SAPIENZA
UNIVERSITÀ DI ROMA

Markov Representations: Learning in MDP Abstractions and Non-Markovian Environments

Department of Computer, Control and Management Engineering
Ph.D. in Engineering in Computer Science (XXXVI cycle)

Roberto Cipollone
ID number 1528014

Advisor	Co-Advisor
Giuseppe De Giacomo	Fabio Patrizi

Academic Year 2023/2024

Markov Representations: Learning in MDP Abstractions and Non-Markovian Environments
Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: cipollone.rt@gmail.com

Alla mia famiglia

Abstract

One of the main features we expect AI agents to have is being capable of autonomous decision-making in complex environments. Reinforcement Learning (RL) is a very general formulation for this learning problem because it focuses on training agents through the use of repeated attempts and numeric feedbacks. Due to the little prior knowledge required, RL already has a significant record of successes in many fields, including robotics, strategy games, finance, advertising, and fine-tuning of machine learning models, such as Large Language Models, recently.

Despite many efforts, improving the efficiency and generality of RL algorithms remains a very relevant research topic to this day. Although efficiency is a commonly shared objective among RL researchers, the development of general RL algorithms remains much less explored, in comparison. This should not be attributed to the lack of interest from the community. Rather, this is mainly motivated by the intrinsic complexity associated to learning in non-Markovian environments. However, both of these important research directions share one common need, that is, the selection of appropriate, Markovian representations of the environment state. In MDPs, which are already Markovian, such selection is often the intended result of the abstraction process, a central concept for Hierarchical Reinforcement Learning (HRL). In non-Markovian environments, on the other hand, a Markov state is not available from the start, and it should be constructed.

This thesis addresses both of these complementary directions. In the first part of this work, we will explore the concept of MDP abstractions in the context of HRL. Specifically, in two respective chapters, (i) this thesis proposes an approach for exploiting MDP abstractions, with the objective of improving learning efficiency; (ii) this work gives a clear formalization of how accurate and compositional MDP abstraction should be defined, contributing to embed the common intuitions behind HRL into applicable and precise notions. Then, in a second part of this work, I will discuss how RL algorithms can be also applied in presence of partial observations or complex non-Markovian dependencies. Specifically, (i) I analyze the expressive power of a recently introduced model, the Regular Decision Process (RDP), and how it relates to the well-known decision process for partial observations (POMDP); (ii) finally, the last chapter proposes an offline RL algorithm for learning near-optimal policies in RDPs, and it provides the associated sample efficiency guarantees.

Both the parts above, and this thesis as a whole, aim to contribute to the joint research effort to identify the necessary and sufficient information for effective decision making in RL. Selecting approximate state representations is essential for HRL, being focused on efficiency and abstract reasoning, as well as for RL in non-Markovian environments, because the states should preserve all the relevant past events and forget those that are irrelevant for future decisions.

Acknowledgments

Completing a PhD with great satisfaction is far from a foregone conclusion. I know very well that this has only been possible thanks to the people who have been by my side in these years, in various ways, and for this, I am truly grateful. My greatest thoughts go to my family, all of them, and to Sofia, my future wife. I am sure I will have the opportunity to demonstrate my gratitude towards them on several occasions, and I prefer not to list them on this small page, as well as the many friends who have been there for me. However, I would like to take a moment to thank the friends, colleagues, and other important people that I have met during these years of doctoral studies.

I would like to first thank my advisor, Giuseppe De Giacomo, who has greatly contributed to my passion for research and academic life. Presenting me with challenging research problems has been essential and highly motivating for me. His example has also left an impact: seeing him work hard in his office has pushed me in turn, on many days. Finally, I am confident that his extensive knowledge in logical reasoning and knowledge representation has allowed me to explore aspects of AI that many reinforcement learning researchers cannot claim to have delved into.

I extend my gratitude to Fabio Patrizi, my co-advisor, and Luca Iocchi. Both have facilitated the creation of a creative and lively work environment, where I felt like an equal contributor to research ideas. This was a strong incentive for me to arrive at the first paper submissions.

In the department, the working environment has been especially pleasant, thanks to my office mates and colleagues: Francesco, Gabriel, Marco, Antonio, Alessandro, Giuseppe, Ramon, Shufang, and the whole WhiteMech group. This group has allowed the creation of precious friendships and, with Gabriel, I was happy to share all the years of this doctorate. Finally, among the many collaborations at Sapienza, I would at least like to thank Nicolo' Brandizzi and Francesco Frattolillo.

I thank Anders Jonsson for his warm hospitality in his group at UPF in Barcelona during my visit period. The months in Barcelona were a time of very intense work for me, fortunately, but also a very balanced and enjoyable experience. I believe that Anders' contribution to both of these aspects was essential. I thank the entire research group, primarily for the friendships formed and the advanced training that has allowed me to pursue. I thank Gergely, Vicenç, Matteo, Vincent, Antoine, Germano, Ludovic, Lorenzo, Guillermo, Nneka, Sergio, Emma and Francielle. I hope to nurture these new friendships over time as I desire.

A final greeting goes to all those whom, regrettably, I may have overlooked in this little and unsatisfactory list. To all of you too, thank you. And to you, with this small volume, happy reading!

Contents

I Preliminaries 1

1 Introduction 3

1.1 Outline and Contributions 7

2 Background 11

2.1 Decision Processes 12

2.2 Classes of Decision Processes 16

2.3 Planning in MDPs 20

2.3.1 Finite Horizon 20

2.3.2 Infinite Horizon 21

2.4 Learning in MDPs 22

II Learning With MDP Abstractions 25

Introduction to part II 27

3 Exploiting MDP Abstractions 29

3.1 Introduction 29

3.2 Hierarchical Reinforcement Learning 31

3.3 Preliminaries and Formulation 35

3.4 Exploiting Abstractions With Reward Shaping 38

3.4.1 Reward Shaping for Episodic RL 39

3.4.2 The Algorithm 40

3.5 Abstraction Quality 41

3.6 Validation 46

3.6.1 Return-Invariant Shaping 47

3.6.2 Robustness to Modelling Errors 48

3.6.3 Interaction Task 49

3.7 Discussion 50

3.8 Proofs 52

4 Realizing MDP decompositions 57
4.1 Introduction . . . 57
4.2 Preliminaries . . . 59
4.3 Realizable Abstractions . . . 61
4.4 Properties . . . 65
4.5 Abstracting and Realizing . . . 69
4.6 Discussion . . . 76
4.7 Proofs . . . 79

III Learning in Non-Markov Decision Processes 91
Introduction to part III . . . 93

5 The Expressive Power of RDPs 95
5.1 Introduction . . . 95
5.1.1 Contributions . . . 97
5.1.2 Related Work . . . 99
5.2 Preliminaries . . . 101
5.3 The Expressive Power of RDPs . . . 104
5.3.1 Strict Relations . . . 105
5.3.2 Belief Covers . . . 107
5.3.3 POMDP Approximations . . . 109
5.3.4 Learning With Partial Observations via RDPs . . . 114
5.4 Discussion . . . 116
5.5 Proofs . . . 117

6 Offline Reinforcement Learning in RDPs 129
6.1 Introduction . . . 129
6.1.1 Contributions . . . 130
6.1.2 Related Work . . . 130
6.2 Preliminaries . . . 135
6.3 Offline RL in RDPs . . . 138
6.4 `RegORL`: Learning an Episodic RDP . . . 140
6.5 Theoretical Guarantees . . . 143
6.6 Sample Complexity Lower Bound . . . 145
6.7 Discussion . . . 146
6.8 Proofs . . . 147
6.8.1 Preliminaries . . . 147
6.8.2 RDP Properties . . . 148
6.8.3 Sample Complexity of `AdaCT-H` . . . 151

6.8.4 Sample Complexity of AdaCT-H-A 154
6.8.5 Distinguishability Parameters 159
6.8.6 RegORL With Subsampled VI-LCB 160
6.8.7 Sample Complexity Lower Bound 162

7 Conclusion 173

Bibliography 177

Nomenclature 197

Part I

Preliminaries

Chapter 1

Introduction

Autonomous decision-making is arguably one of the most important manifestations of intelligence, since it refers to the generic ability to select appropriate actions in response to external inputs and complex situations. As such, the study of autonomous decision-making is a fundamental topic that encompasses many branches of AI. Sequential Decision-Making (SDM) refers to the setting in which the AI agent is required to take a series of consecutive actions, possibly reacting to the outcomes previous decisions. In general settings, SDM is a complex problem, because the sequential interaction that takes place between the agent and the environment requires the agent to reason over sequences of past events and long-term outcomes. The term “environment” is commonly used to refer to everything the decision-maker interacts with (Russell and Norvig 2009). SDM can be instantiated as many distinct sub-problems, depending on some general assumptions. If the dynamics of the environment is assumed to be known, the setting is referred to as *planning* (Geffner and Bonet 2013). In the most general case, on the other hand, the environment is initially unknown and SDM must be solved through various forms of *learning*.

