretation of half-spaces.

### Example 5.3

For  $\mathbf{w} = (1, 2)$  and  $o = 8$ , the half-space  $H = H(\mathbf{w}, o)$  is illustrated by the gray area in Figure 5.2(a). The solid purple line marks the so-called *hyperplane* of  $H$  given by the set  $\{\mathbf{p} \in \mathbb{R}^d \mid \mathbf{w} \cdot \mathbf{p} = o\}$ . Note that the hyperplane is always orthogonal to the direction vector  $\mathbf{w}$  (illustrated by the dashed arrow). Moreover, the point  $\mathbf{p} = (2, 3)$  which lies on the hyperplane satisfies  $\mathbf{w} \cdot \mathbf{p} = 8 = o$ , i.e.,  $H = H(\mathbf{w}, \mathbf{p})$ . The distance of  $H$  to the point  $\mathbf{p}' = (2.5, 4)$  is given by

$$\text{distance}(H, \mathbf{p}') = \frac{|\mathbf{p}' \cdot \mathbf{w} - o|}{\|\mathbf{w}\|} = \frac{10.5 - 8}{\sqrt{5}} = \frac{\sqrt{5}}{2}. \quad \blacksquare$$

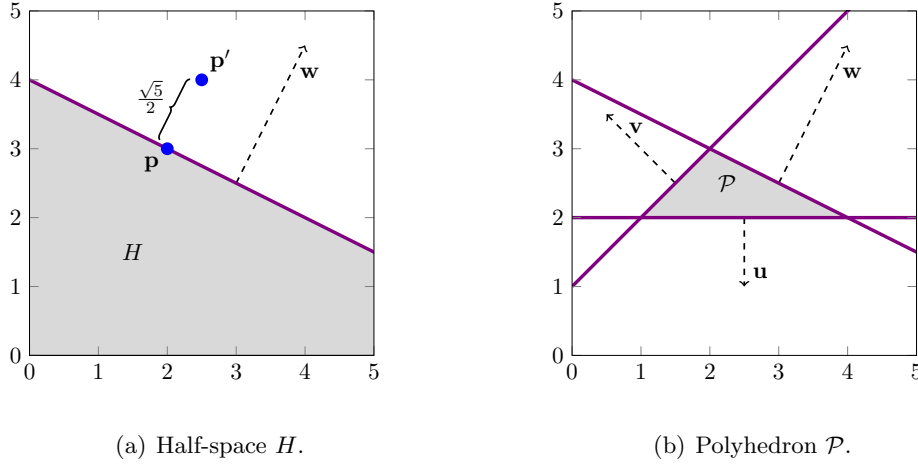


Figure 5.2: Illustration of a half-space and a polyhedron (cf. Example 5.3 and Example 5.5).

**Definition 5.4 (Polyhedron)**

A ( $d$ -dimensional) polyhedron  $\mathcal{P} \subseteq \mathbb{R}^d$  is the intersection of finitely many half-spaces  $H_1, \dots, H_m$ , i.e.,  $\mathcal{P} = \bigcap_{i=1}^m H_i$ . ■

A polyhedron  $\mathcal{P}$  is a *convex* set which means that for each pair of points  $\mathbf{p}, \mathbf{q} \in \mathcal{P}$  and for all  $w \in [0, 1]$  we have  $\mathbf{r} = w \cdot \mathbf{p} + (1 - w) \cdot \mathbf{q} \in \mathcal{P}$ . Intuitively, this means that any point  $\mathbf{r}$  that lies on the line between  $\mathbf{p} \in \mathcal{P}$  and  $\mathbf{q} \in \mathcal{P}$  is also contained in  $\mathcal{P}$ . We write  $\mathcal{H}(\mathcal{P})$  for a minimal set of half-spaces that define the polyhedron  $\mathcal{P}$ , i.e., the assertion  $\mathcal{P} = \bigcap_{H \in \mathcal{H}(\mathcal{P})} H$  holds.

**Example 5.5**

Let  $H_1 = H = H(\mathbf{w}, 8)$  for  $\mathbf{w} = (1, 2)$  be the half-space from Example 5.3. In addition, we consider the half-spaces  $H_2 = H(\mathbf{v}, 1)$  and  $H_3 = H(\mathbf{u}, -2)$ , where  $\mathbf{v} = (-1, 1)$  and  $\mathbf{u} = (0, -1)$ . The gray area in Figure 5.2(b) depicts the polyhedron  $\mathcal{P} = H_1 \cap H_2 \cap H_3$ . ■

**Definition 5.6 (Convex Hull, Downward Closure)**

Let  $Q = \{\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(n)}\} \subseteq \mathbb{R}^d$  be a set of points. The *convex hull* of  $Q$  is given by

$$\text{conv}(Q) = \left\{ \mathbf{q} \in \mathbb{R}^d \mid \mathbf{q} = \sum_{i=1}^n w_i \cdot \mathbf{q}^{(i)} \text{ for some } \mathbf{w} \in \mathbb{R}_{\geq 0}^n \text{ with } \sum_{i=1}^n w_i = 1 \right\}.$$

The *downward closure* of  $Q$  is given by

$$\text{down}(Q) = \{ \mathbf{p} \in \mathbb{R}^d \mid \mathbf{p} \leq \mathbf{q} \text{ for some } \mathbf{q} \in \text{conv}(Q) \}. \quad \blacksquare$$

It can be shown that  $\text{conv}(Q)$  and  $\text{down}(Q)$  are polyhedra. Given the points in  $Q$ , there are efficient algorithms to obtain the sets of half-spaces  $\mathcal{H}(\text{conv}(Q))$  and  $\mathcal{H}(\text{down}(Q))$ , e.g., the Quickhull algorithm [BDH96].

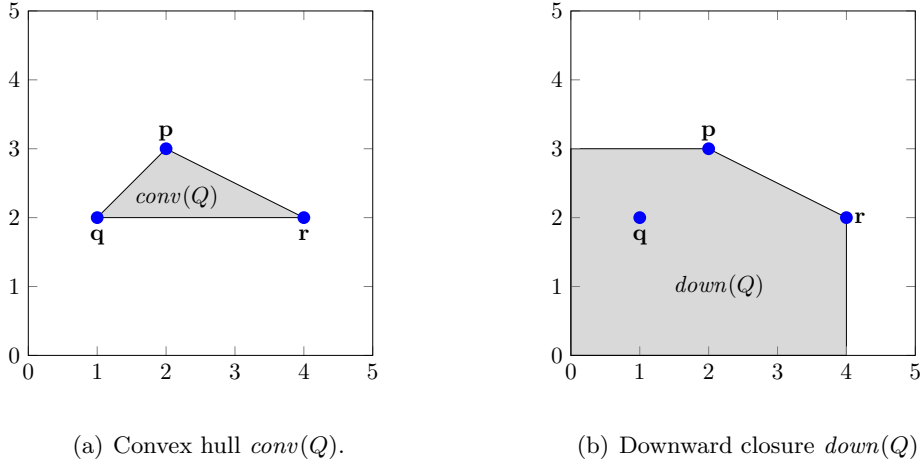


Figure 5.3: Illustration of a convex hull and a downward closure (cf. Example 5.7).

### Example 5.7

Consider the points  $\mathbf{p} = (2, 3)$ ,  $\mathbf{q} = (1, 2)$ , and  $\mathbf{r} = (4, 2)$  as well as the set of points  $Q = \{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$ . The convex hull  $\text{conv}(Q)$  is depicted by the gray area in Figure 5.3(a). Note that this set coincides with the polyhedron  $\mathcal{P}$  from Example 5.5. Hence, we have  $\mathcal{H}(\text{conv}(Q)) = \{H_1, H_2, H_3\}$ .

The downward closure  $\text{down}(Q)$  with  $\mathcal{H}(\text{down}(Q)) = \{H_1, H((1, 0), 4), H((0, 1), 3)\}$  is illustrated in Figure 5.3(b). ■

## 5.2 Exploration of Achievable Points

Consider an MDP  $\mathcal{D} = (S, \text{Act}, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  and a list of objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . The Pareto curve approximation algorithm can be employed to answer achievability, quantitative, and Pareto queries. In this section, we describe the procedure for Pareto queries for which the algorithm iteratively explores the set of achievable points (cf. Definition 3.11 on page 37)

$$A = \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p})\}.$$

The set of Pareto optimal points can then be inferred from  $A$ : A point  $\mathbf{q} \in A$  is Pareto optimal iff there is no other point  $\mathbf{p} \in A \setminus \{\mathbf{q}\}$  for which  $\mathbf{p} \triangleright \mathbf{q}$  holds. Solving achievability or quantitative queries is similar but only requires a partial exploration of  $A$ . More details are given in [FKP12].

The exploration of the set of achievable points is conducted on a model and objectives that satisfy the following assumptions. Inputs where these assumptions do not hold are handled in a preprocessing phase which we describe in Section 5.3.

**Assumption 1:**  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  has exactly  $d$  reward functions and  $\mathbb{O}$  only lists expected total reward objectives such that  $\mathbb{O}_i = \mathbb{E}(\#i)$  for each  $i \in \{1, \dots, d\}$ .

**Assumption 2:** Each scheduler induces finite reward, i.e.,  $\max_{\sigma \in \text{TA}} \text{eR}_\sigma(\rho_i) < \infty$  for all  $i \in \{1, \dots, d\}$ .

**Assumption 3:** All objectives are maximizing and consider non-strict thresholds, i.e.,  $\triangleright = (\geq, \dots, \geq)$ .

The key task of the Pareto curve approximation algorithm is to compute optimal points w.r.t. a weighted combination of the objectives.

**Definition 5.8 (Weighted Rewards, Optimal Scheduler, Optimal Point)**

Let  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  be an MDP and  $\mathbf{w} \in \mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$  be a weight-vector. The *weighted reward function* w.r.t.  $\mathbf{w}$  is given by  $\rho_{\mathbf{w}}: S \times Act \rightarrow \mathbb{R}$  with

$$\rho_{\mathbf{w}}(s, \alpha) = \sum_{i=1}^d w_i \cdot \rho_i(s, \alpha)$$

for each  $s \in S$  and  $\alpha \in Act$ .  $\sigma \in \text{TA}$  is an *optimal scheduler* for  $\mathbf{w}$  iff

$$\text{eR}_\sigma(\rho_{\mathbf{w}}) = \max_{\sigma' \in \text{TA}} (\text{eR}_{\sigma'}(\rho_{\mathbf{w}})).$$

$\mathbf{q} \in \mathbb{R}^d$  is an *optimal point* for  $\mathbf{w}$  iff there is an optimal scheduler  $\sigma$  for  $\mathbf{w}$  such that

$$\mathbf{q} = (\text{eR}_\sigma(\rho_1), \dots, \text{eR}_\sigma(\rho_d)). \quad \blacksquare$$

Intuitively, the considered weight-vector  $\mathbf{w} \in \mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$  assigns weights to the respective objectives. A new reward function  $\rho_{\mathbf{w}}$  is considered that combines  $\rho_1, \dots, \rho_d$  by weighting each  $\rho_i$  with  $w_i$ . An optimal scheduler for  $\mathbf{w}$  induces the maximal expected total reward for  $\rho_{\mathbf{w}}$ . The corresponding optimal point is obtained by computing the values represented by  $\mathbb{O}_1, \dots, \mathbb{O}_d$  under this scheduler. Note that there is always an optimal point for any given weight-vector due to Assumption 2. Off-the-shelf techniques for single objective MDP model checking<sup>2</sup> can be employed to algorithmically obtain such an optimal point for a given vector  $\mathbf{w}$ . An approach based on *value iteration* [Put94] is discussed below (see Algorithm 2).

Algorithm 1 depicts the general procedure of the Pareto curve approximation algorithm. It maintains the set of points  $Q$  and the polyhedron  $\mathcal{P}$  such that the invariant  $\text{down}(Q) \subseteq A \subseteq \mathcal{P}$  holds at any step of the computation. Hence,  $\text{down}(Q)$  represents an *under-approximation* and  $\mathcal{P}$  represents an *over-approximation* of the set of achievable points  $A$ .

<sup>2</sup>e.g., value iteration or scheduler iteration.

---

**Algorithm 1** The Pareto curve approximation algorithm
 

---

**Input:** MDP  $\mathcal{D}$ , objectives  $\mathbb{O}$ , relations  $\triangleright$ , approximation threshold  $\eta \in \mathbb{R}_{\geq 0}$ 
**Output:** Under- and over-approximation of the set of achievable points

$$A = \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p})\}$$

- 1: preprocess  $\mathcal{D}$  and  $\mathbb{O}$  (cf. Section 5.3)
  - 2: assert assumptions 1-3
  - 3:  $Q := \emptyset$ ;  $\mathcal{P} := \mathbb{R}^d$
  - 4: **repeat**
  - 5:   choose weight-vector  $\mathbf{w} \in \mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$
  - 6:   compute optimal point  $\mathbf{q}$  for  $\mathbf{w}$
  - 7:    $Q := Q \cup \{\mathbf{q}\}$ ;  $\mathcal{P} := \mathcal{P} \cap H(\mathbf{w}, \mathbf{q})$
  - 8: **until**  $\sup\{\text{distance}(H, \mathbf{p}) \mid H \in \mathcal{H}(\text{down}(Q)), \mathbf{p} \in \mathcal{P}\} \leq \eta$
  - 9: **return**  $(\text{down}(Q), \mathcal{P})$
- 

The algorithm heuristically chooses a weight-vector  $\mathbf{w} \in \mathbb{R}^d$  at Line 5 and computes an optimal point  $\mathbf{q} \in \mathbb{R}^d$  w.r.t.  $\mathbf{w}$  at Line 6 (details are given below). The under-approximation  $\text{down}(Q)$  is enlarged by adding the point  $\mathbf{q}$  to  $Q$ . Furthermore, the over-approximation  $\mathcal{P}$  is updated by considering the intersection of  $\mathcal{P}$  with the half-space  $H(\mathbf{w}, \mathbf{q})$ . These steps are repeated until the largest distance between a half-space in  $\mathcal{H}(\text{down}(Q))$  and a point in  $\mathcal{P}$  is below the given threshold  $\eta$ . For  $\eta = 0$ , this criterion implies that  $\text{down}(Q) = A = \mathcal{P}$ . To improve the runtime of the procedure, larger thresholds  $\eta > 0$  can be considered. This yields an  $\eta$ -approximation of the set of achievable points, meaning that the maximal distance between a point  $\mathbf{p} \in \mathcal{P}$  and the closest point  $\mathbf{p}' \in \text{down}(Q)$  is at most  $\eta$ .

In the following, we discuss the correctness of the returned approximation. Then, the selection of suitable weight-vectors at Line 5 as well as the termination of the procedure is presented. Finally, we consider the computation of optimal points depicted in Line 6.

**Correctness.** The correctness of the Pareto curve approximation algorithm is a consequence of the following proposition. We refer to [FKP12] for more details.

**Proposition 5.9**

Let  $\mathcal{D}$ ,  $\mathbb{O}$ ,  $\triangleright$ , and  $\eta$  be an input for Algorithm 1 such that assumptions 1-3 are satisfied. Further, let  $A = \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p})\}$  be the set of achievable points. During the execution of Algorithm 1 the invariant  $\text{down}(Q) \subseteq A \subseteq \mathcal{P}$  holds. ■

*Proof.* First consider some point  $\mathbf{p} \in \text{down}(Q)$  and let  $\mathbf{r} \in \text{conv}(Q)$  such that  $\mathbf{p} \leq \mathbf{r}$ . We show that  $\mathbf{p} \in A$  holds by proving that  $\mathbf{r}$  is achievable. Note that each  $\mathbf{q} \in Q$  is optimal for some weight-vector  $\mathbf{w}$  which particularly means that  $\mathbf{q}$  is achievable. It has been shown in [EKVY08, FKN<sup>+</sup>11] that the set of achievable points is convex, implying that  $\mathbf{r} \in \text{conv}(Q)$  has to be achievable as well.

Now let  $\mathbf{p} \in \mathbb{R}^d$  be achievable with scheduler  $\sigma$ . To verify  $\mathbf{p} \in \mathcal{P}$ , we show  $\mathbf{p} \in H$  for each  $H \in \mathcal{H}(\mathcal{P})$ . To this end, note that  $\mathcal{H}(\mathcal{P})$  consists of half-spaces of the form  $H(\mathbf{w}, \mathbf{q})$ , where  $\mathbf{w} \in \mathbb{R}^d$  is a weight-vector and  $\mathbf{q}$  is the corresponding optimal point, induced by an optimal scheduler  $\sigma'$ . It follows that  $\mathbf{p} \in H(\mathbf{w}, \mathbf{q})$  since

$$\mathbf{w} \cdot \mathbf{p} \leq \sum_{i=1}^d w_i \cdot \text{eR}_\sigma(\rho_i) = \text{eR}_\sigma(\rho_{\mathbf{w}}) \stackrel{*}{\leq} \text{eR}_{\sigma'}(\rho_{\mathbf{w}}) = \sum_{i=1}^d w_i \cdot \text{eR}_{\sigma'}(\rho_i) = \mathbf{w} \cdot \mathbf{q},$$

where  $*$  holds since  $\sigma'$  induces the maximum expected total reward for  $\rho_{\mathbf{w}}$ .  $\square$

It follows that the output  $(\text{down}(Q), \mathcal{P})$  of Algorithm 1 corresponds to an under- and an over-approximation of the set of achievable points satisfying the termination criterion

$$\sup\{\text{distance}(H, \mathbf{p}) \mid H \in \mathcal{H}(\text{down}(Q)), \mathbf{p} \in \mathcal{P}\} \leq \eta.$$

**Selection of weight-vectors.** For the correctness proof, we assumed that the vector  $\mathbf{w}$  chosen in Line 5 of Algorithm 1 is some arbitrary weight-vector. To accomplish an actual refinement and to guarantee termination of the procedure,  $\mathbf{w}$  has to be selected more thoughtfully. We follow the suggestion of [FKP12] and adhere to the following strategy:

In the first  $d$  iterations, the vectors  $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(d)}$  are selected, where  $\mathbf{w}^{(j)}$  is given by

$$w_i^{(j)} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

for each  $i, j \in \{1, \dots, d\}$ . Thus, the maximum achievable values are obtained for the single objectives. In subsequent iterations, we select the direction vector of a half-space in  $\mathcal{H}(\text{down}(Q))$  with a maximal distance to any point in  $\mathcal{P}$ . More precisely, for each  $H \in \mathcal{H}(\text{down}(Q))$  we compute the value

$$\max_{\mathbf{p} \in \mathcal{P}} (\text{distance}(H, \mathbf{p}))$$

by solving a linear optimization problem<sup>3</sup>. Let  $H(\mathbf{w}, o) \in \mathcal{H}(\text{down}(Q))$  be a half-space for which this value is maximal. The above heuristic chooses the direction vector  $\mathbf{w}$  of  $H(\mathbf{w}, o)$  as the next weight-vector.

### Example 5.10

We illustrate an example execution of Algorithm 1 in Figure 5.4. Consider the MDP  $\mathcal{D} = (S, \text{Act}, \mathbf{P}, s_0, \{\rho_1, \rho_2\})$  from Figure 5.4(a) and objectives  $\mathbb{O} = (\mathbb{E}(\#1), \mathbb{E}(\#2))$ . Further, let  $\eta = 0.23$  be the threshold for the accuracy of the approximation.

<sup>3</sup>The problem considers  $d$  variables and  $|\mathcal{H}(\mathcal{P})|$  constraints and can be solved using an LP-solver.

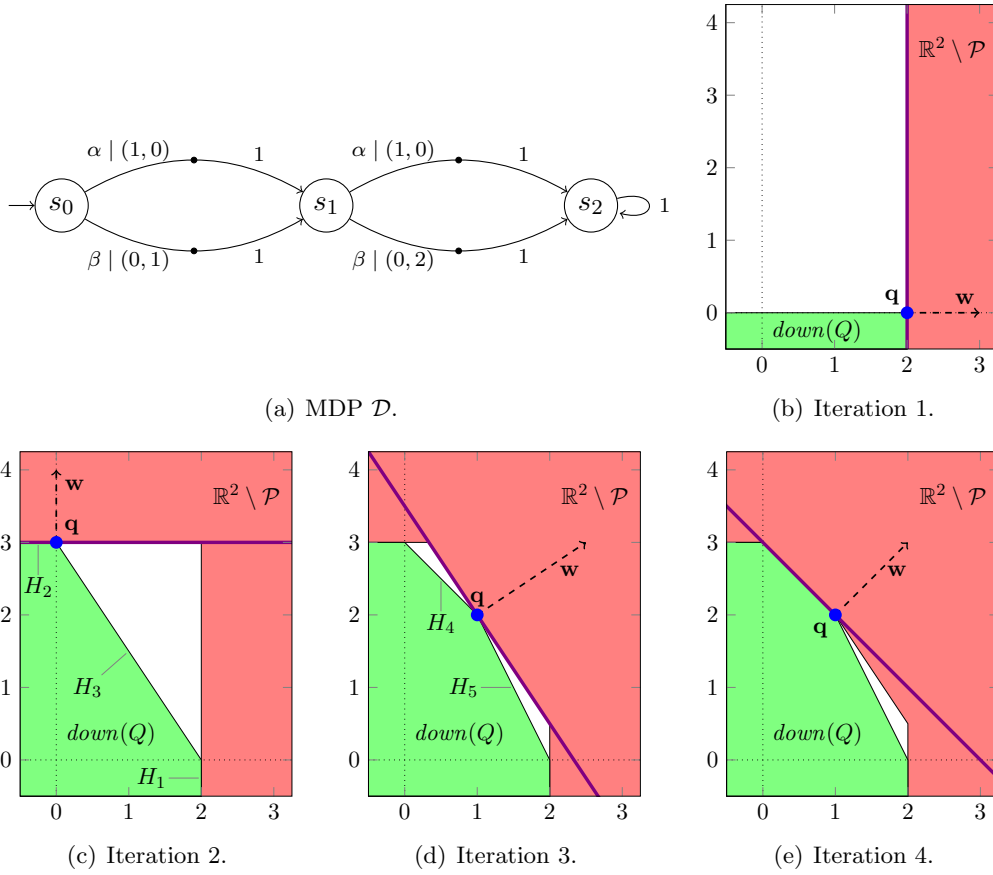


Figure 5.4: Example execution of Algorithm 1 (cf. Example 5.10).

In the first iteration, we choose the weight-vector  $\mathbf{w} = (1, 0)$ . For the corresponding weighted reward function it holds that  $\rho_{\mathbf{w}} = \rho_1$ . A scheduler that maximizes the expected total reward for this function is given by  $\sigma$  with  $\sigma(s_0) = \sigma(s_1) = \alpha$ . Note that we employ the notation of stationary deterministic schedulers as functions from states to actions (as discussed in Section 2.4.3). Computing the expected total reward for the two functions  $\rho_1$  and  $\rho_2$  under  $\sigma$  yields the optimal point for  $\mathbf{w}$ , which is given by  $\mathbf{q} = (\text{eR}_{\sigma}(\rho_1), \text{eR}_{\sigma}(\rho_2)) = (2, 0)$ . We insert  $(2, 0)$  into  $Q$  and add the half-space  $H((1, 0), 2)$  to the polyhedron  $\mathcal{P}$ . Figure 5.4(b) depicts the vector  $\mathbf{w}$  and the point  $\mathbf{q}$ . The green area depicts the resulting under-approximation  $\text{down}(Q)$ . The red area stands for the complement of the over-approximation, i.e.,  $\mathbb{R}^2 \setminus \mathcal{P}$ .

For the second iteration, we consider the weight-vector  $\mathbf{w} = (0, 1)$ . Similar to the previous iteration, we obtain the optimal scheduler  $\sigma$  with  $\sigma(s_0) = \sigma(s_1) = \beta$ . The corresponding point is given by  $\mathbf{q} = (0, 3)$ . The updated approximations are illustrated

in Figure 5.4(c). Note that we have

$$\mathcal{H}(\text{down}(Q)) = \left\{ \underbrace{H((1, 0), 2)}_{=H_1}, \underbrace{H((0, 1), 3)}_{=H_2}, \underbrace{H((3/2, 1), 3)}_{=H_3} \right\}.$$

The maximum distance between the half-space  $H_3$  and a point in  $\mathcal{P}$  is given by

$$\text{distance}(H_3, (2, 3)) = 6/\sqrt{13} \approx 1.6641 > \eta.$$

The third iteration considers the direction vector of  $H_3$ , i.e., we set  $\mathbf{w} = (3/2, 1)$ . An optimal scheduler for  $\mathbf{w}$  satisfies  $\sigma(s_0) = \alpha$  and  $\sigma(s_1) = \beta$  as it induces  $\text{eR}_\sigma(\rho_{\mathbf{w}}) = 3/2 \cdot 1 + 1 \cdot 2 = 7/2$ . Applying this scheduler for the individual objectives yields  $\mathbf{q} = (1, 2)$ . Figure 5.4(d) illustrates the current status of the approximation. We have

$$\mathcal{H}(\text{down}(Q)) = \left\{ \underbrace{H((1, 0), 2)}_{=H_1}, \underbrace{H((0, 1), 3)}_{=H_2}, \underbrace{H((1, 1), 3)}_{=H_4}, \underbrace{H((2, 1), 4)}_{=H_5} \right\}.$$

The maximum distance between half-spaces  $H_4$  and  $H_5$  and a point in  $\mathcal{P}$  is given by

$$\begin{aligned} \text{distance}(H_4, (1/3, 3)) &= 1/\sqrt{18} \approx 0.236 > \eta \text{ and} \\ \text{distance}(H_5, (2, 0.5)) &= 1/\sqrt{20} \approx 0.224 \leq \eta. \end{aligned}$$

Hence, we choose the direction vector of  $H_4$  for the fourth iteration. We consider the optimal scheduler for  $\mathbf{w} = (1, 1)$  given by  $\sigma(s_0) = \beta$  and  $\sigma(s_1) = \alpha$ . Notice that  $\sigma$  coincides with the scheduler of the previous iteration, inducing the same point  $\mathbf{q} = (1, 2)$ . As seen in Figure 5.4(e), the set  $\text{down}(Q)$  does not change in this iteration. However, the over-approximation  $\mathcal{P}$  is refined, yielding  $\max_{\mathbf{p} \in \mathcal{P}} (\text{distance}(H_4, \mathbf{p})) = 0$ . The algorithm terminates since

$$\sup\{\text{distance}(H, \mathbf{p}) \mid H \in \mathcal{H}(\text{down}(Q)), \mathbf{p} \in \mathcal{P}\} = \text{distance}(H_5, (2, 0.5)) \approx 0.224 \leq \eta. \quad \blacksquare$$

**Termination.** Selecting weight-vectors according to the strategy above yields a point  $\mathbf{q}$  that lies on a unique face of the set of achievable values. Intuitively, a face of a  $d$ -dimensional polyhedron is some  $k$ -dimensional object (for  $k \leq d$ ) that lies on the border of the polyhedron. As shown in [EKVY08], the number of faces is (at most) exponential in the size of the MDP and the dimension  $d$ . It follows that after sufficiently many iterations we have  $\text{down}(Q) = A = \mathcal{P}$ . Hence, Algorithm 1 terminates for each threshold  $\eta \in \mathbb{R}_{\geq 0}$ .

The experimental results in [FKP12] and in Chapter 6 of this work indicate that the number of required iterations on practical examples is comparably small. This particularly holds when thresholds  $\eta > 0$  are considered<sup>4</sup>.

<sup>4</sup>In Chapter 6 we consider the cases  $\eta = 0.01$  and  $\eta = 0.001$ .

---

**Algorithm 2** Value iteration-based computation of optimal point
 

---

**Input:** MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$ , weight-vector  $\mathbf{w}$ , threshold  $\varepsilon \in \mathbb{R}_{>0}$ 
**Output:** Optimal point  $\mathbf{q}$  for  $\mathbf{w}$ 

```

    // obtain optimal deterministic scheduler  $\sigma: S \rightarrow Act$ 
1:  $\mathbf{x} := \mathbf{0}$ 
2: repeat
3:    $\mathbf{y} := \mathbf{x}$ 
4:   for all  $s \in S$  do
5:      $x_s := \max_{\alpha \in Act(s)} (\rho_{\mathbf{w}}(s, \alpha) + \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot y_{s'})$ 
6:      $\sigma(s) := \arg \max_{\alpha \in Act(s)} (\rho_{\mathbf{w}}(s, \alpha) + \sum_{s' \in S} \mathbf{P}(s, \alpha, s') \cdot y_{s'})$ 
7:   end for
8: until  $\max\{|x_s - y_s| \mid s \in S\} \leq \varepsilon$ 

    // compute expected total reward under  $\sigma$  for each objective
9: for all  $i \in \{1, \dots, d\}$  do
10:   $\mathbf{x}^{(i)} := \mathbf{0}$ 
11:  repeat
12:     $\mathbf{y} := \mathbf{x}^{(i)}$ 
13:    for all  $s \in S$  do
14:       $x_s^{(i)} := \rho_i(s, \sigma(s)) + \sum_{s' \in S} \mathbf{P}(s, \sigma(s), s') \cdot y_{s'}$ 
15:    end for
16:  until  $\max\{|x_s^{(i)} - y_s| \mid s \in S\} \leq \varepsilon$ 
17: end for
18:  $\mathbf{q} := (x_{s_0}^{(1)}, \dots, x_{s_0}^{(d)})$ 
19: return  $\mathbf{q}$ 

```

---

**Computation of optimal points.** Algorithm 2 depicts a variant of *value iteration* that computes an optimal point  $\mathbf{q}$  for a given weight-vector  $\mathbf{w}$ . There are two phases.