Reinforcement Learning (RL) is a very general and powerful formulation for learning problems in SDM (Sutton and Andrew G. Barto 2018). According to the RL formulation, at each decision step, the environment produces two outputs in response to the agent’s action: an observation and a reward. An observation is a piece of information that is provided to the agent in order to allow a more sensible selection of actions at the next time instant. Rewards, on the other hand, are quantitative measures of immediate performance. The action selection rule used by an agent is generally called *policy*. The exact formulation of an RL problem will be presented later, but, for the moment, we can informally say that the RL problem is the agent’s task of finding the policy that maximises some performance measure, defined over the sequence of rewards, when then environment dynamics is unknown. One of the strongest features of RL is that this flexible paradigm allows agents to learn from experience with very little prior knowledge required about the environment. As such, RL already has a significant record of successes for a variety of applications, ranging from robotics, navigation, strategy games, finance, advertising, and fine-tuning of other Machine Learning (ML) models, such as Large Language Models, recently.

However, reinforcement learning is only a very generic learning framework. In order to develop any specific RL algorithm, it is necessary to restrict the class of environments that are being considered. One of the strongest restrictions that is often assumed is related to how past events can impact future outcomes. On one extreme of the spectrum, there are environments in which it is safe to assume that the value of current actions are not affected by past decisions and outcomes at all, as is the case in classic multi-armed bandits. Thanks to this simplifying assumption, the current bandits' literature has been able to develop very efficient learning algorithms, with excellent performance guarantees (Lattimore and Szepesvári 2020).

Gradually moving toward more complex dynamics, we can consider the fully observable stochastic processes, in which the outcome of each agent's action depends both on the action and on the last observation of the environment. These can be naturally modelled as Markov Decision Processes (MDPs) (Puterman 1994). Although more complex, many fundamental principles that have been derived for bandits have been suitably extended for MDPs. In fact, MDPs are a very interesting middle ground both for research and for applications, because the increased expressiveness does not prevent the development of very efficient learning algorithms. In fact, RL in MDPs is the most common setting in both introductory textbooks (Sutton and Andrew G. Barto 2018) and research. The research field concerned with efficient learning in MDPs is far too extensive to be represented here, but we may recall some seminal works (Kearns and S. Singh 2002; Brafman and Tennenholtz 2003; Strehl, L. Li, et al. 2006; Jaksch, Ortner, et al. 2010; Sham Machandranath Kakade 2013; Dann and Brunskill 2015; Azar, Osband, et al. 2017), as well as other works that go beyond the finite and tabular setting and are suitable for MDP with very large or infinite state spaces (Schulman, Levine, et al. 2015; C. Jin, Z. Yang, et al. 2020).

Efficient RL algorithms for MDPs allowed us to expand the range of domains that can be efficiently solved. However, sample efficiency is not the only desirable property that learning algorithms should possess. Compositionality, interpretability, and policy reuse are all equally important in most settings, since they guarantee that agents solving new tasks can take advantage of components of the previous solutions. Hierarchical Reinforcement Learning (HRL) is a wide subfield of RL that aims to solve RL problems by exploiting the internal structure of the environment dynamics and/or the task structure, with the purpose of identifying independent subproblems and combining their solutions to obtain the global policy (Hutsebaut-Buysse, Mets, et al. 2022). With respect to efficiency, HRL methods also aim to be more scalable with respect to problems that could be efficiently solved with compositional approaches.

While in MDPs it is possible to pursue secondary learning objectives, such as the ones addressed by HRL, when the environment belongs to a more general and expressive class of decision processes, even finding near-optimal policies becomes a very demanding task. At the root of this major difference in complexity, there is the presence, or the absence, of an environment property called the Markov property. Specifically, the two Markov assumptions state that, at any step, the probability of the next observation and the next reward is conditionally independent on all past events, given the most recent observation and action. Using the notation that we will use from section 2.1, for every time step t, we write this condition as

$$o_{t+1} \perp o_0, a_1, \ldots, o_{t-1}, a_t \mid o_t, a_{t+1} \tag{1.1}$$

$$r_{t+1} \perp o_0, a_1, \ldots, o_{t-1}, a_t \mid o_t, a_{t+1} \tag{1.2}$$

We list these two separately, as they serve different purposes and we might refer to them independently. In fact, some systems could be Markovian in rewards or observations independently. Generally, we say that a decision process is non-Markovian to mean that it does not satisfy (1.1) or (1.2). Non-Markovian decision processes may arise due to many circumstances, but the most common cause is partial information.

Partially Observable MDPs (POMDPs) are arguably the most important group of non-Markovian decision processes, because they generalise classic MDPs with the introduction of a generic observation function (Åström 1965). POMDPs are very successful models, especially when modelling robotics and multi-agent systems. In fact, modelling the local agent's perceptions naturally gives rise to a partially observable environment, since on-board sensors cannot give complete information about the outside world. If we focus on more realistic interaction scenarios, we could really say that partial observations are always present. However, the added expressiveness is also the cause of the increased complexity that is required. This complexity involves all relevant resources, which are time, memory, and the number of environment interactions. As shown in Papadimitriou and Tsitsiklis (1987), while computing the optimal policy of a known MDP is P-complete for all horizon settings, computing the optimal policy of a known POMDP is PSPACE-hard in the horizon length, and it becomes undecidable for infinite horizons (Madani, Hanks, et al. 1999). When the model is unknown, the associated intractability result for RL in POMDPs has been stated in Krishnamurthy, Agarwal, et al. (2016). For comparison, the same problem is polynomial in MDPs over the same parameters. As a consequence of this fundamental difficulty, POMDP algorithms often avoid formal convergence guarantees but only rely on approximation techniques and methods based on appropriate Neural Networks architectures (Hausknecht and Stone 2015; Heess, Hunt, et al. 2015; Mnih, Badia, et al. 2016; Lample and Chaplot 2017; Igl, Zintgraf, et al. 2018; Kapturowski, Ostrovski, et al. 2019). On the other hand, algorithms that do provide formal efficiency guarantees must rely on assumptions that considerably restrict the class of POMDPs under consideration. In particular, there exist learning algorithms for POMDPs, provided that the environment satisfies some assumptions, such as undercompleteness (H. Guo, Cai, et al. 2022; C. Jin, Sham M. Kakade, et al. 2020), few-steps reachability (Z. D. Guo, Doroudi, et al. 2016), ergodicity (Azizzadenesheli, Lazaric, et al. 2016), few-steps decodability (Krishnamurthy, Agarwal, et al. 2016; Efroni, C. Jin, et al. 2022), or weakly-revealing (Q. Liu, Chung, et al. 2022). Broadly speaking, these requirements ensure that the environment either satisfies some regularity assumption on the transition function or that some quantifiable amount of information is continuously revealed about the hidden trajectory of states. In other words, they exclude sustained evolutions that can remain unpredictable and unobservable for an indefinite number of steps.