In the first phase (lines 1-8), an optimal scheduler  $\sigma$  is computed. The procedure employs that there is always a deterministic stationary scheduler inducing the maximal expected total reward for a given reward function (in this case  $\rho_{\mathbf{w}}$ ). We consider two vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{|S|}$ , each storing one entry for every state of  $\mathcal{D}$ . Assume that we are in the  $j$ -th iteration of the outer loop. Vector  $\mathbf{y}$  stores the contents of  $\mathbf{x}$  from the previous iteration (Line 3) which corresponds to the maximal expected reward (w.r.t.  $\rho_{\mathbf{w}}$ ) that is collected within  $j - 1$  transitions. Then, we set  $\mathbf{x}$  to the maximal expected reward within  $j$  transitions by combining the reward for taking one transition with the values stored in  $\mathbf{y}$  (Line 5). An action that induces this maximum is stored in  $\sigma(s)$  for every  $s \in S$ . (Line 6). The procedure stops as soon as the maximum difference between two entries of  $\mathbf{x}$  and  $\mathbf{y}$  is below the threshold  $\varepsilon$ .

In the second phase (lines 9-18), the optimal point  $\mathbf{q}$  is computed on the DTMC induced by the scheduler  $\sigma$  (cf. Definition 2.14 on page 17). The procedure is similar

to the first phase besides the separate computation of the values for each objective and the choice of the actions according to  $\sigma$ . The entries of the resulting point  $\mathbf{q}$  are set to the computed values for the initial state  $s_0$  (Line 18).

**Remark 5.11**

Value iteration is a widely used technique for the verification of MDPs. However, the approach as presented above is not sound. The issue relates to the termination criterion  $\max\{|x_s - y_s| \mid s \in S\} \leq \varepsilon$  (as in lines 8 and 16 of Algorithm 2) which gives no guarantee for the convergence of the computed result. Details are given in [HM14], where the authors also present *interval iteration*, a variant of value iteration that intuitively approaches the solution from below and above, yielding arbitrary tight lower and upper bounds. An adaption of interval iteration to our scenario is possible, although further care has to be taken when no a priori bound to the expected rewards is known. Alternatively, *scheduler iteration* can be employed where such problems do not arise. ■

### 5.3 Treatment of a Broader Class of Inputs

The Pareto curve approximation algorithm requires a certain form of input as described by assumptions 1-3 on page 88. In this section, aim to translate other inputs to this form. We consider an MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  and a list of objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  with threshold relations  $\triangleright = (\triangleright_1, \dots, \triangleright_d)$ . We assume that each objective  $\mathbb{O}_i$  is either an unbounded until or an expected reachability reward objective. ds-bounded until objectives are treated separately in Section 5.4.

#### 5.3.1 Transformation to Expected Total Reward Objectives

The objectives  $\mathbb{O}_1, \dots, \mathbb{O}_d$  are transformed to equivalent expected total reward objectives in order to satisfy Assumption 1.

**Assumption 1:**  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  has exactly  $d$  reward functions and  $\mathbb{O}$  only lists expected total reward objectives such that  $\mathbb{O}_i = \mathbb{E}(\#i)$  for each  $i \in \{1, \dots, d\}$ .

We achieve this in two steps. First, the occurring unbounded until objectives are transformed to expected *reachability* reward objectives by adding new reward functions to the model. Then, we modify the MDP such that the reachability reward objectives can be replaced with *total* reward objectives. In both steps, each objective  $\mathbb{O}_i$  is considered individually and we construct a new MDP  $\mathcal{D}'$  from  $\mathcal{D}$  that allows us to replace  $\mathbb{O}_i$  with the corresponding expected reward objective  $\mathbb{O}'_i$ . The construction

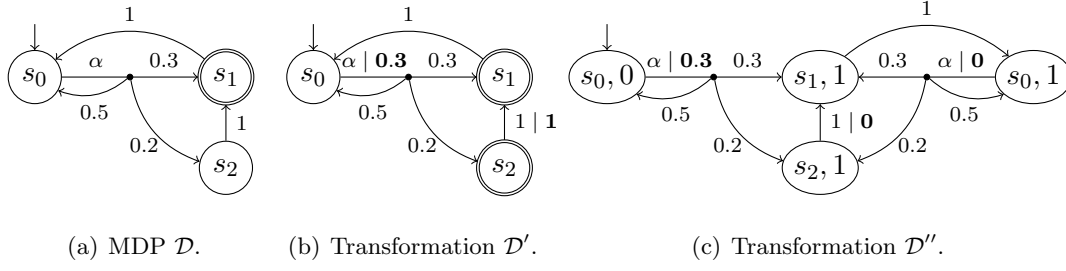


Figure 5.5: Transformation for objective  $\mathbb{P}(\{s_0\} \mathcal{U} \{s_1\})$  to expected total reward objective (cf. Example 5.12 and Example 5.15).

preserves the values represented by other objectives. The objectives are translated successively by repeating the construction for each  $i \in \{1, \dots, d\}$ .

**Unbounded until to expected reachability reward objectives.** Assume the objective  $\mathbb{O}_i = \mathbb{P}(H \mathcal{U} G)$  for  $H, G \subseteq S$ . We create a new reward function  $\rho_{\ell+1}$  that awards reward 1 for entering a state in  $G$ . This is accomplished by setting

$$\rho_{\ell+1}(s, \alpha) = \mathbf{P}(s, \alpha, G) = \sum_{s' \in G} \mathbf{P}(s, \alpha, s')$$

for each  $s \in S$  and  $\alpha \in Act$ . Let  $\mathcal{D}' = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell, \rho_{\ell+1}\})$  be the MDP we obtain from  $\mathcal{D}$  by adding the reward function  $\rho_{\ell+1}$ . The objective  $\mathbb{P}(H \mathcal{U} G)$  is replaced by  $\mathbb{E}(\#\ell+1, G')$  for  $G' = (S \setminus H) \cup G$ . By considering the goal states  $G'$ , we make sure that reward is only collected for paths that visit  $G$  without leaving  $H$ . Furthermore, no reward is obtained when  $G$  is visited a second time. We have (without proof)

$$\Pr_\sigma^{\mathcal{D}}(H \mathcal{U} G) = eR_\sigma^{\mathcal{D}'}(\rho_{\ell+1}, G').$$

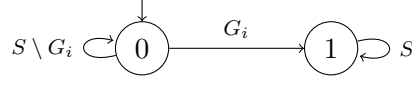
for each scheduler  $\sigma$  for  $\mathcal{D}$  (and  $\mathcal{D}'$ ). Since other objectives are not affected by adding the reward function  $\rho_{\ell+1}$ , it follows that

$$achieve^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p}) \iff achieve^{\mathcal{D}'}(\mathbb{O}' \triangleright \mathbf{p}).$$

for any point  $\mathbf{p} \in \mathbb{R}^d$  and  $\mathbb{O}' = (\mathbb{O}_1, \dots, \mathbb{O}_{i-1}, \mathbb{E}(\#\ell+1, G'), \mathbb{O}_{i+1}, \dots, \mathbb{O}_d)$ .

**Example 5.12**

Consider the MDP  $\mathcal{D}$  from Figure 5.5(a) and the objective  $\mathbb{P}(\{s_0\} \mathcal{U} \{s_1\})$ . To transform the objective to an equivalent expected reachability reward objective, we add a new reward function  $\rho_1$  to  $\mathcal{D}$  which results in the MDP  $\mathcal{D}'$  as shown in Figure 5.5(b). The given objective is replaced by  $\mathbb{E}(\#1, G')$ , where  $G' = S \setminus \{s_0\} \cup \{s_1\} = \{s_1, s_2\}$ . ■

Figure 5.6: Memory structure  $\mathfrak{M}$ .

**Expected reachability reward to expected total reward objectives.** From now on assume the list of expected reachability reward objectives  $\mathbb{O} = (\mathbb{E}(\#1, G_1), \dots, \mathbb{E}(\#d, G_d))$  and the MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  with exactly  $d$  reward functions. Note that it might be necessary to rename, duplicate, or remove the occurring reward functions accordingly. Consider the objective  $\mathbb{O}_i = \mathbb{E}(\#i, G_i)$  for  $G_i \subseteq S$ . For the transformation to an expected *total* reward objective we have to make sure that no more reward is collected as soon as a state in  $G_i$  is visited. This is accomplished by encoding the visit of a state in  $G_i$  in the state space of the new MDP  $\mathcal{D}'$ . We can then set the rewards at states that have been reached via  $G_i$  to zero.

**Definition 5.13 (Memory Structure)**

Let  $\mathcal{D}$  be an MDP with state space  $S$ . A *memory structure* for  $\mathcal{D}$  is a tuple  $\mathfrak{M} = (M, \delta, m_0)$ , where

- $M$  is a finite set of *memory states* with *initial state*  $m_0 \in M$  and
- $\delta: M \times S \rightarrow M$  is a (deterministic) transition function. ■

Memory structures are employed to observe and store certain events regarding the visited states of the corresponding MDP. In our case, we are aimed to store the visit of a state in  $G_i$ . This is accomplished by the memory structure  $\mathfrak{M} = (\{0, 1\}, \delta, 0)$  for  $\mathcal{D}$  with

$$\delta(m, s) = \begin{cases} 1 & \text{if } s \in G_i \\ m & \text{otherwise} \end{cases}$$

for each  $m \in \{0, 1\}$ . Intuitively,  $\mathfrak{M}$  switches from memory state 0 to memory state 1 as soon as a state in  $G_i$  is visited. We illustrate  $\mathfrak{M}$  in Figure 5.6. The next step is to combine  $\mathfrak{M}$  with the MDP  $\mathcal{D}$ .

**Definition 5.14 (Product of an MDP and a Memory Structure)**

Let  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  be an MDP and  $\mathfrak{M} = (M, \delta, m_0)$  be a memory structure for  $\mathcal{D}$ . The *product* of  $\mathcal{D}$  and  $\mathfrak{M}$  is given by the MDP  $\mathcal{D} \times \mathfrak{M} = (S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}, s_0^{\mathfrak{M}}, \{\rho_1^{\mathfrak{M}}, \dots, \rho_d^{\mathfrak{M}}\})$ , where

- $S^{\mathfrak{M}} = S \times M$ ,  $s_0^{\mathfrak{M}} = (s_0, \delta(m_0, s_0))$

and for each  $(s, m), (s', m') \in S^{\mathfrak{M}}$ ,  $\alpha \in Act$ , and  $j \in \{1, \dots, d\}$  we have

- $\mathbf{P}^{\mathfrak{M}}((s, m), \alpha, (s', m')) = \begin{cases} \mathbf{P}(s, \alpha, s') & \text{if } m' = \delta(m, s') \\ 0 & \text{otherwise, and} \end{cases}$

- $\rho_j^{\mathfrak{M}}((s, m), \alpha) = \rho_j(s, \alpha)$ . ■

Intuitively, the transition probabilities and rewards of  $\mathcal{D}$  and  $\mathcal{D} \times \mathfrak{M} = (S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}, s_0^{\mathfrak{M}}, \{\rho_1^{\mathfrak{M}}, \dots, \rho_d^{\mathfrak{M}}\})$  coincide. However, the states of  $\mathcal{D} \times \mathfrak{M}$  contain additional information as stored by the memory structure  $\mathfrak{M}$ . For  $\mathfrak{M}$  as above, this means that the second component of a state  $(s, m) \in S^{\mathfrak{M}}$  satisfies  $m = 1$  iff  $(s, m)$  is only reachable via a state in  $G_i \times \{0, 1\}$ .

To transform the expected reachability reward objective  $\mathbb{E}(\#i, G_i)$  to an expected total reward objective, we consider the MDP  $\mathcal{D}' = (S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}, s_0^{\mathfrak{M}}, \{\rho'_1, \dots, \rho'_d\})$  such that

- $S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}$ , and  $s_0^{\mathfrak{M}}$  are as for  $\mathcal{D} \times \mathfrak{M}$ ,
- $\rho'_j = \rho_j^{\mathfrak{M}}$  for each  $j \neq i$ , and
- $\rho'_i((s, m), \alpha) = (1 - m) \cdot \rho_i^{\mathfrak{M}}((s, m), \alpha)$  for each  $(s, m) \in S^{\mathfrak{M}}$  and  $\alpha \in Act$ .

Let  $\mathbb{O}' = (\mathbb{O}'_1, \dots, \mathbb{O}'_d)$  such that  $\mathbb{O}'_i = \mathbb{E}(\#i)$  and  $\mathbb{O}'_j = \mathbb{E}(\#j, (G_j \times \{0, 1\}))$  for each  $j \neq i$ . We can show that

$$achieve^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p}) \iff achieve^{\mathcal{D}'}(\mathbb{O}' \triangleright \mathbf{p})$$

holds for any point  $\mathbf{p} \in \mathbb{R}^d$ .

### Example 5.15

We continue the transformation of the unbounded until objective from Example 5.12, i.e., consider  $\mathcal{D}'$  as in Figure 5.5(b) as well as the expected reachability reward objective  $\mathbb{E}(\#1, G')$  for  $G' = \{s_1, s_2\}$ . The construction above yields the MDP  $\mathcal{D}''$  whose reachable fragment is shown in Figure 5.5(c). Note that no transition with non-zero reward can be taken as soon as a state in  $G' \times \{0, 1\}$  has been reached, allowing us to consider the expected total reward objective  $\mathbb{E}(\#1)$  for  $\mathcal{D}''$ . ■

Assumption 1 holds after applying the two constructions above for each objective  $\mathbb{O}_i$ . Notice that the number of states of the considered MDP potentially increases by a factor of up to  $2^d$ . From now on, we assume the MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  as well as the list of objectives  $\mathbb{O} = (\mathbb{E}(\#1), \dots, \mathbb{E}(\#d))$ .

### 5.3.2 Infinite Rewards

Recall Assumption 2 which forbids that infinite reward can be collected with a positive probability.

**Assumption 2:** Each scheduler induces finite reward, i.e.,  $\max_{\sigma \in \text{TA}} eR_{\sigma}(\rho_i) < \infty$  for all  $i \in \{1, \dots, d\}$ .

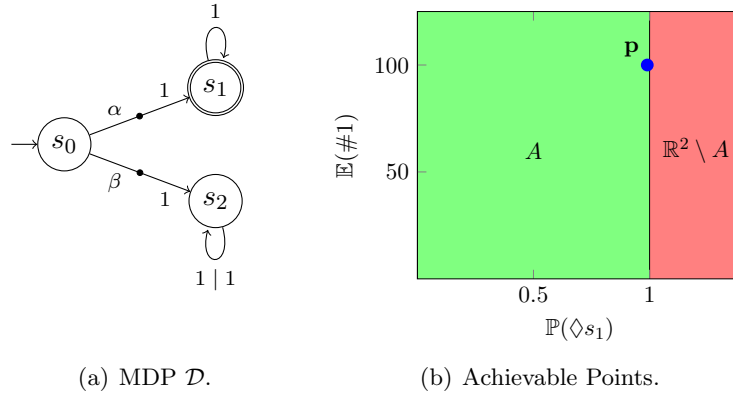


Figure 5.7: MDP with possibly infinite expected total reward and resulting set of achievable points (cf. Example 5.16).

The following example illustrates the inconveniences that arise when the assumption is violated.

**Example 5.16 (from [FKN<sup>+</sup>11])**

Consider the MDP  $\mathcal{D}$  in Figure 5.7(a) and the two objectives  $\mathbb{O}_1 = \mathbb{P}(\diamond s_1)$  and  $\mathbb{O}_2 = \mathbb{E}(\#1)$  with threshold relations  $\triangleright = (\geq, \geq)$ . We assume that  $\mathbb{O}_1$  represents the probability that the modeled system runs safely and that  $\mathbb{O}_2$  represents the expected number of served customers. For any scheduler  $\sigma$  that chooses action  $\beta$  at state  $s_0$  with a non-zero probability, we have that  $eR_\sigma(\rho_1) = \infty$ . The set of achievable points  $A$  is depicted by the green area in Figure 5.7(b). For instance, the point  $\mathbf{p} = (0.99, 100)$  is achievable with scheduler  $\sigma$  such that  $\sigma(s_0, \alpha) = 0.99$ . Practically speaking, the system is able to serve an expected number of at least 100 customers while being safe with probability 0.99. This is accomplished by serving no customer at all with probability 0.99. Also notice that there is no optimal scheduler (and thus no optimal point) for, e.g., weight-vector  $\mathbf{w} = (0, 1)$ . An execution of Algorithm 2 would not terminate for  $\mathbf{w}$ . ■

The authors of [FKN<sup>+</sup>11] claim that “this rather unnatural behaviour would lead to misleading verification results, masking possible errors in the model design”. We follow their suggestion and restrict our analysis to models where such situations do not occur. Hence, inputs that violate Assumption 2 are rejected. We check the assumption algorithmically by performing an analysis of the end components of  $\mathcal{D}$ .

**Definition 5.17 (End Component)**

Let  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  be an MDP. For a set of states  $\tilde{S} \subseteq S$  and a set of state-action pairs  $\{(s, \alpha) \mid s \in \tilde{S} \text{ and } \alpha \in Act(s)\}$ , we say that  $(\tilde{S}, E)$  is an *end component* (EC) of  $\mathcal{D}$  iff

1. for all  $(s, \alpha) \in E$  we have that  $\mathbf{P}(s, \alpha, s') > 0$  implies  $s' \in \tilde{S}$  and

2. the graph  $(\tilde{S}, \{(s, s') \mid \exists(s, \alpha) \in E: \mathbf{P}(s, \alpha, s') > 0\})$  is strongly connected. ■

An EC of the form  $(\{s\}, \emptyset)$  for some  $s \in S$  is called *trivial*. Once a non-trivial EC  $(\tilde{S}, E)$  is reached, a scheduler can enforce that the states in  $\tilde{S}$  are not left anymore and that every state in  $\tilde{S}$  is visited infinitely often. An EC  $(\tilde{S}, E)$  is *maximal* if there is no other EC  $(\tilde{S}', E')$  with  $E \subsetneq E'$ . A *bottom* EC  $(\tilde{S}, E)$  is a maximal EC such that there is no scheduler under which the EC can be left, i.e., for all  $s \in \tilde{S}$  and  $\alpha \in Act(s)$  we have  $(s, \alpha) \in E$ . There are algorithms for identifying the maximal ECs of a given MDP in polynomial time [dA97].

Assume an EC  $(\tilde{S}, E)$  such that there is a state-action pair  $(s, \alpha) \in E$  with  $\rho_i(s, \alpha) > 0$  for some reward function  $\rho_i$ . We note that infinite reward (w.r.t.  $\rho_i$ ) can be collected from the states in  $\tilde{S}$  as there is a scheduler under which we perform the action  $\alpha$  at state  $s$  infinitely often. Assumption 2 is checked by verifying that the maximal probability to reach such an EC is zero. Here, it suffices to consider the reward functions  $\rho_i$  for maximizing objectives  $\mathbb{O}_i$  with  $\triangleright_i \in \{>, \geq\}$ . Infinite rewards induced by minimizing objectives are treated below.

### 5.3.3 Transformation to Threshold Relations $(\geq, \dots, \geq)$

An objective  $\mathbb{O}_i = \mathbb{E}(\#i)$  with threshold relation  $\triangleright_i \in \{<, \leq, >\}$  violates Assumption 3.

**Assumption 3:** All objectives are maximizing and consider non-strict thresholds, i.e.,  $\triangleright = (\geq, \dots, \geq)$ .

We support arbitrary threshold relations as follows. For strict relations  $\triangleright_i \in \{<, >\}$ , we conduct the analysis as for non-strict relations and shrink the resulting approximation of the set of achievable points by considering *open* half-spaces<sup>5</sup>. For the case  $\triangleright_i = \leq$ , the rewards are negated. More precisely, we replace  $\rho_i$  with the new reward function  $\rho'_i: S \times Act \rightarrow \mathbb{R}_{\leq 0}$  such that

$$\rho'_i(s, \alpha) = -\rho_i(s, \alpha)$$

for each  $s \in S$  and  $\alpha \in Act$ . Let  $\mathcal{D}'$  be the MDP obtained from  $\mathcal{D}$  by considering  $\rho'_i$  instead of  $\rho_i$ . For each scheduler  $\sigma$  for  $\mathcal{D}$  (and  $\mathcal{D}'$ ) and for each  $p_i \in \mathbb{R}$  we have

$$\text{eR}_\sigma^{\mathcal{D}}(\rho_i) \leq p_i \iff -\text{eR}_\sigma^{\mathcal{D}}(\rho_i) \geq -p_i \iff \text{eR}_\sigma^{\mathcal{D}'}(\rho'_i) \geq -p_i.$$

Let  $\triangleright' = (\triangleright_1, \dots, \triangleright_{i-1}, \geq, \triangleright_{i+1}, \dots, \triangleright_d)$  and  $\mathbf{p}' = (p_1, \dots, p_{i-1}, -p_i, p_{i+1}, \dots, p_d)$  for some point  $\mathbf{p} = (p_1, \dots, p_d) \in \mathbb{R}^d$ . It follows that

$$\text{achieve}^{\mathcal{D}}(\mathbb{O} \triangleright \mathbf{p}) \iff \text{achieve}^{\mathcal{D}'}(\mathbb{O} \triangleright' \mathbf{p}').$$

<sup>5</sup>An open half-space is a set of the form  $\{\mathbf{p} \in \mathbb{R}^d \mid \mathbf{w} \cdot \mathbf{p} < o\}$  for  $\mathbf{w} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$  and  $o \in \mathbb{R}$ .

Assumption 3 is satisfied by repeating this transformation for each objective  $\mathbb{O}_i$  where the threshold relation  $\triangleright_i$  is  $\leq$ . The result is an MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  and objectives  $\mathbb{O} = (\mathbb{E}(\#1), \dots, \mathbb{E}(\#d))$  with threshold relations  $\triangleright = (\geq, \dots, \geq)$ . Each reward function  $\rho_i$  of  $\mathcal{D}$  considers either non-negative or non-positive rewards. Note that we deviate from our definition of MDPs from Section 2.3 by allowing negative rewards. This introduces some technical complications that we discuss below.

**Infinite negative rewards.** Since we assert Assumption 2, there is no EC  $(\tilde{S}, E)$  with  $\rho_i(s, \alpha) > 0$  for any  $(s, \alpha) \in E$ . However,  $\rho_i(s, \alpha) < 0$  for  $(s, \alpha) \in E$  is still possible, potentially inducing an expected total reward of  $-\infty$ . To assert the termination of Algorithm 2, we need to impose another assumption.

**Assumption 4:** There exists a scheduler under which all expected total rewards are greater than  $-\infty$ , i.e.,  $\exists \sigma \in \text{TA} : \forall i \in \{1, \dots, d\} : \text{eR}_\sigma(\rho_i) > -\infty$ .

The set of achievable points is empty whenever Assumption 4 is violated. To check the assumption algorithmically, we determine the ECs  $(\tilde{S}, E)$  of  $\mathcal{D}$  in which no reward is obtained, i.e.,  $\rho_i(s, \alpha) = 0$  for each  $(s, \alpha) \in E$  and  $i \in \{1, \dots, d\}$ . This can be done by identifying the maximal ECs of an auxiliary MDP  $\mathcal{D}_{\text{aux}}$  that corresponds to  $\mathcal{D}$  except that transitions yielding a non-zero reward are removed. Assumption 4 holds if there is a scheduler under which a component where no reward is obtained is reached with probability one. Together with Assumption 2 we have that

$$-\infty < \max_{\sigma \in \text{TA}} \sum_{i=1}^d w_i \cdot \text{eR}_\sigma^{\mathcal{D}}(\rho_i) < \infty$$

for each weight-vector  $\mathbf{w} = (w_1, \dots, w_d) \in \mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$ , implying that there is an optimal scheduler for each  $\mathbf{w}$ .

**Treating positive and negative rewards.** Due to the possible presence of positive and negative rewards, the value iteration-based approach of Algorithm 2 may yield wrong results.

### Example 5.18

Consider the MDP  $\mathcal{D}$  with the weighted reward function  $\rho_{\mathbf{w}}$  depicted in Figure 5.8(a). Note that the maximal expected total reward is obtained for a scheduler  $\sigma_{\text{opt}}$  that always chooses action  $\alpha$ , yielding  $\text{eR}_{\sigma_{\text{opt}}}(\rho_{\mathbf{w}}) = 2 - 1 = 1$ .

We illustrate an execution of the first phase of Algorithm 2 by denoting the vector  $\mathbf{x} = (x_{s_0}, x_{s_1}, x_{s_2})$  and the scheduler choice  $\sigma(s_0)$  at the end of each iteration.

Initialization:	$\mathbf{x} = ($	$0$	$,$	$0$	$,$	$0$	$)$	
Iteration 1:	$\mathbf{x} = ($	$\max(2, 0) = 2$	$,$	$-1$	$,$	$0$	$)$	$\sigma(s_0) = \alpha$
Iteration 2:	$\mathbf{x} = ($	$\max(1, 2) = 2$	$,$	$-1$	$,$	$0$	$)$	$\sigma(s_0) = \beta$

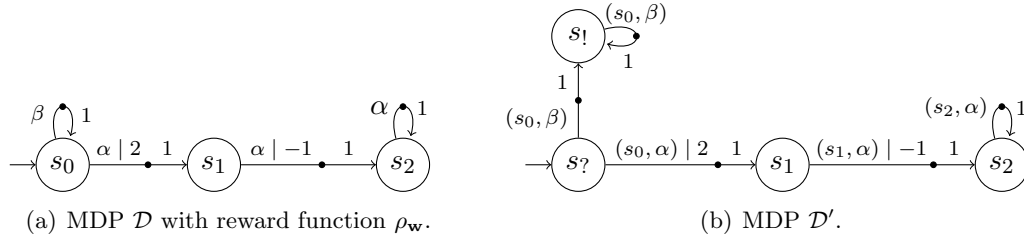


Figure 5.8: MDP where Algorithm 2 yields wrong results and the transformed MDP with eliminated EC (cf. Example 5.18 and Example 5.19).

The procedure converges as the values of  $\mathbf{x}$  do not change in the last iteration. Note that the retrieved scheduler  $\sigma$  with  $\sigma(s_0) = \beta$  does not induce the maximum expected total reward w.r.t.  $\rho_{\mathbf{w}}$ . ■

To avoid this problematic, [FKP12] requires that the initially considered expected reward objectives are either all minimizing or all maximizing. Such a restriction prevents a trade-off analysis between costs (which we want to minimize) and rewards (which we want to maximize). For instance, the quantitative query “What is the maximum expected number of products manufactured with an expected energy consumption of less than ten energy units?” can not be processed with this requirement. We discuss the elimination of certain ECs which allows us to treat both, positive and negative rewards using Algorithm 2.

The problem arises when there are non-bottom ECs in which no reward (neither positive nor negative) is earned. In the example above, this is the case for the EC  $(\{s_0\}, \{(s_0, \beta)\})$ . More formally, we say that a non-trivial EC  $(\tilde{S}, E)$  is *neutral* w.r.t. a reward function  $\rho$  iff  $\rho(s, \alpha) = 0$  for all  $(s, \alpha) \in E$ . It can be shown that the iterations in the first phase of Algorithm 2 converge to an optimal scheduler for the given weight-vector  $\mathbf{w}$  if all the neutral ECs w.r.t.  $\rho_{\mathbf{w}}$  are bottom ECs. Hence, we only eliminate the neutral *non-bottom* ECs before each execution of Algorithm 2. This is accomplished by following an approach similar to the max-reduction presented in [HM14, Definition 10] (see also [dA97, Algorithm 3.3]).

Consider the MDP  $\mathcal{D}$  with weighted reward function  $\rho_{\mathbf{w}}$  and let  $(\tilde{S}, E)$  be a neutral non-bottom EC w.r.t.  $\rho_{\mathbf{w}}$ . We eliminate the EC by replacing the states in  $\tilde{S}$  with two new states:  $s_?$  and  $s_!$ . The outgoing transitions of  $s_?$  model the following cases:

1. The EC is left via a state  $s \in \tilde{S}$  by performing an action  $\alpha$  such that  $(s, \alpha) \notin E$ .
2. We never leave the EC.

The state  $s_!$  represents a bottom EC to which we move when we decide to stay at the EC (Case 2 above). The following example illustrates the idea.

**Example 5.19**

Reconsider the MDP  $\mathcal{D}$  with weighted reward function  $\rho_{\mathbf{w}}$  from Figure 5.8(a). We eliminate the neutral EC  $(\tilde{S}, E) = (\{s_0\}, \{(s_0, \beta)\})$ . The result is given in Figure 5.8(b). At the new state  $s_?$ , a scheduler chooses either the action  $(s_0, \alpha) \notin E$  (representing the case that the EC is left via state  $s_0$  with action  $\alpha$ ) or the action  $(s_0, \beta) \in E$  (representing that we stay at the EC).

We illustrate the execution of the first phase of Algorithm 2 as in Example 5.18 but now consider the transformed MDP  $\mathcal{D}'$ . Let  $\mathbf{x} = (x_{s_?}, x_{s_1}, x_{s_1}, x_{s_2})$ .