As demonstrated by such a rich body of research, identifying tractable subsets of POMDPs is a very relevant research topic, both from a theoretical perspective and for practical applications, which can hardly fit into the full observability assumption of MDPs. The Regular Decision Process (RDP) (Brafman and De Giacomo 2019) is a recently introduced model for non-Markovian environments, whose expressiveness falls between MDPs and POMDPs. These environments do not respect the Markov properties, but instead rely on the assumption that future outputs depend on past events in a regular way. Here, "regular" should be indented with its standard meaning in the context of formal languages and finite automata. Unlike the POMDP assumptions and subclasses considered above, RDPs can capture complex temporal dependencies with lasting effects that can extend arbitrarily far into the future. This model has been described in Brafman and De Giacomo (2019) using temporal logics over finite traces and regular languages. However, RDPs can be also equivalently expressed in terms of finite-state transducers. This is the notion that has been preferred in later works on RDPs (Abadi and Brafman 2020; Ronca and De Giacomo 2021; Ronca, Licks, et al. 2022), and, with some changes, it is the one adopted here. Regardless of the precise formalism, this model is fundamentally defined by the regularity property that its traces satisfy. As a consequence, the expressive power of RDPs goes beyond that of MDPs, but, as shown in Brafman and De Giacomo (2019), it may not exceed that of POMDPs. Some RL algorithms have already been developed for RDPs in Abadi and Brafman (2020), Ronca and De Giacomo (2021), and Ronca, Licks, et al. (2022).

As we have seen, a portion of the current RL literature is focusing on two distinct aspects for effective decision-making. On one hand, effective MDP algorithms need to satisfy additional requirements, such as sample efficiency, policy reuse, and interpretability. On the other hand, provably correct RL algorithms for expressive non-Markovian environment models are still relatively scarce. These two research directions, HRL and RL for non-Markovian domains, are often regarded as completely independent branches of the Reinforcement Learning literature. Although they originally arise from different subfields of RL, they both address one common need from two separate perspectives: *identifying an appropriate state representation* that approximately satisfies the Markov assumptions for the given environment. In HRL, the input domain is usually an MDP, which is already Markovian. However, for constructing MDP abstractions, it is necessary to identify an alternative state representation that preserves some of the original environment dynamics while ignoring most of the other details. Its purpose is to consider a proxy model of the environment, which is more effective for finding near-optimal policies and reusing their components. In Non-Markovian domains, on the other hand, a Markov state is not available from the start. Therefore, it is necessary to construct a new state representation of the environment that at least satisfies the Markov assumption on rewards (Hutter 2009). We could informally phrase these two aspects as follows. While HRL is concerned with what an agent could "forget" about the full state, without losing near-optimality, RL for non-Markovian domains focuses on what to "remember" about past events, in order to become near-optimal.

1.1 Outline and Contributions

This thesis presents my recent research activity and the most relevant scientific contributions that I developed during my PhD. Since I have been active in both subfields of RL that we discussed in the introduction, namely, RL with MDP abstractions and RL for non-Markovian decision processes, this thesis is organised accordingly. After part I – Preliminaries, there are two other parts, one for each macro topic: part II – Learning With MDP Abstractions, and part III – Learning in Non-Markov Decision Processes. Both parts contain two chapters each. Since each pair of chapters largely shares a common context, the two parts start with two respective introductions on pages 27 and 93. The content of each chapter and their contributions are summarised in the following paragraphs. Then, a more detailed list of contributions can be found in each chapter.

2 – Background Rather than presenting each contribution separately, this thesis aims to give an organic and integrated view of the results. Therefore, this initial chapter sets the common language, the notation, and the basic definitions that will be used throughout the thesis in all the following chapters. Since these preliminaries should also be appropriate for RL in non-Markovian environments, the RL problem is presented with generality by talking about histories, the values of histories and some classes of decision processes.

3 – Exploiting MDP Abstractions Opening part II, this is the first of the four chapters of contributions. Here, the abstraction of any MDP (ground) is defined as another MDP (abstract), together with a function that maps the states of the two decision processes. This chapter presents an intuitive RL algorithm that optimises the policy for the original MDP, by exploiting the additional prior knowledge coming from the abstract MDP simulator. Intuitively, through a specific form of Reward Shaping, the solution of the abstract decision process is used to construct an exploration heuristic for collecting samples in the ground MDP. This allows to drive exploration towards more promising regions of the state space, while still guaranteeing optimal convergence when the abstraction contains severe modelling errors. The effectiveness of the algorithm has been tested experimentally.

For each MDP and associated abstraction, the theoretical analysis quantifies the sub-optimality gap between the induced exploration policy and the original optimum. Apart from our specific application, the result is more generally applicable and strongly improves on similar results in the HRL literature. In the process, the analysis identifies some relevant parameters that suggest how abstract MDP and the associated state mapping should be defined. These observations serve as a basis for the work of chapter 4.

Part of this chapter is based on previous published work, result of a joint effort, which appeared in the paper: Roberto Cipollone, Giuseppe De Giacomo, Marco Favorito, Luca Iocchi, and Fabio Patrizi (2023a). “Exploiting Multiple Abstractions in Episodic RL via Reward Shaping”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37, pp. 7227–7234. Preliminary results have also been accepted and presented at the PRL workshop (Bridging the Gap Between AI Planning and Reinforcement Learning), co-located within the IJCAI 22 conference.

4 – Realizing MDP decompositions The previous contribution already identified some important principles for defining accurate MDP abstractions. However, the algorithmic approach, which is based on Reward Shaping, is not fully compositional. This chapter directly addresses some fundamental questions that recur in the HRL literature. Specifically, it aims to answer the following questions: How should MDP abstraction be defined? How can they be used inside RL algorithms in a compositional way?

The theoretical contribution of this chapter is to propose the original concept of *realisable abstractions*. A realizable abstraction is a decision process whose individual abstract actions can be realized as specific sub-policies in the ground MDP. Specifically, it is possible to associate to each abstract state and action a specific sub-policy which achieves similar transition probabilities and expected cumulative rewards. This ensures that abstract states are good representatives for the expected returns that are really achievable in the ground MDP. Regarding the formal properties, the chapter also demonstrates that realizable abstract policies can be translated into near-optimal ground policies, in a compositional way. Finally, the chapter also analyses how to implement two complementary processes: realizing abstractions and abstracting ground dynamics.

Part of this chapter is based on original work, result of a joint effort, whose full list of authors is: Roberto Cipollone, Luca Iocchi, Matteo Leonetti. This work will soon be included as part of a future submission to an AI conference.

5 – The Expressive Power of RDPs Opening part III, this chapter first provides the necessary background for planning in Partially-Observable MDPs (POMDPs), which are the most famous class of non-Markovian environments. The Regular Decision Process (RDP) is a different non-Markovian model that has recently been proposed in the literature. Since this is the model studied in both chapters 5 and 6, we discuss how they work in detail and propose a specific formalisation for stochastic environments.

The objective of this chapter is to analyse the expressive power of RDPs and put them in relation with POMDPs. Our contributions are multiple: we prove that RDPs are strictly less expressive than POMDPs; we characterise which POMDPs admit an equivalent RDP; after providing a notion of approximation between heterogeneous decision processes, we show that some POMDP classes admit RDP approximations. Thanks to the properties above, the chapter observes that, in many cases, RDP algorithms can be applied to environments that have been originally defined as POMDPs, without any modification. Finally, we show that the exponential lower bound for POMDPs also applies to RDPs. Regardless of this result, the RDP formalism remains relevant thanks to the simplicity of planning in these domains.

This chapter is primarily based on original work which may be included as part of a future submission.