Initialization:	$\mathbf{x} = ($	$0$	$, 0 , 0 , 0 )$	
Iteration 1:	$\mathbf{x} = ($	$\max(2, 0) = 2$	$, 0 , -1 , 0 )$	$\sigma(s_?) = (s_0, \alpha)$
Iteration 2:	$\mathbf{x} = ($	$\max(1, 0) = 1$	$, 0 , -1 , 0 )$	$\sigma(s_?) = (s_0, \alpha)$
Iteration 3:	$\mathbf{x} = ($	$\max(1, 0) = 1$	$, 0 , -1 , 0 )$	$\sigma(s_?) = (s_0, \alpha)$

The computation correctly yields a scheduler  $\sigma$  for  $\mathcal{D}'$  that induces a maximal expected total reward. We obtain a corresponding scheduler  $\sigma_{\text{opt}}$  for the original MDP  $\mathcal{D}$  by observing that  $\sigma(s_?) = (s_0, \alpha)$  represents that the EC is left via  $s_0$  by performing  $\alpha$ , yielding  $\sigma_{\text{opt}}(s_0) = \alpha$ . ■

Formally, we eliminate an EC as follows.

**Definition 5.20 (Elimination of End Components)**

Let  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_\ell\})$  be an MDP and  $(\tilde{S}, E)$  be an EC of  $\mathcal{D}$ . The elimination of the EC yields the MDP  $\mathcal{D}' = (S', Act', \mathbf{P}', s_0', \{\rho'_1, \dots, \rho'_d\})$ , where

$$\bullet S' = (S \setminus \tilde{S}) \cup \{s_?, s_1\}, \quad Act' = S \times Act, \quad s_0' = \begin{cases} s_0 & \text{if } s_0 \notin \tilde{S} \\ s_? & \text{otherwise,} \end{cases}$$

and for each  $s, s' \in S'$ ,  $\alpha = (s_{old}, \alpha_{old}) \in Act'$ , and  $i \in \{1, \dots, d\}$  we have that

$$\bullet \rho'_i(s, \alpha) = \begin{cases} \rho_i(s, \alpha_{old}) & \text{if } \alpha = (s, \alpha_{old}) \\ 0 & \text{otherwise,} \end{cases}$$

- and the function  $\mathbf{P}'$  satisfies

- if  $\alpha = (s_{old}, \alpha_{old}) \notin E$  and either  $(s_{old} \notin \tilde{S} \text{ and } s = s_{old})$  or  $(s_{old} \in \tilde{S} \text{ and } s = s_?)$ , then

$$\mathbf{P}'(s, \alpha, s') = \begin{cases} \mathbf{P}(s_{old}, \alpha_{old}, s') & \text{if } s' \in S \setminus \tilde{S} \\ \sum_{\tilde{s} \in \tilde{S}} \mathbf{P}(s_{old}, \alpha_{old}, \tilde{s}) & \text{if } s' = s_?, \end{cases}$$

- if  $\alpha \in E$ , then  $\mathbf{P}'(s_?, \alpha, s_1) = \mathbf{P}'(s_1, \alpha, s_1) = 1$ , and
- otherwise  $\mathbf{P}'(s, \alpha, s') = 0$ . ■

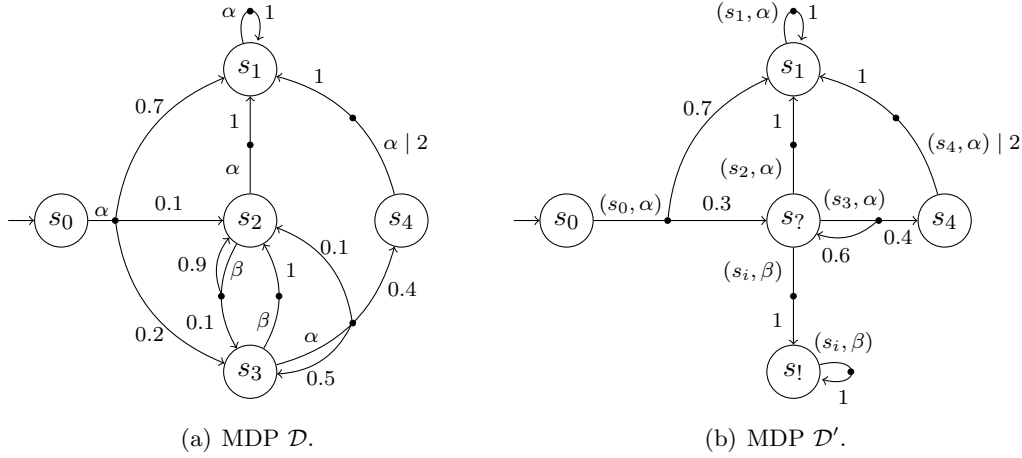


Figure 5.9: Elimination of a neutral EC (cf. Example 5.21 and Example 5.22).

**Example 5.21**

Let  $\mathcal{D}$  be the MDP in Figure 5.9(a). The tuple  $(\{s_2, s_3\}, \{(s_2, \beta), (s_3, \beta)\})$  corresponds to a neutral EC w.r.t. the depicted reward function. By applying the construction above, we obtain the MDP  $\mathcal{D}'$  as shown in Figure 5.9(b). Note that we merged the two actions  $(s_2, \beta)$  and  $(s_3, \beta)$  into  $(s_i, \beta)$  as both actions yield the same transitions. ■

Let  $\mathcal{D}'$  be the MDP obtained from eliminating the neutral EC  $(\tilde{S}, E)$  of  $\mathcal{D}$ . We want to obtain an optimal scheduler for a given weight-vector  $\mathbf{w}$  by analyzing  $\mathcal{D}'$  instead of  $\mathcal{D}$ . To this end, we need to transform a scheduler  $\sigma'$  for  $\mathcal{D}'$  to a scheduler  $\sigma$  for  $\mathcal{D}$ .

We denote by  $\rho_{\mathbf{w}}$  and  $\rho'_{\mathbf{w}}$  the weighted reward functions for  $\mathcal{D}$  and  $\mathcal{D}'$ , respectively. Further, let  $\sigma'$  be a deterministic stationary scheduler for  $\mathcal{D}'$  that induces the maximum expected total reward w.r.t.  $\rho'_{\mathbf{w}}$ . Assume  $\sigma'(s?) = (s_{out}, \alpha_{out})$ . We construct a corresponding scheduler  $\sigma$  for the original MDP  $\mathcal{D}$  satisfying

- $\sigma(s_{out}) = \alpha_{out}$ ,
- for each  $\tilde{s} \in \tilde{S} \setminus \{s_{out}\}$  we have  $(\tilde{s}, \sigma(\tilde{s})) \in E$  such that  $s_{out}$  is eventually reached from  $\tilde{s}$  with probability one, and
- $\sigma'(s) = (s, \alpha)$  implies  $\sigma(s) = \alpha$  for each  $s \notin \tilde{S}$ .

Put differently,  $\sigma$  ensures that we eventually perform action  $\alpha_{out}$  at state  $s_{out}$  whenever we reach the EC. On states outside of the end component,  $\sigma$  coincides with  $\sigma'$ . We can show that  $\sigma$  induces the maximum expected total reward w.r.t.  $\rho_{\mathbf{w}}$ , i.e.,

$$\max_{\hat{\sigma} \in \text{TA}^{\mathcal{D}'}} eR_{\hat{\sigma}}^{\mathcal{D}'}(\rho'_{\mathbf{w}}) = eR_{\sigma'}^{\mathcal{D}'}(\rho'_{\mathbf{w}}) = eR_{\sigma}^{\mathcal{D}}(\rho_{\mathbf{w}}) = \max_{\hat{\sigma} \in \text{TA}^{\mathcal{D}}} eR_{\hat{\sigma}}^{\mathcal{D}}(\rho_{\mathbf{w}}).$$

**Example 5.22**

Let  $\mathcal{D}$  and  $\mathcal{D}'$  be as in Example 5.21. Given the scheduler  $\sigma'$  for  $\mathcal{D}'$  with  $\sigma'(s?) = (s_3, \alpha)$ ,

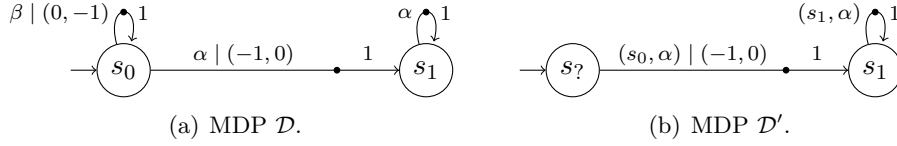


Figure 5.10: Elimination of a neutral EC at which we should not stay (cf. Example 5.23).

a scheduler  $\sigma$  for  $\mathcal{D}$  is obtained by setting  $\sigma(s_3) = \alpha$  and  $\sigma(s_2) = \beta$ . Both schedulers induce the same expected total rewards w.r.t. to the depicted reward function. ■

Schedulers under which we stay at the EC are excluded from the analysis by omitting the state  $s_1$  (and transitions leading to it) when the EC is eliminated. This is necessary if there is a reward function  $\rho_i$  for which staying at the EC always yields an expected total reward of  $-\infty$ . Note that such an EC is still neutral w.r.t. a weighted reward function  $\rho_{\mathbf{w}}$  when the weight-vector  $\mathbf{w}$  satisfies  $w_i = 0$ . We illustrate this case in the next example.

**Example 5.23**

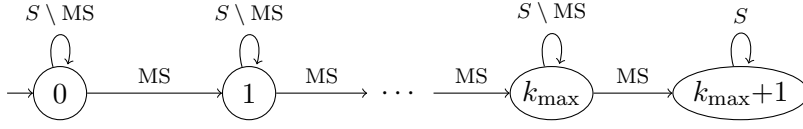
Let  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \rho_2\})$  be the MDP depicted in Figure 5.10(a). We consider the weight-vector  $\mathbf{w} = (1, 0)$ , i.e.,  $\rho_{\mathbf{w}}$  coincides with  $\rho_1$ . The EC  $(\{s_0\}, \{(s_0, \beta)\})$  is neutral w.r.t.  $\rho_{\mathbf{w}}$  and the scheduler  $\sigma$  that uniformly picks  $\beta$  at  $s_0$  is optimal for  $\mathbf{w}$ . However, we observe that  $eR_{\sigma}(\rho_2) = -\infty$ . Hence, there is no point  $\mathbf{p} \in \mathbb{R}^d$  that is achievable with  $\sigma$ . We therefore omit the state  $s_1$  when we eliminate the EC, yielding the MDP  $\mathcal{D}'$  shown in Figure 5.10(b). ■

The construction above is repeated for each neutral non-bottom EC, yielding an MDP  $\mathcal{D}''$  in which all such ECs are eliminated. By considering  $\mathcal{D}''$  in the first phase of Algorithm 2, an optimal scheduler  $\sigma''$  for  $\mathbf{w}$  is computed.  $\sigma''$  is translated to a scheduler  $\sigma$  for  $\mathcal{D}$  such that  $\sigma$  is optimal for  $\mathbf{w}$ . The computation proceeds with the second phase of the algorithm where the original MDP  $\mathcal{D}$  under  $\sigma$  is considered.

## 5.4 Approximation for Bounded Until Objectives

In Chapter 4, we have seen that multi-objective queries for an MA  $\mathcal{M}$  considering bounded until objectives can be approximated by checking ds-bounded until objectives (cf. Definition 4.46) on a digitization  $\mathcal{M}_{\delta}$  of  $\mathcal{M}$ . This section extends the Pareto curve approximation algorithm to process such objectives. To this end, we consider a digitization  $\mathcal{M}_{\delta}$  (recall that  $\mathcal{M}_{\delta}$  is an MDP) and a list of objectives  $\mathbb{O} = (\mathbb{O}_1, \dots, \mathbb{O}_d)$  with threshold relations  $\triangleright$  such that each objective  $\mathbb{O}_i$  is either

- an unbounded until objective  $\mathbb{P}(H_i \mathcal{U} G_i)$ ,

Figure 5.11: Memory structure  $\mathfrak{M}$ .

- an expected reachability reward objective  $\mathbb{E}(\#j_i, G_i)$ , or
- a ds-bounded until objective  $\mathbb{P}(H_i \mathcal{U}_{\text{ds}}^{\leq k_i} G_i)$ .

For simplicity, we initially omit objectives of the form  $\mathbb{P}(H \mathcal{U}_{\text{ds}}^{(j,k]} G)$  or  $\mathbb{P}(H \mathcal{U}_{\text{ds}}^{>j} G)$  in our explanations and discuss them below.

The preprocessing steps for the Pareto curve approximation algorithm are conducted as presented in the previous section, where we treat ds-bounded until objectives such as  $\mathbb{P}(H \mathcal{U}_{\text{ds}}^{\leq k} G)$  similar to the corresponding unbounded objectives  $\mathbb{P}(H \mathcal{U} G)$ . We emphasize that the end component elimination described in Section 5.3.3 is not executed at this point<sup>6</sup>. The result is the MDP  $\mathcal{D} = (S, \text{Act}, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  and the list of expected total reward objectives  $(\mathbb{E}(\#1), \dots, \mathbb{E}(\#d))$ . We further denote by  $\text{MS} \subseteq S$  the set of states of  $\mathcal{D}$  that represent the Markovian states of the initially considered MA. Consider the objective  $\mathbb{O}_i = \mathbb{P}(H_i \mathcal{U}_{\text{ds}}^{\leq k_i} G_i)$ . The corresponding reward function  $\rho_i$  of  $\mathcal{D}$  awards a one-off reward for reaching  $G$  via  $H$ . However, we should not collect this reward when  $G$  is only visited after more than  $k_i$  digitization steps, i.e., after more than  $k_i$  transitions have been taken from the states in MS.

**Transformation to expected total reward objectives.** The idea is to encode the current number of digitization steps within the state space of the model. This allows us to set a reward function  $\rho_i$  to zero whenever this number is too high. In order to apply the approach in practice, we later discuss an optimization that avoids a massive enlargement of the state space. The procedure is based on the idea of [HH12, Algorithm 1].

The MDP  $\mathcal{D}$  is transformed to a new MDP  $\mathcal{D}'$  which keeps track of the number of visited states in MS. Let  $k_{\max}$  be the maximum step bound that occurs in the considered ds-bounded until objectives. Let  $\mathfrak{M} = (\{0, \dots, k_{\max} + 1\}, \delta, 0)$  be the memory structure for  $\mathcal{D}$  with

$$\delta(m, s) = \begin{cases} \min(m + 1, k_{\max} + 1) & \text{if } s \in \text{MS} \\ m & \text{otherwise} \end{cases}$$

for each  $m \in \{0, \dots, k_{\max} + 1\}$ . Intuitively,  $\mathfrak{M}$  counts the number of visited states in MS and stops counting at  $k_{\max} + 1$ . The memory structure  $\mathfrak{M}$  is illustrated in Figure 5.11.

<sup>6</sup>Strictly speaking, the EC elimination is a preprocessing step of Algorithm 2.

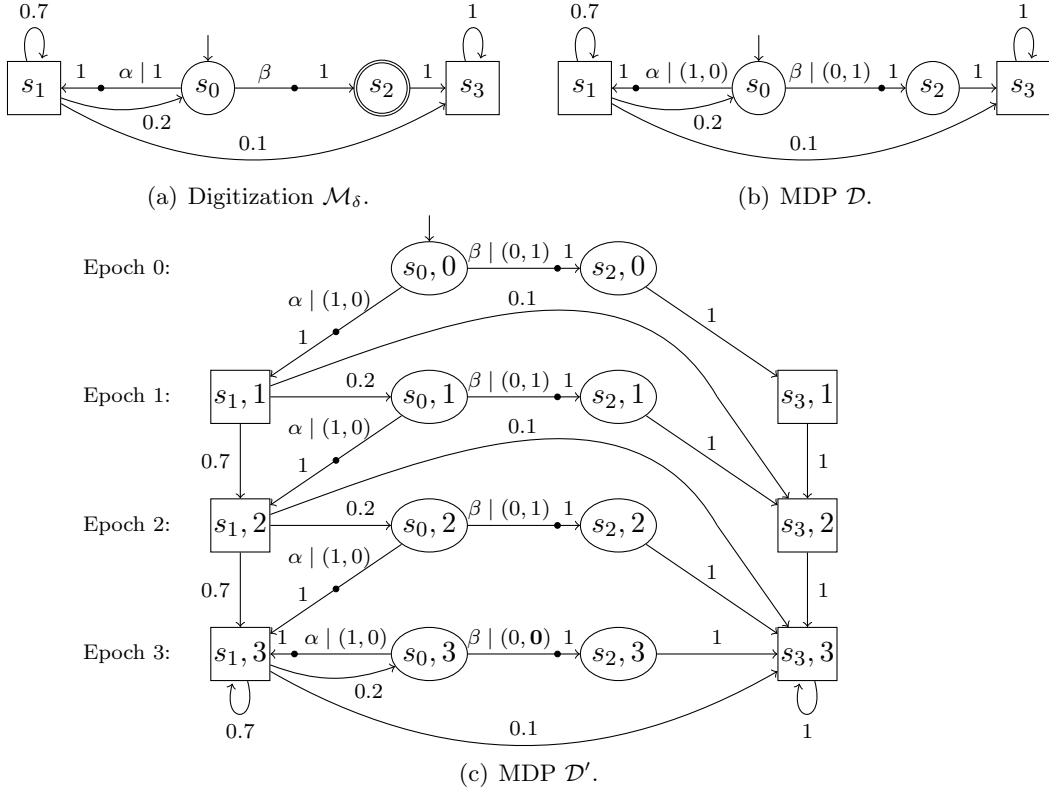


Figure 5.12: Transformation for ds-bounded until objectives (cf. Example 5.24).

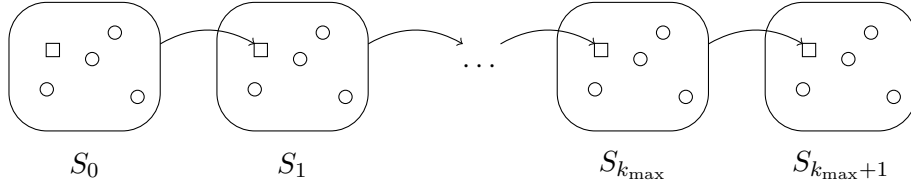
Consider the product  $\mathcal{D} \times \mathfrak{M} = (S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}, s_0^{\mathfrak{M}}, \{\rho_1^{\mathfrak{M}}, \dots, \rho_d^{\mathfrak{M}}\})$ . We define  $\mathcal{D}' = (S^{\mathfrak{M}}, Act, \mathbf{P}^{\mathfrak{M}}, s_0^{\mathfrak{M}}, \{\rho'_1, \dots, \rho'_d\})$  such that

- $S^{\mathfrak{M}}$ ,  $Act$ ,  $\mathbf{P}^{\mathfrak{M}}$ , and  $s_0^{\mathfrak{M}}$  are as for  $\mathcal{D} \times \mathfrak{M}$  and
- for each  $(s, m) \in S^{\mathfrak{M}}$ ,  $\alpha \in Act$ , and  $i \in \{1, \dots, d\}$  we set

$$\rho'_i((s, m), \alpha) = \begin{cases} 0 & \text{if } \mathbb{O}_i = \mathbb{P}(H_i \mathcal{U}^{\leq k_i} G_i) \text{ and } m > k_i \\ \rho_i^{\mathfrak{M}}((s, m), \alpha) & \text{otherwise.} \end{cases}$$

### Example 5.24

We illustrate the construction for the digitization  $\mathcal{M}_\delta$  depicted in Figure 5.12(a) and the objectives  $\mathbb{O} = (\mathbb{E}(\#1, \{s_2\}), \mathbb{P}(S \mathcal{U}_{\text{ds}}^{\leq 2} \{s_2\}))$ . States that correspond to Markovian states of the original MA are depicted with rectangles. Applying the preprocessing steps as discussed in the previous section yields the MDP  $\mathcal{D}$  whose reachable fragment is shown in Figure 5.12(b). The construction above yields the MDP  $\mathcal{D}'$  as illustrated in Figure 5.12(c), where we, again, omit non-reachable states. Note that no reward is collected for the second objective at the states  $(s, m)$  with  $m = k_{\max} + 1 = 3$ . ■

Figure 5.13: Illustration of the structure of  $\mathcal{D}'$ .

Let  $(s, m)$  be a state of  $\mathcal{D}'$ . We refer to the number  $m$  as the *epoch* of the state. It represents the number of states in  $\text{MS}' = \text{MS} \times \{0, \dots, k_{\max} + 1\}$  that have been visited on a path  $\pi \in \text{FPaths}^{\mathcal{D}'}$  with  $\text{last}(\pi) = (s, m)$ . For an objective  $\mathbb{P}(H_i \mathcal{U}_{\text{ds}}^{\leq k_i} G_i)$  it follows that reward is only collected in  $\mathcal{D}'$  if one of the goal states is reached within  $m \leq k_i$  digitization steps. Let  $\mathbb{O}' = (\mathbb{E}(\#1), \dots, \mathbb{E}(\#d))$  and  $\triangleright' = (\geq, \dots, \geq)$ . We can show that

$$\text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p}) \iff \text{achieve}^{\mathcal{D}'}(\mathbb{O}' \triangleright' \mathbf{p}')$$

holds for all points  $\mathbf{p}, \mathbf{p}' \in \mathbb{R}^d$  with

$$p'_i = \begin{cases} p_i & \text{if } \mathbb{O}_i \text{ is maximizing} \\ -p_i & \text{if } \mathbb{O}_i \text{ is minimizing.} \end{cases}$$

**Practical implementation.** Our observations above yield that the set of achievable points of  $\mathcal{M}_\delta$  can be obtained by executing the Pareto curve approximation algorithm on  $\mathcal{D}'$ . In practice, this approach is not feasible as the number  $k_{\max}$  can take high values, leading to huge model sizes. To avoid the problematic, we adapt the computation of optimal points such that the different epochs of  $\mathcal{D}'$  are analyzed individually.

Let  $\mathcal{D}$ ,  $k_{\max}$  and  $\mathcal{D}'$  be as above and let  $S_m = S \times \{m\}$  for  $m \in \{0, \dots, k_{\max} + 1\}$  refer to the set of states of  $\mathcal{D}'$  at epoch  $m$ . To allow more condensed notations, we also define  $S_{k_{\max}+2} = \emptyset$ . Furthermore, assume a weight-vector  $\mathbf{w} \in \mathbb{R}_{\geq 0}^d \setminus \{\mathbf{0}\}$  for which we want to compute an optimal point  $\mathbf{q} \in \mathbb{R}^d$  (w.r.t.  $\mathcal{D}'$ ).

The structure of  $\mathcal{D}'$  is illustrated in Figure 5.13. Let  $m \in \{0, \dots, k_{\max} + 1\}$ . Notice that every transition emerging from a state in  $S_m$  always leads to a state in either  $S_m$  or  $S_{m+1}$ . It follows that the expected rewards obtained in  $S_m$  can be deduced from the sub-model w.r.t.  $S_m$  and the expected rewards at  $S_{m+1}$ . Following this observation, we start the computation of an optimal point with the states at epoch  $k_{\max} + 1$  and successively decrease the considered epoch until the values for the initial state of  $\mathcal{D}'$  are known. To formalize this idea, we introduce the notion of sub-MDPs.

#### Definition 5.25 (Sub-MDP)

For an MDP  $\mathcal{D} = (S, \text{Act}, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$  and a set of states  $S' \subseteq S$ , the sub-MDP of  $\mathcal{D}$  w.r.t.  $S'$  is given by  $\mathcal{D}[S'] = (S', \text{Act}, \mathbf{P}')$ , where  $\mathbf{P}': S' \times \text{Act} \times S' \rightarrow [0, 1]$  is the restriction of  $\mathbf{P}$  to  $S'$ , i.e.,  $\mathbf{P}'(s, \alpha, s') = \mathbf{P}(s, \alpha, s')$  for each  $s, s' \in S'$  and  $\alpha \in \text{Act}$ . ■

---

**Algorithm 3** Computation of optimal point with bounded objectives

---

**Input:** MDP  $\mathcal{D} = (S, Act, \mathbf{P}, s_0, \{\rho_1, \dots, \rho_d\})$ , weight-vector  $\mathbf{w}$

**Output:** Optimal point  $\mathbf{q}$  for  $\mathbf{w}$

---

```

// consider  $\mathcal{D}' = (S \times \{0, \dots, k_{\max} + 1\}, Act, \mathbf{P}', s_0', \{\rho'_1, \dots, \rho'_d\})$  as above
// (not build explicitly)
1:  $\mathbf{x}^{(1)} := \mathbf{0}$ ; ...  $\mathbf{x}^{(d)} := \mathbf{0}$ ;  $k_{\min} := \text{epoch of } s_0'$ 
2: for  $m := k_{\max} + 1$  down to  $k_{\min}$  do
3:   for all  $i \in \{1, \dots, d\}$ ,  $s \in S_m$ , and  $\alpha \in Act$  do
4:      $\tilde{\rho}_i(s, \alpha) := \rho'_i(s, \alpha) + \sum_{s' \in S_{m+1}} \mathbf{P}'(s, \alpha, s') \cdot x_{s'}^{(i)}$ 
5:   end for
6:   build sub-MDP  $\mathcal{D}'[S_m]$ 
7:    $(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(d)}) := \text{VALUEITERATION}(\mathcal{D}'[S_m], \tilde{\rho}_1, \dots, \tilde{\rho}_d, \mathbf{w})$  // Algorithm 2
8: end for
9:  $\mathbf{q} := (x_{s_0'}^{(1)}, \dots, x_{s_0'}^{(d)})$ 
10: return  $\mathbf{q}$ 

```

---

Algorithm 3 depicts the procedure. For each epoch  $m$  (starting with  $k_{\max} + 1$ ), the algorithm computes the optimal values at the states in  $S_m$  w.r.t. the weight-vector  $\mathbf{w}$ . To this end, the reward functions  $\tilde{\rho}_1, \dots, \tilde{\rho}_d$  are defined such that the reward obtained in the current epoch as well as the results from the previous iteration (i.e., from the states in  $S_{m+1}$ ) are incorporated. Then, the algorithm calls the function VALUEITERATION. This is a variant of Algorithm 2, that takes a sub-MDP and  $d$  reward functions as input instead of an MDP. Moreover, VALUEITERATION returns the optimal results  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(d)}$  for all objectives and all states (as computed in Algorithm 2) instead of just the optimal point. Hence, at the end of each iteration, the vector  $\mathbf{x}^{(i)}$  stores the expected total rewards obtained from the states in  $S_m$  w.r.t.  $\rho_i$  and an optimal scheduler  $\sigma$  for  $\mathbf{w}$ . An explicit representation of the MDP  $\mathcal{D}'$  is not necessary as the required information can be computed on-the-fly by following the construction above. Moreover, the considered sub-MDPs  $\mathcal{D}'[S_m]$  are only slight variations of the MDP  $\mathcal{D}$ , enabling an efficient implementation.

**Lower digitization step bounds.** For objectives of the form  $\mathbb{P}(HU_{\text{ds}}^{(j,k)} G)$  or  $\mathbb{P}(HU_{\text{ds}}^{>j} G)$  we need to make sure that reward can still be collected if a state in  $G$  was reached at an epoch  $m \leq j$ . We omit a formal description of the required steps and illustrate the procedure in terms of an example instead.

**Example 5.26**

Let  $\mathcal{M}_\delta$  be the digitization depicted in Figure 5.14(a), where  $s_0$  corresponds to a Markovian state of the original MA. We consider the objective  $\mathbb{P}(SU_{\text{ds}}^{(1,3]} \{s_1\})$ . Conducting the preprocessing as for the corresponding unbounded objective yields the MDP  $\mathcal{D}$  illustrated in Figure 5.14(b). Note that the path  $s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_0 \xrightarrow{\perp} s_1$  of  $\mathcal{M}_\delta$

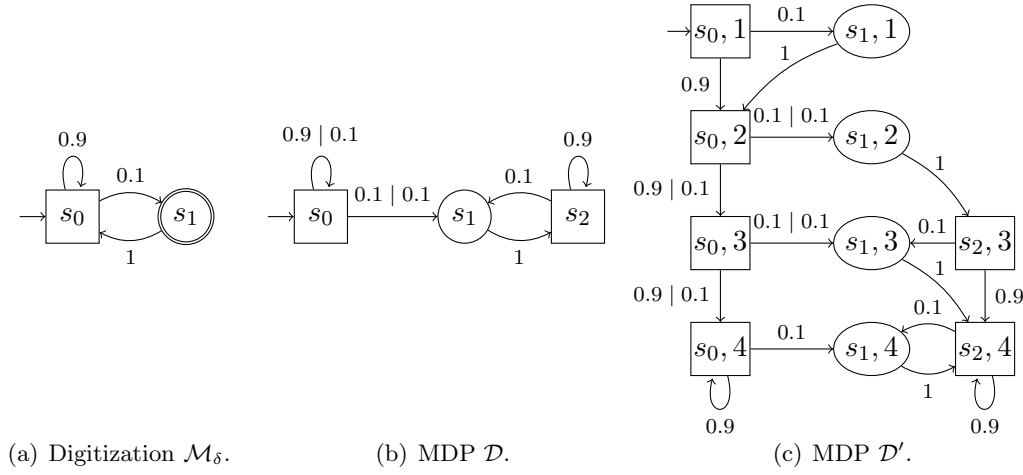


Figure 5.14: Transformation for ds-bounded until objectives with lower digitization step bound (cf. Example 5.26).

reaches  $s_1$  with two digitization steps and should therefore be considered for the probability  $\Pr^{\mathcal{M}_\delta}(S\mathcal{U}_{\text{ds}}^{(1,3)}\{s_1\})$ . However, the corresponding path  $s_0 \xrightarrow{\perp} s_1 \xrightarrow{\alpha} s_2 \xrightarrow{\perp} s_1$  of  $\mathcal{D}$  only yields the reward for the first visit of  $s_1$  which intuitively should not be collected as  $s_1$  is reached too early. For the transformation to an expected total reward objective we therefore consider the MDP  $\mathcal{D}'$  depicted in Figure 5.14(c). Note that the sub-MDPs  $\mathcal{D}'[S_m]$  for epochs  $m > 1$  are similar to  $\mathcal{D}$ , while  $\mathcal{D}'[S_m]$  for  $m \leq 1$  are similar to  $\mathcal{M}_\delta$ , i.e., the MDP before applying the preprocessing for the objective. ■

**Achievable points of the original MA.** The procedure above approximates the set of achievable points of some digitization  $\mathcal{M}_\delta$ . That is, the procedure computes two polyhedra  $\text{down}(Q)$  and  $\mathcal{P}$  such that

$$\text{down}(Q) \subseteq \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}_\delta}(\mathbb{O} \triangleright \mathbf{p})\} \subseteq \mathcal{P}.$$

To obtain an under- and an over-approximation for the original MA  $\mathcal{M}$ , the possible error introduced due to the digitization approach has to be considered as well. From Theorem 4.51 on page 81 we infer that the sets

$$\begin{aligned} A^- &= \{\mathbf{p}' \in \mathbb{R}^d \mid \forall \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ implies } \mathbf{p} \in \text{down}(Q)\} \text{ and} \\ A^+ &= \{\mathbf{p}' \in \mathbb{R}^d \mid \exists \mathbf{p} \in \mathbb{R}^d: \mathbf{p}' \in \varepsilon(\mathbb{O}, \mathbf{p}) \text{ and } \mathbf{p} \in \mathcal{P}\} \end{aligned}$$

satisfy

$$A^- \subseteq \{\mathbf{p} \in \mathbb{R}^d \mid \text{achieve}^{\mathcal{M}}(\mathbb{O} \triangleright \mathbf{p})\} \subseteq A^+.$$

Hence, any point in  $A^-$  is achievable in  $\mathcal{M}$  while the points in  $\mathbb{R}^d \setminus A^+$  are not achievable.