6 – Offline Reinforcement Learning in RDPs This final chapter of contributions proposes an Offline Reinforcement Learning algorithm for RDPs. In Offline RL, the input of the algorithm is a dataset of interactions, collected with some unknown behaviour policy. The objective of the algorithm is to estimate a near-optimal policy, without further exploration. To the authors' knowledge, the proposed method is the first Offline RL algorithm for RDP, with formal efficiency guarantees. In particular, this work shows that the algorithm satisfies a high-probability sample efficiency upper bound, whose expression is polynomial in all relevant parameters. This chapter also derives a general sample-efficiency lower bound for RDPs. The algorithm has two important features. Firstly, its computational complexity is very low. Secondly, the method is fully modular because it internally transforms the input data into an associated dataset for a Markovian environment. Therefore, the output dataset can be optimised with any off-the-shelf Offline RL algorithm for MDPs from the literature.

Part of this chapter is based on previous published work, result of a joint effort, which appeared in the paper: Roberto Cipollone, Anders Jonsson, Alessandro Ronca, and Mohammad Sadegh Talebi (2024). “Provably Efficient Offline Reinforcement Learning in Regular Decision Processes”. In: *Thirty-Seventh Conference on Neural Information Processing Systems, NeurIPS 2024*.

Chapter 2

Background

This chapter introduces some of the main definitions and notions that will be used throughout the thesis. Although most of the material presented here can be found in the literature, this chapter aims to provide an organic presentation of the topics that we will encounter in the following parts. A classical introduction to Reinforcement Learning would usually begin from Markov Decision Processes. Some of the contributions of this thesis, however, involve more general environment dynamics that fall well beyond this environments class. Thus, we necessarily need to approach the topic of environments models and Reinforcement Learning from a very general perspective. We first introduce the basic notation.

Basic Notation Generic sets and their elements are usually written in lowercase and calligraphic letters, respectively, such as $\mathcal{X}$ and $x \in \mathcal{X}$. The juxtaposition of two sets $\mathcal{X}\mathcal{Y}$ is an abbreviation of $(\mathcal{X} \times \mathcal{Y})$. Similarly, xy is preferred to (x, y) whenever this cannot be misunderstood with a product. For $N \in \mathbb{N}_+$, $[N]$ is an abbreviation for the set $\{0, \dots, N-1\}$. The indicator function $\mathbb{I}(\varphi)$ evaluates to 1 if the condition φ is true, 0 otherwise. For any function f, we write $f \equiv g$, if $f(x) = g(x)$ for all x, or $f \equiv a$, if $f(x) = a$, for some constant a.

The set of probability distributions on a set $\mathcal{X}$ is written $\Delta(\mathcal{X})$. If $\mathcal{X}$ is finite, the set $\Delta(\mathcal{X})$ includes every function $f : \mathcal{X} \to [0, 1]$ such that $\sum_{x \in \mathcal{X}} f(x) = 1$. The support of some f is written $\operatorname{supp}(f) \coloneqq \{x \in \mathcal{X} \mid f(x) > 0\}$. We write $z \sim \mu$ with $\mu \in \Delta(\mathcal{X})$ if z is a random variable distributed according to μ. For $x \in \mathcal{X}$, we use the Kronecker delta $\delta_x \in \Delta(\mathcal{X})$ for the discrete and deterministic probability distribution centred on x, that is $\delta_x(x') \coloneqq \mathbb{I}(x' = x)$. A function g returning a probability distribution is written $g : \mathcal{X} \to \Delta(\mathcal{Y})$. In this case, we write $g(x)$ to denote the resulting probability distribution (which, for discrete sets, is itself a function) or $g(y \mid x)$ to represent the probability of y in $g(x)$. Whenever appropriate, we also interpret functions and probability distributions $f \in \Delta(\mathcal{X})$ as column vectors, for some fixed order of $\mathcal{X}$. Then, both $f(x)$ and the inner product $\langle f, \delta_x \rangle$ evaluate to the probability of x in f. Similarly, when explicitly stated, a function $g : \mathcal{X} \to \Delta(\mathcal{Y})$ may also be interpreted as a matrix of $|\mathcal{X}|$ rows and $|\mathcal{Y}|$ columns.

Throughout this thesis, O, Ω, Θ are the symbols for the big-O asymptotic notation, and $\tilde{O}, \tilde{\Omega}, \tilde{\Theta}$ are the respective symbols if poly-logarithmic terms are ignored. For completeness, their meaning is summarised here. Given a function $g : \mathbb{N} \to \mathbb{N}$, we write $O(g)$ (and $\Omega(g)$) for the set of functions $f : \mathbb{N} \to \mathbb{N}$ for which there exist $c > 0$, $n_0 \in \mathbb{N}$, such that $f(n) \leq c\, g(n)$ (and $f(n) \geq c\, g(n)$, respectively), for all $n \geq n_0$. Also, $\Theta(g) \coloneqq \Omega(g) \cap O(g)$.

2.1 Decision Processes

As common in AI, we represent decision-making problems as agent–environment interactions. A *decision process* is any environment model that represents a discrete-time stochastic control process. In other words, in a decision process, the agent and the environment take turns in selecting some variables. At each time step, the agent selects one of the available actions, $a \in \mathcal{A}$, and the environment selects an observation $o \in \mathcal{O}$ and a reward $r \in \mathcal{R}$. Actions and observations are visible to both. The rewards $\mathcal{R} \subset \mathbb{R}$, on the other hand, are scalar values that are used to evaluate the agent's performance. This interaction gives rise to a random sequence $o_0\, a_1\, r_1\, o_1\, a_2\, \dots$. Depending on the horizon setting, which might be finite or infinite, this sequence is terminated at some fixed time step $H \in \mathbb{N}_+$, or it continues forever. We refer to the sequence generated by this interaction as *trace*. It may be convenient to include a null dummy reward r_0 at the initial time step, $r_0 o_0$, and an irrelevant dummy action a_{H+1} at the final step. Thus, we can define traces as follows:

$$r_0\, o_0\, a_1\, \dots r_H\, o_H\, a_{H+1} \in (\mathcal{ROA})^{H+1} \qquad \text{(finite horizon)} \tag{2.1}$$

$$r_0\, o_0\, a_1\, r_1\, o_1\, a_2 \dots \qquad \in (\mathcal{ROA})^{\infty} \qquad \text{(infinite horizon)} \tag{2.2}$$

depending on whether we consider finite or infinite horizons. For either case, we define finite traces containing t triples as elements of the set $\mathcal{T}_t \coloneqq (\mathcal{ROA})^t$. Similarly, we call *history* any finite sequence of actions and observations, $h_t = o_0 a_1 \dots o_{t-1} a_t$, and we define the set of histories of length t as $\mathcal{H}_t \coloneqq (\mathcal{OA})^t$. With $\mathcal{T}$ and $\mathcal{H}$ we denote the sets of all possible traces and histories, respectively. Thus, these are defined $\mathcal{T} \coloneqq \cup_{t \in 0,\dots,H+1} \mathcal{T}_t$ and $\mathcal{H} \coloneqq \cup_{t \in 0,\dots,H} \mathcal{H}_t$, for the finite horizon, and $\mathcal{T} \coloneqq \cup_{t \in \mathbb{N}} \mathcal{T}_t$ and $\mathcal{H} \coloneqq \cup_{t \in \mathbb{N}} \mathcal{H}_t$, for the infinite horizon. The length of a trace or history, denoted as $|\cdot|$, is defined as the number of tuples it contains. The history h_0 is the empty sequence.

Unlike other works that also define values on sequences (Hutter 2009), this thesis distinguishes between traces and histories, because rewards may or may not be regarded as observable in the strict sense. With the convention that histories omit rewards, it is possible to represent the fact that the agent may not condition its decisions on past rewards. The main motivation for this choice is to remain consistent with a large part of the POMDP literature. However, we notice that this is not a true restriction, since rewards may also be included as part of the observations, whenever it is appropriate to regard them as observable.

Generic rewards r_t, observations o_t and actions a_t can be all considered random variables. In this thesis, if x is some free random variable, $\mathbb{P}(x)$ represents the probability distribution over every value of x. In discrete random variables, this is a function or a vector of probabilities. On the other hand, when x appears as a quantified variable, or is bound as in $\sum_x$, then $\mathbb{P}(x)$ should be interpreted as the scalar probability that the associated random variable takes the single value x. This is done to avoid cumbersome notations such as $\mathbb{P}(O_t = o_t, R_t = r_t \mid O_{t-1} = o_{t-1}, A_{t-1} = a_{t-1}, \dots)$, and only write $\mathbb{P}(o_t, r_t \mid o_{t-1}, a_{t-1}, \dots)$ without ambiguity, when these variables are properly quantified.

Acting Acting in a decision process amounts to following some action selection rule, commonly called *policy.* In the most general case, policies are functions from histories to the following actions, and they may be stochastic. Thus, a policy is a function $\mathcal{HO} \to \Delta(\mathcal{A})$, and the set of all policies is $\Pi \coloneqq \Delta(\mathcal{A})^{\mathcal{HO}}$. Like in Puterman (1994), we will also consider more restricted policy classes. We say that some $\pi \in \Pi$ is a *deterministic* policy if there exists some $\pi_\mathsf{d} : \mathcal{HO} \to \mathcal{A}$ such that $\pi(ho) = \delta_{\pi_\mathsf{d}(ho)}$, for every $ho \in \mathcal{HO}$. The set of deterministic policies is $\Pi_\mathsf{d} \subset \Pi$. With a slight abuse of notation, we will often refer to π_d as the deterministic policy and say that the output is simply the selected action instead of the deterministic distribution.

The input space of policies may also be restricted. The set of *stationary* policies Π_s is composed of all $\pi \in \Pi$, for which there exists some $\pi_\mathsf{s} : \mathcal{O} \to \Delta(\mathcal{A})$ such that $\pi(ho) = \pi_\mathsf{s}(o)$, for every $ho \in \mathcal{HO}$. In other words, stationary policies only depend on the last observation. Under the infinite horizon, the terms *Markovian* and stationary are used interchangeably. In the finite-horizon setting, instead, the set of Markovian policies Π_m is the set of all $\pi \in \Pi$, for which there exists a collection of H stationary policies $\pi_\mathsf{m} \coloneqq \{\pi_t\}_{t \in [H]}$, such that, for every $t \in [H]$ and $ho \in \mathcal{H}_t\mathcal{O}$, $\pi(ho) = \pi_t(o)$. So, Markovian policies only depend on the last observation and, under the finite horizon, they may also access the current time step. The policies just defined may be combined arbitrarily. For example, the important class of stationary deterministic policies is $\Pi_\mathsf{s} \cap \Pi_\mathsf{d}$. For simplicity, we will often use $\pi_\mathsf{d}, \pi_\mathsf{s}, \pi_\mathsf{m}$ directly and call them policies.

Evaluating Decision processes define both the environment dynamics and the task that the agent should accomplish. In fact, the only purpose of rewards is to provide a scalar measure to be maximised by the agent's actions. More precisely, the agent's task is to maximise the expected return, which is the expected sum of future rewards (Puterman 1994; Sutton and Andrew G. Barto 2018). Depending on the horizon setting considered, the *return* at time $t \in \mathbb{N}_+$ is a random variable defined as follows:

$$g_t \coloneqq \sum_{i=t,\dots,H} r_i \qquad \text{(finite horizon)} \tag{2.3}$$

$$g_t \coloneqq \sum_{i=t,\dots} \gamma^{i-t} r_i \qquad \text{(infinite horizon)} \tag{2.4}$$

where $\gamma \in (0, 1)$ is a fixed discount factor, that gives more importance to immediate rewards, compared to those distant in the future. For either horizon setting, at any time step t, and history $h_t o_t \in \mathcal{H}_t \mathcal{O}$, we define the value of any policy $\pi \in \Pi$ in a decision process $\mathbf{D}$ and $h_t o_t$ as

$$V_t^\pi(h_t o_t) \coloneqq \mathbb{E}[g_{t+1} \mid \mathbf{D}, \pi, h_t o_t] \tag{2.5}$$

where π is used to select a_{t+1} and all the following actions. If, in addition to the past variables $h_t o_t$, the following action is also set, for $t \geq 1$, we write:

$$Q_t^\pi(h_t) \coloneqq \mathbb{E}[g_t \mid \mathbf{D}, \pi, h_t] \tag{2.6}$$

To emphasize the differences between the two functions, we observe that $h_t o_t$ is the sequence of observable variables up to o_t. On the other hand, the sequence $h_t o_t a_{t+1}$, which we also write as h_{t+1}, is the sequence of observables with an additional action taken by the agent. For MDPs, $Q_t^\pi(h_t)$ is commonly called "Q-function". Now, if $\mu \in \Delta(\mathcal{O})$ is the distribution over the initial observation in $\mathbf{D}$, without referring to any history, we define the value of a policy π as

$$V_\mu^\pi \coloneqq \mathbb{E}[V_0^\pi(o_0) \mid \mu] \tag{2.7}$$

The dependency on the decision process will be often left implicit, when clear from context. In particular, any decision process defines a unique initial distribution μ.

Note that V_μ^π is a scalar. A policy $\pi^* \in \Pi$ is optimal if it satisfies $V_\mu^{\pi^*} = \sup_{\pi \in \Pi} V_\mu^\pi$. $V_\mu^{\pi^*}$ is often written as V_μ^*. When planning and learning in decision processes, it is often impossible or impractical to find an optimal policy (owing to numerical estimates of known and unknown quantities). Thus, our learning objective is often to reach near-optimality. Given an accuracy parameter $\varepsilon \geq 0$, we say that a policy π is ε-optimal if $V_\mu^* - V_\mu^\pi \leq \varepsilon$.

The two horizon settings often need separate treatments and definitions, such as the one of eqs. (2.3) and (2.4), because their arguments may rely on slightly different mathematical arguments. However, the two are closely related. Short finite horizons H are related to small discount factors γ. In fact, in the infinite-horizon setting, there is an effective finite horizon that scales proportionally to $1/(1-\gamma)$, after which the rewards play little role in determining the overall value, due to the geometric scaling. This relationship is well represented by Kearns and S. Singh (2002, Lemma 2). This was originally stated for MDPs, but as its proof does not depend on the Markov assumptions, it can also be formulated for generic decision processes, as we do here. For any $t \in \mathbb{N}$, let $g_{t,H}$ and $g_{t,\gamma}$ be the respective returns of eqs. (2.3) and (2.4), for the finite- and infinite-horizon settings. If

$$H - t \geq \frac{1}{1-\gamma} \log \left(\frac{1}{\varepsilon(1-\gamma)} \right) \tag{2.8}$$

then, $g_{t,H} \leq g_{t,\gamma} \leq g_{t,H} + \varepsilon$.

Planning and Learning The most common task in decision processes is finding a (near-)optimal policies using the least amount of resources. Depending on the prior knowledge that is assumed, the setting can be phrased as a planning or a learning problem. In stochastic *planning* (Geffner and Bonet 2013), it is assumed that the explicit decision process is known. Thus, the output policy can be directly computed from the known conditional distribution generating the output trace, without any interaction with the environment. For concreteness, we postpone the treatment of specific planning algorithms after some classes of decision processes have been defined.

When *learning* in decision processes, on the other hand, since the environment dynamics remains unknown, rewards and observations are only accessible via sampling. As described at the beginning of this section, this thesis considers a very classic sampling protocol. Starting from some random initial observation, the agent and environment interleave their responses. The task of the learning agent is to find some near-optimal policy, possibly satisfying some additional efficiency constraints in the process. This is what is commonly referred to as the *Reinforcement Learning* (RL) problem (Sutton and Andrew G. Barto 2018). Learning requires multiple interactions with the environment. In some decision processes, actions selected at the beginning of the episode may have a strong impact on future returns. Thus, to ensure sufficient exploration from the initial distribution, the environment must be regularly *reset* during learning. This may be done every H steps in both horizon settings, or at random stopping times, specifically for infinite horizons. In fact, if the episode is interrupted at each step with probability $1-\gamma$, then the sum of all rewards collected so far is an unbiased estimate of eq. (2.4) (Puterman 1994). Since this thesis focuses mainly on RL in decision processes, not planning, when we say “given an MDP”, we only require simulator access to it, one that allows episode-based sampling, as specified in this interaction protocol.