**Remark 5.27**

For the choice of the digitization constant  $\delta \in \mathbb{R}_{>0}$  we observe the following trade-off:

- If  $\delta$  is low, the number of considered digitization steps becomes high. This increases the run-times for the computation of an optimal point as in Algorithm 3.
- If  $\delta$  is high, the error arising from the digitization approach needs to be compensated by either analyzing more weight vectors or by repeating the analysis with a smaller constant  $\tilde{\delta} < \delta$ .

We can adjust the digitization constant during the computation without restarting the whole procedure from scratch as previously computed approximations for the MA remain valid. ■

## Chapter 6

# Experimental Evaluation

We discuss the performance of the presented approaches for multi-objective model checking. Section 6.1 presents details of our implementation of the Pareto-curve approximation algorithm. Experimental results on multi-objective MDPs are discussed in Section 6.2, where the focus lies on a comparison with the implementation in PRISM [KNP11]. Finally, we evaluate our experiments on three different case studies for multi-objective MAs in Section 6.3.

All experiments in this chapter were conducted on a 2.68 GHz Intel Core i7 processor with 6 GB RAM running Mac OS X.

### 6.1 Implementation

The Pareto curve approximation algorithm as discussed in Chapter 5 has been implemented in C++ within **StoRM**. The implementation considers Pareto queries of the form  $\text{pareto}^{\mathcal{M}}(\mathbf{opt} \circledast)$ , where  $\mathcal{M}$  is an MA or an MDP and  $\circledast$  is a list of  $d$  quantitative objectives with optimization directions  $\mathbf{opt} \in \{\min, \max\}^d$ . An under- and an over-approximation of the set of achievable points are returned such that the maximal distance between any point in the over-approximation and the closest point in the under-approximation is at most  $\eta$ . Here,  $\eta \in \mathbb{R}_{>0}$  is a predefined approximation threshold controlling the precision of the result.

The considered models are given in the PRISM language [KNP11] which has been extended to allow the specification of MAs. The transitions are stored by employing sparse matrices. We follow the value iteration-based approach for the computation of optimal points as presented in [FKP12] as well as Algorithm 2 on page 93 of this work. If one or more time-bounded until objectives for an MA are considered, the digitization approach is employed (cf. Section 4.3 and Section 5.4). We choose the digitization constant  $\delta \in \mathbb{R}_{>0}$  such that the maximal distance between two points in

	benchmark			PRISM		MultiStoRM		
	instance	#states	$\mathbb{O}$	prep	total	pts	prep	total
<i>consensus</i>	2.3_2	691	$\mathbb{P}, \mathbb{P}$	0.099	0.109	3	0.003	<b>0.006</b>
	2.4_2	1 517	$\mathbb{P}, \mathbb{P}$	0.183	0.198	3	0.003	<b>0.009</b>
	2.5_2	3 169	$\mathbb{P}, \mathbb{P}$	0.295	0.324	3	0.004	<b>0.016</b>
	3.3_2	17 455	$\mathbb{P}, \mathbb{P}$	0.763	0.910	3	0.025	<b>0.118</b>
	3.4_2	61 017	$\mathbb{P}, \mathbb{P}$	2.088	2.588	3	0.078	<b>0.412</b>
	3.5_2	181 129	$\mathbb{P}, \mathbb{P}$	4.816	6.331	3	0.232	<b>1.370</b>
<i>zeroconf(-tb)</i>	4	5 449	$\mathbb{P}, \mathbb{P}$	3.445	3.522	2	0.008	<b>0.060</b>
	6	10 543	$\mathbb{P}, \mathbb{P}$	7.591	7.738	2	0.013	<b>0.132</b>
	8	17 221	$\mathbb{P}, \mathbb{P}$	14.291	14.528	2	0.025	<b>0.323</b>
	2_14	29 572	$\mathbb{P}, \mathbb{P}$	29.733	29.878	2	0.041	<b>0.470</b>
	4_10	19 670	$\mathbb{P}, \mathbb{P}$	26.559	26.717	2	0.029	<b>0.390</b>
	4_14	42 968	$\mathbb{P}, \mathbb{P}$	69.273	69.489	2	0.056	<b>1.464</b>
<i>team-form.</i>	3	12 475	$\mathbb{P}, \mathbb{E}$	9.764	9.858	6	0.057	<b>0.153</b>
	4	96 665	$\mathbb{P}, \mathbb{E}$	226.994	227.946	6	9.392	<b>10.346</b>
	5	907 993	$\mathbb{P}, \mathbb{E}$	7 035.674	7 041.040	6	1 256.770	<b>1 273.680</b>
	3	12 475	$\mathbb{P}, \mathbb{E}, \mathbb{P}$	not supported <sup>1</sup>		12	0.065	<b>0.624</b>
	4	96 665	$\mathbb{P}, \mathbb{E}, \mathbb{P}$	not supported <sup>1</sup>		12	9.743	<b>12.267</b>
	5	907 993	$\mathbb{P}, \mathbb{E}, \mathbb{P}$	not supported <sup>1</sup>		12	1 320.162	<b>1 356.890</b>
<i>sched.</i>	5	31 965	$\mathbb{E}, \mathbb{E}$	incorrect <sup>2</sup>		–	0.022	<b>0.022</b>
	25	633 735	$\mathbb{E}, \mathbb{E}$	incorrect <sup>2</sup>		–	0.529	<b>0.529</b>
	50	2 457 510	$\mathbb{E}, \mathbb{E}$	incorrect <sup>2</sup>		–	2.092	<b>2.092</b>
<i>dpm</i>	100	636	$\mathbb{C}^{\leq}, \mathbb{C}^{\leq}$	0.015	0.132	9	0.006	<b>0.103</b>
	200	636	$\mathbb{C}^{\leq}, \mathbb{C}^{\leq}$	0.016	0.158	8	0.004	<b>0.148</b>
	300	636	$\mathbb{C}^{\leq}, \mathbb{C}^{\leq}$	0.014	0.172	6	0.004	<b>0.154</b>

Table 6.1: Run-times (in seconds) for our experiments on MDPs.

$\varepsilon(\mathbb{O}, \mathbf{p})$  (for objectives  $\mathbb{O}$  and arbitrary point  $\mathbf{p} \in \mathbb{R}^d$ ) is at most  $\eta/2$ . This choice is motivated by finding a compromise between our observations for high and low values for  $\delta$  as discussed in Remark 5.27 on page 110. Adjusting the digitization constant during the computation in order to increase the precision of the approximation is not necessary for this choice.

<sup>1</sup>PRISM issues an error, indicating that queries with more than two maximizing objectives are not supported.

<sup>2</sup>Both objectives yield infinite expected reward under any scheduler. PRISM does not detect this and gives an incorrect answer.

## 6.2 Experiments on MDPs

For the experiments on multi-objective MDPs we consider the same case studies as in [FKP12, FKN<sup>+</sup>11]. Our implementation (referred to as **MultiStoRM**) is compared with the implementation in **PRISM**<sup>3</sup> [KNP11]. Both implementations are based on [FKP12].

We check Pareto queries, assuming  $\eta = 0.001$  for the goal precision of the approximation and  $\varepsilon = 10^{-6}$  for the termination criterion of the value iteration-based computations (cf. Algorithm 2 on page 93).

Table 6.1 depicts the results. The table shows the considered instances of the different case studies together with the resulting number of states of the model. Column  $\mathbb{O}$  represents the analyzed objectives, where  $\mathbb{P}$  is an unbounded probabilistic objective,  $\mathbb{E}$  is an expected value objective, and  $\mathbb{C}^{\leq}$  is a cumulative expected reward objective representing that reward is collected for a certain number of transitions. For **PRISM**, we depict the total run-time of the procedure (excluding the time for the initial model building as we are only interested in the model analysis). We also indicate the time required for the preprocessing steps which guarantee assumptions 1-3 on page 88. For **MultiStoRM**, we additionally depict the number of computed optimal point (i.e., the number of iterations in Algorithm 1). All run-times are given in seconds.

We observe that **MultiStoRM** outperforms **PRISM** on *all* benchmarks. This is primarily caused by the fact that **PRISM** spends a huge fraction of run-time during preprocessing. For the experiments on *team-form.*, the preprocessing is also costly in **MultiStoRM**. The reason is that the maximal end component decomposition (which is done to assert finite reward, cf. Section 5.3.2) is expensive on this benchmark. **PRISM** can not answer the Pareto queries considering three objectives. Moreover, only our implementation correctly detects that both objectives for the instances of *sched.* yield infinite reward, while **PRISM** gives an incorrect answer.

## 6.3 Experiments on MAs

We conduct experiments on MAs with respect to three different case studies:

**Video streaming client.** We consider the video streaming client from Example 1.1 on page 1. It receives  $N$  packages and stores them into a buffer. During the playback of the video, the packages are processed. We analyze the possible strategies to start the playback w.r.t. combinations of the following objectives:

$\mathbb{E}_1$ : Minimize the expected buffering time until the playback is finished.

$\mathbb{E}_2$ : Minimize the expected number of buffer underruns during the playback.

$\mathbb{E}_3$ : Minimize the expected time to start the playback.

---

<sup>3</sup>We consider **PRISM** in version 4.3.1, obtained from its website [www.prismmodelchecker.org](http://www.prismmodelchecker.org).

$\mathbb{P}_1^<$ : Minimize the probability for a buffer underrun within two time units.

**Polling system.** The polling system is based on [Sri91, TKvdPS12]. It considers two stations, each having a separate queue storing up to  $Q$  jobs of  $N$  different types. The jobs arrive at Station  $i$  (for  $i \in \{1, 2\}$ ) with some rate  $\lambda_i$  as long as the queue of the station is not full. A server polls the two stations and processes the jobs by (nondeterministically) taking a job from a non-empty queue. The time for processing a job is given by a rate which depends on the type of the job. Erasing a job from a queue is unreliable, i.e., there is a 10% chance that an already processed job stays in the queue. For  $i \in \{1, 2\}$  we assume the following objectives:

$\mathbb{E}_i$ : Maximize the expected number of processed jobs of Station  $i$  until its queue is full.

$\mathbb{P}_i^<$ : Minimize the probability that the queue of Station  $i$  is full within two time units.

**Probabilistic mutual exclusion protocol.** This case study regards a mutual exclusion protocol based on [PZ86, TKvdPS12]. Three processes nondeterministically choose a job for which they need to enter the critical section. The amount of time a process spends in its critical section is given by a rate which depends on the chosen job. There are  $N$  different types of jobs. For each  $i \in \{1, 2, 3\}$  the following objective are considered:

$\mathbb{P}_i^<$ : Maximize the probability that Process  $i$  enters its critical section within one time unit.

We check Pareto queries where the goal precision of the resulting approximations is set to either  $\eta = 0.01$  or  $\eta = 0.001$ . For the value iteration-based computations we consider the threshold  $\varepsilon = 10^{-6}$ .

Table 6.2 summarizes the results of our experiments. We depict the different benchmark instances with the number of states of the corresponding model. Column  $\textcircled{O}$  indicates the considered combination of objectives. For the tested goal precisions ( $\eta = 0.01$  and  $\eta = 0.001$ ), the table shows the number of computed optimal points, the time for preprocessing, and the total run-time of the procedure (excluding model building). The run-times are depicted in seconds. A timeout (TO) represents that the computation time exceeds the time limit of two hours. In this case, we indicate the progress of the computation by depicting the number of points that have been computed so far as well as the precision of the obtained approximation (i.e., the value that is compared to  $\eta$ ).

The memory usage during the experiments did not exceed 2 GB.