The Efficiency of Reinforcement Learning Unlike in planning, the most precious resource that RL algorithms use is the environment that generates the samples. Thus, under the common requirement that RL methods must generally be tractable in time and memory, the efficiency of each algorithm is commonly measured in relation to the number of samples that they require. More precisely, in RL theory, the *sample complexity* of an RL algorithm is often quantified with *Probably Approximately Correct* (PAC) guarantees (Fiechter 1994). Such statements involve an accuracy parameter $\varepsilon > 0$, a confidence parameter $\delta > 0$, and possibly other parameters that depend on the specific decision process. An algorithm is said to be (ε, δ)-PAC for a class of decision processes $\mathcal{D}$ if, for every $\mathbf{D} \in \mathcal{D}$, the algorithm returns a policy that is ε-optimal in $\mathbf{D}$, with probability at least $1 - \delta$. For many authors, being PAC also implies that the algorithm uses a number of samples that is polynomial in all relevant parameters, including $|\mathcal{O}|$, $|\mathcal{A}|$, $1/\varepsilon$, $\log(1/\delta)$ and the horizon H or $1/(1-\gamma)$. This is a sensible convention for algorithms that learn in Markov Decision Processes, which is the most common setting from the literature. However, the same complexity class cannot be required for more complex decision processes, as they could be inherently intractable in these quantities.

Another very common performance measure is called *regret* (Auer, Cesa-Bianchi, et al. 2002). Unlike sample efficiency, regret guarantees measure the sub-optimality of the algorithm throughout learning. If we denote by T_j the global time step at which episode j is initiated, and by π_j the policy used in episode j, then the regret of an RL algorithm at time T can be defined as: $\sum_{j \colon T_j < T} V^*_\mu - V^{\pi_j}_\mu$. As such, regret is measured as a function of T, often in asymptotic notation. Similarly to sample efficiency, any regret guarantee statement can be made in expectation or as a high-probability statement. For other performance guarantees, the reader can refer to Dann, Lattimore, et al. (2017).

Finally, the performance of RL algorithms can also be analysed and compared empirically, both by estimating the value of the final policy and by observing the associated learning curves. Clearly, both of these estimates should come with empirical estimates of their average and standard deviation, in order to account for the intrinsic stochasticity of the process or the algorithm.

2.2 Classes of Decision Processes

The previous section introduced values, policies, and all the most relevant variables for RL. For greatest generality, these elements were defined as individual random variables. However, an efficient RL algorithm must rely on some additional structure and independence assumptions within the decision process. Such dependencies are captured by classes of decision processes, such as MDPs, that restrict how future observations and rewards depend on past events. It is worth emphasising that these models are possible formalizations of the environment dynamics over actions, observations, and rewards. To be precise, these are *not* environments themselves, and they may or may not be appropriate formalizations, depending on the decision process at hand. Before proceeding, it is necessary to establish some basic notation related to finite automata, which will be needed in defining one of the decision processes in which we are interested.

In the context of formal languages, alphabets are finite sets of arbitrary symbols. If Σ is an alphabet, any $\sigma \in \Sigma$ is a symbol. The Kleene star applied to an alphabet, Σ^*, is the set of all elements that belong to the free monoid constructed over Σ. In other words, Σ^* contains all sequences $\sigma_1 \dots \sigma_n \in \Sigma^*$, also called strings, composed of a finite number of arbitrary concatenations over the elements in Σ.

A *Deterministic Finite Automaton* (DFA) (Rabin and Scott 1959) is a tuple $\langle \mathcal{Q}, \mathcal{F}, \Sigma, \tau, q_0 \rangle$, where $\mathcal{Q}$ is a finite set of states, $\mathcal{F} \subseteq \mathcal{Q}$ is the set of final states, $q_0 \in \mathcal{Q}$ is the initial state, Σ is the input alphabet and $\tau : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is the transition function. With $\bar{\tau} : \mathcal{Q} \times \Sigma^* \to \mathcal{Q}$ we denote the transition function, extended over strings, defined as $\bar{\tau}(q, \lambda) \coloneqq q$, where λ is the empty string, and $\bar{\tau}(q, x\sigma) \coloneqq \tau(\bar{\tau}(q, x), \sigma)$, for $x \in \Sigma^*$ and $\sigma \in \Sigma$. A DFA is an acceptor on the set of strings Σ^*, where some string $x \in \Sigma^*$ is accepted if and only if $\bar{\tau}(q_0, x) \in \mathcal{F}$.

A deterministic *finite state transducer* (Moore machine) is a tuple $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$, where $\mathcal{Q}, q_0, \Sigma, \tau$ are defined as in a DFA, Ω is a finite set of output symbols, and $\theta : \mathcal{Q} \to \Omega$ is the output function. The output function, extended over strings, is $\bar{\theta} : \mathcal{Q} \times \Sigma^* \to \Omega^*$, defined as $\bar{\theta}(q, \lambda) \coloneqq \theta(q)$ and $\bar{\theta}(q, \sigma x) \coloneqq \theta(q)\, \bar{\theta}(\tau(q, \sigma), x)$. Each Moore machine defines a unique function from any string in Σ^* to some string in Ω^* of the same length, that is $x \mapsto \bar{\theta}(q_0, x)$. A Mealy machine is defined with the same tuple of a Moore machine, except for the output function $\theta : \mathcal{Q} \times \Sigma \to \Omega$, which also receives an input symbol for generating each output. The two finite-state transducers, Mealy and Moore, have exactly the same expressive power, and they can be translated one into the other, if some marginal padding symbol is ignored.

We can now define the most important classes of decision processes for this thesis. We recall that decision processes can be regarded as having a finite or infinite horizon. The following definitions are valid for both settings, provided that the set of histories $\mathcal{H}$ is adapted as anticipated at the beginning of section 2.1. The horizon H or the discount factor γ are also necessary in the two horizon settings. Since we are mainly interested in the model dynamics here, these are left implicit for now, but H or γ will be appended to the tuple of each decision process later on.

Unless specified, in most of this thesis, the symbols $\mathcal{O}, \mathcal{A}, \mathcal{R}$ are always taken to be finite sets of observations, actions, and rewards, respectively. Moreover, we assume that the rewards are bounded as $\mathcal{R} \subset [0, 1]$. All results of this thesis can be adapted to general reward ranges, by linear scaling of the accuracy parameter ε.

NMDP A *Non-Markov Decision Process* (NMDP) (Hutter 2009; Brafman and De Giacomo 2019) is a tuple $\mathbf{N} \coloneqq \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$, where $\mathcal{O}$, $\mathcal{A}$, $\mathcal{R}$ are finite sets of observations, actions and rewards, $\bar{T} : \mathcal{H} \to \Delta(\mathcal{O})$ is the transition function and $\bar{R} : \mathcal{H} \to \Delta(\mathcal{R})$ is the reward function. The NMDP defines a decision process in which, for each time step t and history $h_t \in \mathcal{H}_t$,

$$\mathbb{P}(o_t, r_t \mid h_t, \mathbf{N}) = \bar{T}(o_t \mid h_t)\, \bar{R}(r_t \mid h_t) \tag{2.9}$$

In particular, the initial observation distribution, written as $\mu \in \Delta(\mathcal{O})$, is encoded as $\mu \coloneqq \bar{T}(h_0)$, computed from the empty history h_0. The NMDP is the most general formalisation of a decision process. Apparently, the only assumption appearing in eq. (2.9) is the conditional independence assumption between observations and rewards, written $o_t \perp r_t \mid h_t$. However, this is a minor simplification since r_t cannot be used to predict o_t, nor vice versa, because both have not yet been observed when selecting a_t.

k-MDP For $k \in \mathbb{N}_+$, a *k-Markov Decision Process* (k-MDP) is a tuple $\mathbf{M} \coloneqq \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, T, R \rangle$, where the transition and reward functions are $T : \mathcal{H}_k \to \Delta(\mathcal{O})$ and $R : \mathcal{H}_k \to \Delta(\mathcal{R})$, and its dynamics only depends on the last k pairs in the history, as $o_t \sim T(o_{t-k} a_{t-k+1} \dots o_{t-1} a_t)$ and $r_t \sim R(o_{t-k} a_{t-k+1} \dots o_{t-1} a_t)$. To represent the initial distribution, we will write designated start symbols $o_\circ$ and $a_\circ$ that cannot be used during normal operation. Then, we assume that $o_j = o_\circ$ and $a_{j+1} = a_\circ$, if $j < 0$. In particular $o_0 \sim \mu \coloneqq T(o_\circ a_\circ \dots o_\circ a_\circ)$.

A *Markov Decision Process* (MDP) is a 1-MDP. The observations of an MDP are also called *states*, and the notation $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, T, R \rangle$ is also common (Sutton and Andrew G. Barto 2018). Despite this simplified representation, MDPs play a central role in the RL literature. Under the finite-horizon setting, MDPs may also depend on the current time step, if this information is included in the observation. However, as in Puterman (1994), it is common to assume that this dependency is always explicitly present. Thus, in the finite-horizon setting, (k-)MDPs would be defined as $\mathbf{M} \coloneqq \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, T, R \rangle$, where $T \coloneqq \{T_t\}_{t \in [H]}$ and $R \coloneqq \{R_t\}_{t \in [H]}$ are collections of H transitions and reward functions, one for each time step. If all such functions are identical, we say that the MDP is stationary and we use the classic notation of $T : \mathcal{O}\mathcal{A} \to \Delta(\mathcal{O})$ and $R : \mathcal{O}\mathcal{A} \to \Delta(\mathcal{R})$. In the infinite-horizon setting, MDPs are always stationary.

RDP For defining RDPs, we first need regular functions. For some input alphabet Σ, a function $f : \Sigma^* \to \Omega$ is said to be *regular* if Ω is finite and, for each $\omega \in \Omega$, the set $f^{-1}(\omega) \coloneqq \{x \in \Sigma^* \mid f(x) = \omega\}$ is a regular language. A *Regular Decision Process* (RDP) (Brafman and De Giacomo 2019) is an NMDP $\langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$, in which the functions $\bar{T}$ and $\bar{R}$ are both regular. Despite the concise characterisation, RDPs have interesting properties. The main principles and definitions are given here. Then, an extended treatment of RDPs will be provided chapter 5.

RDPs have been recently proposed in Brafman and De Giacomo (2019), where they have been mainly described using temporal logics over finite traces, and mainly represented as finite state automata in some following works (Abadi and Brafman 2020; Ronca and De Giacomo 2021; Ronca, Licks, et al. 2022). The formulation we will use in this thesis is the natural extension of these automata-based representations for stochastic observations and rewards, and it is more closely related to the one appearing in Cipollone, Jonsson, et al. (2024). The existence of an automaton representation of RDPs comes from the fact that, if a function f is regular, then there exists a finite state transducer $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$, over some $\mathcal{Q}, \tau, \theta, q_0$, such that for each $x \in \Sigma^*$, $f(x) = \theta(\bar{\tau}(q_0, x))$. Since the functions $\bar{T}$ and $\bar{R}$ are both regular for RDPs, they can be associated with their respective automata, $\mathbf{A}_T$ and $\mathbf{A}_R$. Then, using classic arguments, we can take the synchronous product of $\mathbf{A}_T$ and $\mathbf{A}_R$ and obtain a single finite state transducer with a composite set of states $\mathcal{Q} \coloneqq \mathcal{Q}_T \times \mathcal{Q}_R$ and a composite output space $\Delta(\mathcal{O}) \times \Delta(\mathcal{R})$. In this thesis, we preferentially use this RDP representation in the form of a composite finite-state transducer.

In summary, an RDP is represented as a Moore machine $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$, where the input alphabet is $\Sigma \coloneqq \mathcal{O}\mathcal{A}$, and the output alphabet is some finite set of distribution pairs $\Omega \subset \Delta(\mathcal{O}) \times \Delta(\mathcal{R})$. Thus, if the RDP is in some state $q \in \mathcal{Q}$, the probability of the next observation and reward is given by $\theta(q)$ and the next RDP state becomes $q' = \tau(q, oa)$, where $o \in \mathcal{O}$ is the observation generated by the RDP and $a \in \mathcal{A}$ is the action selected by the agent. We write $\theta_{\mathsf{o}}(o \mid q)$ and $\theta_{\mathsf{r}}(r \mid q)$ for the individual probabilities and $\theta(or \mid q)$ for the joint ones. RDPs could also be defined as Mealy machines, in which we would write $o' \sim \theta_{\mathsf{o}}(q, oa)$ instead of $o' \sim \theta_{\mathsf{o}}(q)$. In fact, RDPs are properly characterised by the regularity of $\bar{T}$ and $\bar{R}$ and they are not tied to a single representation. Chapter 5 will provide further insight and examples for this decision process.

POMDP A *Partially Observable Markov Decision Process* (POMDP) is a classical decision process that generalizes MDPs with incomplete state information (Åström 1965). A POMDP is a tuple $\mathbf{P} \coloneqq \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}, T, R, O \rangle$, where $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, T, R \rangle$ form an MDP, $\mathcal{O}$ is a set of observations, and $O : \mathcal{S} \to \Delta(\mathcal{O})$ is the observation function.

Whenever a POMDP is in some state $s \in \mathcal{S}$, the agent can only observe $o \sim O(s)$, not s. As usual, histories and traces are composed of observations, not states. POMDPs are an intuitive but powerful extension of MDPs. A more in-depth description of POMDPs is given in section 5.2.

2.3 Planning in MDPs

This section summarises some important properties and the most classic planning algorithms for tabular MDPs. Although there are many similarities, due to some technical differences, the two horizon settings will be presented independently in sections 2.3.1 and 2.3.2.

2.3.1 Finite Horizon

As proven in Puterman (1994), in any MDP, there exists an optimal policy that is also Markovian and deterministic. Thus, in this thesis, when discussing MDPs, we will only consider the set of Markov policies Π_{m}. Now, for any MDP $\mathbf{M} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, T, R \rangle$ and policy $\pi \in \Pi_{\mathsf{m}}$, the value functions defined in eqs. (2.5) and (2.6) become $Q_{H+1} \equiv 0$ and, at any $t \in [H+1]$, $h_t = o_0 a_1 \dots o_{t-1} a_t \in \mathcal{H}_t$,

$$\begin{aligned}
Q_t^\pi(h_t) &= \mathbb{E}\left[g_t \mid \mathbf{M}, \pi, h_t\right] && (2.10)\\
&= \mathbb{E}\left[r_t \mid \mathbf{M}, \pi, h_t\right] + \mathbb{E}\left[g_{t+1} \mid \mathbf{M}, \pi, h_t\right] && (2.11)\\
&= \mathbb{E}\left[r_t \mid \mathbf{M}, \pi, h_t\right] + \mathbb{E}\left[V_t^\pi(h_t o_t) \mid \mathbf{M}, \pi, h_t\right] && (2.12)\\
&= \sum_{r \in \mathcal{R}} R_t(r \mid o_{t-1} a_t)\, r + \sum_{o \in \mathcal{O}} T_t(o \mid o_{t-1} a_t)\, V_t^\pi(o) && (2.13)\\
\text{and } V_t^\pi(h_t o) &= \sum_{a \in \mathcal{A}} \pi(a \mid h_t o)\, Q_{t+1}(h_t o a) && (2.14)
\end{aligned}$$

If, for induction hypothesis, $Q_{t+1}(h_t o a)$ is constant with respect to h_t, then we can see that the same is also true for $V_t(h_{t-1} o_t)$ and $Q_t(h_{t-1} o_t a_t)$. In other words, all value functions of Markov policies in MDPs depend only on the last observation and the following action. Thus, we overload the notation and write $V_t^\pi(o_t)$ and $Q_{t+1}^\pi(o_t, a_{t+1})$ in MDPs. For any finite-horizon decision process with rewards bounded in $[0, 1]$, the value of any policy π is bounded as $V_\mu^\pi \in [0, H]$, which is a commonly used fact.