We observe that queries which are analyzed on the underlying MDP are solved efficiently on large models with up to roughly 1.5 million states. If at least one time-bounded objective is considered, the run-times increase drastically due to the costly

	benchmark			$\eta=0.01$			$\eta=0.001$		
	instance	#states	$\mathbb{O}$	pts	prep	total	pts	prep	total
<i>stream</i>	30	1 426	$\mathbb{E}_1, \mathbb{E}_2$	20	0.02	0.52	51	0.05	6.49
	30	1 426	$\mathbb{E}_3, \mathbb{P}_1^{\leq}$	13	0.01	81.71	36	0.03	2 275.46
	30	1 426	$\mathbb{E}_1, \mathbb{E}_3, \mathbb{P}_1^{\leq}$	22	0.03	218.29	73	$TO(0.001)$	
	100	15 251	$\mathbb{E}_1, \mathbb{E}_2$	27	0.04	3.45	79	0.09	29.16
	100	15 251	$\mathbb{E}_3, \mathbb{P}_1^{\leq}$	11	0.02	707.83	11	$TO(0.006)$	
	100	15 251	$\mathbb{E}_1, \mathbb{E}_3, \mathbb{P}_1^{\leq}$	20	0.05	2 004.35	7	$TO(0.127)$	
	250	94 376	$\mathbb{E}_1, \mathbb{E}_2$	31	0.15	33.02	90	0.22	124.21
	250	94 376	$\mathbb{E}_3, \mathbb{P}_1^{\leq}$	11	0.13	5 348.42	1	$TO(\infty)$	
	250	94 376	$\mathbb{E}_1, \mathbb{E}_3, \mathbb{P}_1^{\leq}$	9	$TO(0.050)$		0	$TO(\infty)$	
	1000	1 502 501	$\mathbb{E}_1, \mathbb{E}_2$	41	2.41	2 591.67	113	$TO(0.001)$	
	1000	1 502 501	$\mathbb{E}_3, \mathbb{P}_1^{\leq}$	0	$TO(\infty)$		0	$TO(\infty)$	
	1000	1 502 501	$\mathbb{E}_1, \mathbb{E}_3, \mathbb{P}_1^{\leq}$	0	$TO(\infty)$		0	$TO(\infty)$	
<i>polling</i>	2.2	249	$\mathbb{E}_1, \mathbb{E}_2$	3	0.02	0.03	5	<0.01	0.02
	2.2	249	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	3	<0.01	19.67	7	<0.01	428.92
	2.2	249	$\mathbb{E}_1, \mathbb{E}_2, \mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	16	0.03	223.59	50	$TO(0.002)$	
	3.2	1 020	$\mathbb{E}_1, \mathbb{E}_2$	4	0.01	0.04	8	0.01	0.09
	3.2	1 020	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	4	<0.01	126.02	8	0.01	2 399.24
	3.2	1 020	$\mathbb{E}_1, \mathbb{E}_2, \mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	27	0.06	1 775.12	12	$TO(0.025)$	
	3.3	9 858	$\mathbb{E}_1, \mathbb{E}_2$	5	0.07	0.60	11	0.07	1.42
	3.3	9 858	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	7	0.02	2 510.43	2	$TO(0.113)$	
	3.3	9 858	$\mathbb{E}_1, \mathbb{E}_2, \mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	10	$TO(0.096)$		1	$TO(\infty)$	
	4.4	827 735	$\mathbb{E}_1, \mathbb{E}_2$	12	19.20	460.92	31	19.28	1 260.96
	4.4	827 735	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	0	$TO(\infty)$		0	$TO(\infty)$	
	4.4	827 735	$\mathbb{E}_1, \mathbb{E}_2, \mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}$	0	$TO(\infty)$		0	$TO(\infty)$	
<i>mutex</i>	2	13 476	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}, \mathbb{P}_3^{\leq}$	27	0.04	1 172.00	20	$TO(0.021)$	
	3	38 453	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}, \mathbb{P}_3^{\leq}$	24	0.12	6 350.44	3	$TO(0.659)$	
	4	83 344	$\mathbb{P}_1^{\leq}, \mathbb{P}_2^{\leq}, \mathbb{P}_3^{\leq}$	7	$TO(0.055)$		0	$TO(\infty)$	

Table 6.2: Run-times (in seconds) for our experiments on MAs.

<i>polling</i>		$\mathbb{E}_1$	$\mathbb{P}_1$		<i>stream</i>		$\mathbb{E}_1$	$\mathbb{P}_1$	
$N_Q$	#states		$\eta=0.01$	$\eta=0.001$	$N$	#states		$\eta=0.01$	$\eta=0.001$
3.2	1 020	0.01	0.84	6.92	30	1 426	0.01	0.42	2.76
3.3	9 858	0.09	7.24	70.32	100	15 251	0.11	2.77	25.42
4.4	827 735	27.54	878.24	TO	1000	1 502 501	64.70	361.29	3 213.15

Table 6.3: Run-times (in seconds) for single-objective model checking of MAs.

analysis of  $\delta$ -bounded objectives on the digitization. We are able to process queries considering up to four objectives within the time limit.

Comparing the two goal precisions, we see that the number of computed points that are necessary to obtain the more precise approximation increase by a factor of approximately three. In addition, a lower digitization constant has to be considered which often leads to timeouts in experiments with time-bounded objectives.

Our observations above reflect empirical results from single-objective model checking. To illustrate this, Table 6.3 depicts run-times for a selection of models and (single) objectives. Checking the respective time-bounded objectives takes much more time compared to the expected reward objectives. Furthermore, the choice of  $\eta$  strongly affects the run-time of the procedure.

# Chapter 7

## Conclusion

### 7.1 Summary

This thesis presented multi-objective model checking of MAs considering (un)bounded until and expected reward objectives. To this end, we combined notions from single-objective MAs and multi-objective MDPs. The main idea was to transform a given MA to an MDP on which the actual multi-objective analysis is performed. More precisely, Chapter 4 presented the following main results:

- **Multiple unbounded until and expected reward objectives can be analyzed on the underlying MDP of the MA.** We showed that the results obtained for the underlying MDP also hold for the original MA. For this, we proved that achievable objectives in the MA can also be achieved in the MDP and vice versa (cf. Theorem 4.1 on page 40 and Theorem 4.9 on page 45). We defined a transformation for arbitrary (possibly time-dependent) schedulers for the MA to (time-abstract) schedulers for the MDP such that both schedulers induce the same unbounded until and expected reward properties on the respective models.
- **Multi-objective model checking with one or more bounded until objectives can be conducted on a digitization of the MA.** This yields a sound and arbitrary precise approximation of the result for the original model (cf. Theorem 4.51 on page 81). The digitization approach is known from single-objective MA model checking but had to be generalized to arbitrary schedulers (instead of schedulers that only optimize a single objective). For this generalization, schedulers for the MA were transformed to schedulers of the digitization (which is in fact an MDP) and we showed the connection between both models under these schedulers.

We considered the value iteration-based approach of [FKP12] to check multiple ob-

jectives for the underlying MDP (or for the digitization) of an MA. In Chapter 5 we discussed the details of this approach and extended the original work w.r.t.

- ds-bounded until objectives (which originate from the above-mentioned digitization approach),
- expected *reachability* reward objectives ([FKP12] is restricted to expected *total* reward objectives), and
- arbitrary combinations of *maximizing* and *minimizing* expected reward objectives.

An implementation of the presented approaches was discussed in Chapter 6. We evaluated experiments considering three different case studies for MAs with up to 1.5 million states and up to four objectives.

## 7.2 Future Work

We see the following directions for future work:

**Scheduler synthesis.** In the thesis we focused on the (approximative) computation of the set of achievable points. The next step could be to synthesize a (preferably condensed representation of) a scheduler that achieves a selected point. The authors of [BCC<sup>+</sup>15] describe a promising approach for scheduler synthesis in single-objective model checking that employs decision trees as a representation for the considered schedulers. It would be interesting to lift this approach to multiple objectives.

**Sound computations.** As discussed in Remark 5.11 on page 94, the value iteration-based approach of Algorithm 2 on page 93 is not sound. Hence, it would be useful to adapt the algorithm to sound techniques like interval iteration [HM14] or policy iteration [Put94].

**Reductions of MAs.** To accelerate the presented procedures, possible reductions of the size of the given MA prior to its (multi-objective) analysis could be studied. One could consider the adaptation of ideas from single-objective model checking such as (strong or weak) bisimulation [EHZ10a, EHZ10b] or game-based abstraction [BFH<sup>+</sup>14].

**Long-run average objectives.** Long-run average objectives represent the fraction of time spent in a given set of states on the long run. Approaches for the analysis of such objectives exist for single-objective MAs [GHH<sup>+</sup>13] and multi-objective MDPs [BBC<sup>+</sup>11], both based on linear programming. It is open whether these ideas can be combined to enable the analysis of multiple long-run average objectives on MAs.

# Bibliography

- [ADD00] Robert B. Ash and Catherine Doléans-Dade. *Probability and Measure Theory*. Harcourt/Academic Press, 2000. (Cited on pages 7, 8, 20, 21, 24, and 48.)
- [BBC<sup>+</sup>11] Tomáš Brázdil, Václav Brozek, Krishnendu Chatterjee, Vojtech Forejt, and Antonín Kucera. Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. In *Proc. of LICS*, pages 33–42. IEEE CS, 2011. (Cited on pages 4 and 118.)
- [BCC<sup>+</sup>15] Tomáš Brázdil, Krishnendu Chatterjee, Martin Chmelik, Andreas Feller, and Jan Kretínský. Counterexample Explanation by Learning Small Strategies in Markov Decision Processes. In *Proc. of CAV*, volume 9206 of *LNCS*, pages 158–177. Springer, 2015. (Cited on page 118.)
- [BCS07] Hichem Boudali, Pepijn Crouzen, and Mariëlle Stoelinga. Dynamic Fault Tree Analysis Using Input/Output Interactive Markov Chains. In *Proc. of DSN*, pages 708–717. IEEE CS, 2007. (Cited on page 1.)
- [BDH96] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996. (Cited on page 86.)
- [BFH<sup>+</sup>14] Bettina Braitling, Luis María Ferrer Fioriti, Hassan Hatefi, Ralf Wimmer, Bernd Becker, and Holger Hermanns. MeGARA: Menu-based Game Abstraction and Abstraction Refinement of Markov Automata. In *Proc. of QAPL*, volume 154 of *EPTCS*, pages 48–63, 2014. (Cited on page 118.)
- [BHHK03] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003. (Cited on page 4.)
- [BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008. (Cited on pages 7 and 31.)

- [CFK<sup>+</sup>13] T. Chen, V. Forejt, M. Kwiatkowska, A. Simaitis, and C. Wiltsche. On Stochastic Games with Multiple Objectives. In *Proc. of MFCS*, volume 8087 of *LNCS*, pages 266–277. Springer, 2013. (Cited on page 4.)
- [Cla08] Edmund M. Clarke. The Birth of Model Checking. In *25 Years of Model Checking – History, Achievements, Perspectives*, volume 5000 of *LNCS*, pages 1–26. Springer, 2008. (Cited on pages 2 and 31.)
- [CMH06] Krishnendu Chatterjee, Rupak Majumdar, and Thomas A. Henzinger. Markov Decision Processes with Multiple Objectives. In *Proc. of STACS*, volume 3884 of *LNCS*, pages 325–336. Springer, 2006. (Cited on page 4.)
- [dA97] Luca de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997. (Cited on pages 99 and 101.)
- [EHZ10a] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. Concurrency and Composition in a Stochastic World. In *Proc. of CONCUR*, volume 6269 of *LNCS*, pages 21–39. Springer, 2010. (Cited on page 118.)
- [EHZ10b] Christian Eisentraut, Holger Hermanns, and Lijun Zhang. On Probabilistic Automata in Continuous Time. In *Proc. of LICS*, pages 342–351. IEEE CS, 2010. (Cited on pages 1, 11, and 118.)
- [EKVY08] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. Multi-Objective Model Checking of Markov Decision Processes. *LMCS*, 4(4), 2008. (Cited on pages 4, 84, 89, and 92.)
- [FKN<sup>+</sup>11] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. Quantitative Multi-objective Verification for Probabilistic Systems. In *Proc. of TACAS*, volume 6605 of *LNCS*, pages 112–127. Springer, 2011. (Cited on pages 4, 83, 84, 89, 98, and 113.)
- [FKP12] Vojtěch Forejt, Marta Kwiatkowska, and David Parker. Pareto Curves for Probabilistic Model Checking. In *Proc. of ATVA*, volume 7561 of *LNCS*, pages 317–332. Springer, 2012. (Cited on pages 4, 5, 31, 35, 37, 83, 84, 87, 89, 90, 92, 101, 111, 113, 117, and 118.)
- [GHH<sup>+</sup>13] Dennis Guck, Hassan Hatefi, Holger Hermanns, Joost-Pieter Katoen, and Mark Timmer. Modelling, Reduction and Analysis of Markov Automata. In *Proc. of QEST*, volume 8054 of *LNCS*, pages 55–71. Springer, 2013. (Cited on pages 2, 4, 7, 31, 39, 52, and 118.)
- [GHH<sup>+</sup>14] Dennis Guck, Hassan Hatefi, Holger Hermanns, Joost-Pieter Katoen, and Mark Timmer. Analysis of Timed and Long-Run Objectives for Markov Automata. *LMCS*, 10(3), 2014. (Cited on pages 7, 52, 53, 70, 71, and 81.)
- [GTH<sup>+</sup>14] Dennis Guck, Mark Timmer, Hassan Hatefi, Enno Ruijters, and Mariëlle Stoelinga. Modelling and Analysis of Markov Reward Automata. In *Proc.*

- of *ATVA*, volume 8837 of *LNCS*, pages 168–184. Springer, 2014. (Cited on pages 4, 27, 31, and 39.)
- [Her02] Holger Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *LNCS*. Springer, 2002. (Cited on page 15.)
- [HH12] Hassan Hatefi and Holger Hermanns. Model Checking Algorithms for Markov Automata. *ECEASST*, 53, 2012. (Cited on pages 2, 4, 5, 7, 39, 52, 53, 83, and 105.)
- [HM14] Serge Haddad and Benjamin Monmege. Reachability in MDPs: Refining Convergence of Value Iteration. In *RP*, volume 8762 of *LNCS*, pages 125–137. Springer, 2014. (Cited on pages 94, 101, and 118.)
- [Kat12] Joost-Pieter Katoen. GSPNs Revisited: Simple Semantics and New Analysis Algorithms. In *Proc. of ACSD*, pages 6–11. IEEE CS, 2012. (Cited on page 1.)
- [KNP11] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Proc. of CAV*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011. (Cited on pages 5, 111, and 113.)
- [MCB84] Marco Ajmone Marsan, Gianni Conte, and Gianfranco Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transactions on Computer Systems*, 2(2):93–122, 1984. (Cited on page 1.)
- [Neu10] Martin R. Neuhäuser. *Model checking Nondeterministic and Randomly Timed Systems*. PhD thesis, RWTH Aachen University, 2010. (Cited on pages 7, 10, and 24.)
- [Nor97] James R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. (Cited on pages 1, 16, and 17.)
- [NSK09] Martin R. Neuhäuser, Mariëlle Stoelinga, and Joost-Pieter Katoen. Delayed Nondeterminism in Continuous-Time Markov Decision Processes. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 364–379. Springer, 2009. (Cited on page 19.)
- [Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994. (Cited on pages 1, 4, 16, 83, 88, and 118.)
- [PZ86] Amir Pnueli and Lenore Zuck. Verification of Multiprocess Probabilistic Protocols. *Distributed Computing*, 1(1):53–72, 1986. (Cited on page 114.)

- 
- [Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995. (Cited on page 16.)
- [Sri91] Mandyam M. Srinivasan. Nondeterministic Polling Systems. *Management Science*, 37(6):667–681, 1991. (Cited on page 114.)
- [Tim13] Mark Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. PhD thesis, University of Twente, 2013. (Cited on page 7.)
- [TKvdPS12] Mark Timmer, Joost-Pieter Katoen, Jaco van de Pol, and Mariëlle Stoelinga. Efficient Modelling and Generation of Markov Automata. In *Proc. of CONCUR*, volume 7454 of *LNCS*, pages 364–379. Springer, 2012. (Cited on pages 1 and 114.)
- [Zie95] Günter M. Ziegler. *Lectures on Polytopes*. Springer, 1995. (Cited on page 85.)
- [ZN10] Lijun Zhang and Martin R. Neuhäüßer. Model Checking Interactive Markov Chains. In *Proc. of TACAS*, volume 6015 of *LNCS*, pages 53–68. Springer, 2010. (Cited on page 19.)