Planning in a finite-horizon MDP $\mathbf{M}$ is the task of computing some optimal policy when $\mathbf{M}$ is known. As for other computations in finite horizons, planning can be solved with a linear backward-induction. In particular, computing the value of a policy requires a repeated application of eq. (2.13). On the other hand, the value of an optimal policy can be computed by a repeated application, for $t = H, \dots, 1$, of

$$\begin{aligned} Q_t^*(o,a) &= \sum_{r\in\mathcal{R}} R_t(r \mid o_{t-1}a_t)\, r + \sum_{o\in\mathcal{O}} T_t(o \mid o_{t-1}a_t)\, V_t^*(o) \\ \text{and } V_t^*(o) &= \max_{a\in\mathcal{A}} Q_{t+1}^*(o,a) \end{aligned} \tag{2.15}$$

In particular, the greedy Markov policy with respect to Q^*, that is, $\pi^* \coloneqq \{\pi_t\}_{t\in[H]}$, with $\pi_t(o) \coloneqq \arg\max_{a\in\mathcal{A}} Q_{t+1}^*(o,a)$, is an optimal policy (Puterman 1994). This covers the essentials of planning in finite-horizon MDPs.

2.3.2 Infinite Horizon

Planning In both chapters 3 and 4, the MDPs are treated under the infinite-horizon setting. This is also the most common setting in the RL literature and deserves a more complete preliminary section than its finite-horizon counterpart. Nevertheless, many results for finite horizons are still applicable for infinite horizons. Namely, we know that: the set of Markov policies Π_m always contains an optimal policy; the V- and Q- value functions only depend on the last observation and the following action, and can be written $V^\pi(o)$ and $Q^\pi(o,a)$, for any Markov policy π; finally, all values are bounded in $[0, 1/(1-\gamma)]$, since the rewards are in $[0,1]$ and due to the geometric sum with factor $\gamma < 1$. In addition, there exist optimal policies that are not only Markov but stationary.

Since there is no last instant, the same backward induction arguments for finite horizons cannot be used here. However, thanks to Bellman (1956), it is known that any function $Q : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ is equal to the optimal Q-function, Q^*, iff, for every o, a,

$$\begin{aligned} Q(o,a) &= \sum_{r\in\mathcal{R}} R(r \mid oa)\, r + \gamma \sum_{o\in\mathcal{O}} T(o \mid oa)\, V(o) \\ \text{with } V(o) &= \max_{a\in\mathcal{A}} Q(o,a) \end{aligned} \tag{2.16}$$

When treated as an assignment, eq. (2.16) can be applied repeatedly to compute a sequence of updated Q-functions $Q^{(0)}, Q^{(1)}$, ..., with their respective V-functions $V^{(0)}, V^{(1)}, \dots$. This is known as the Value Iteration (VI) algorithm (Bellman 1958). Since this assignment is a contraction mapping for Q, VI converges to the fixed point Q^*, in the limit. Moreover, terminating VI after a finite number of iterations returns near-optimal policies. In particular, if $k \geq -\log((1-\gamma)^2\varepsilon/2)/(1-\gamma)$, then $V^{(k)}(o) \geq V^*(o) - \varepsilon$, at all $o \in \mathcal{O}$ (Littman, Dean, et al. 1995; Agarwal, N. Jiang, et al. 2021). VI is a planning algorithm for MDPs. Another very effective planning algorithm is Policy Iteration (PI) (Howard 1960), that is simple to implement (Sutton and Andrew G. Barto 2018), and it also enjoys similar convergence properties to those of VI (Ye 2011). Note that both VI and PI depend on a factor of $1/(1-\gamma)$.

This is analogous to what happens to the backward induction algorithm for finite horizons, which depend on H. The effective horizon of $(1-\gamma)^{-1}$ will often appear when analysing planning and learning algorithms for MDPs.

As we can observe in eqs. (2.13), (2.15) and (2.16), the reward function always appears as $\sum_r R(r \mid oa)\, r$. In fact, when maximising the expected return, the value of a policy depends only on the expected rewards, not the specific reward distributions. For this reason, most planning and learning algorithms make the simplifying assumption that rewards are deterministic and equal to their expected value. So, in the remaining part of this section, in section 2.4 and throughout part II, we represent rewards with a deterministic function $R : \mathcal{O}\mathcal{A} \to [0,1]$ returning the immediate reward, not its distribution. Since MDP complexity is dominated by transition dynamics, most MDP algorithms can be extended to stochastic rewards with low overhead.

Occupancy Measures A group of important quantities for MDPs, which will be thoroughly used in chapter 4, are occupancy measures. The *state-action occupancy measure* under a policy $\pi \in \Pi$, is the discounted probability of reaching some state and action (o, a) when starting from some state o_p. Namely, it is defined as $d^\pi_{\mathsf{sa}} : \mathcal{O} \to \mathcal{O}\mathcal{A}$ with

$$d^\pi_{\mathsf{sa}}(oa \mid o_p) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(o_t = o, a_{t+1} = a \mid o_0 = o_p, \pi) \tag{2.17}$$

Marginalizing over the next action, we can also define the *state occupancy measure*

$$d^\pi_{\mathsf{s}}(o \mid o_p) \coloneqq \sum_{a \in \mathcal{A}} d^\pi_{\mathsf{sa}}(oa \mid o_p) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(o_t = o \mid o_0 = o_p, \pi) \tag{2.18}$$

Occupancy measures are important because the value function of any policy can be expressed as a simple linear combination between d^π_{sa} and the expected reward function. Namely, using the vector notation, this is

$$V^\pi(o) = \frac{\langle d^\pi_{\mathsf{sa}}(o), R\rangle}{1-\gamma} = \frac{1}{1-\gamma} \sum_{o'a'} d^\pi_{\mathsf{sa}}(o'a' \mid o)\, R(o'a') \tag{2.19}$$

and the value from the initial distribution is $V^\pi_\mu = \langle V^\pi, \mu \rangle$.

2.4 Learning in MDPs

After we covered the fundamental planning algorithms, we now focus on the associated learning problem. As for generic decision processes, Reinforcement Learning in MDPs is the problem of learning near-optimal policies from interaction. The interaction protocol has been described in the paragraph on page 15. In particular, the MDP and the agent interleave their outputs and the environment is periodically reset from the initial distribution μ. Although we mostly focus on infinite-horizon MDPs in this section, some of the following references also cover the finite-horizon case.

A classic RL algorithm for MDPs is Q-learning (Watkins and Dayan 1992). This is arguably the most famous RL algorithm, because of its simplicity. The algorithm acts on some stochastic exploration policy π^{b} to collect samples. The samples collected at each transition (o_{i-1}, a_i, r_i, o_i) are used only once to update the Q-function according to:

$$Q(o_{i-1}, a_i) \leftarrow (1 - \alpha_i)\, Q(o_{i-1}, a_i) + \alpha_i\, (r_i + \gamma \max_{a \in \mathcal{A}} Q(o_i, a)) \tag{2.20}$$

where $\alpha_0, \alpha_1, \dots$ is a sequence of learning rates that satisfy $\alpha_i \in [0, 1)$, $\sum_{i=0}^{\infty} \alpha = \infty$, and $\sum_{i=0}^{\infty} \alpha^2 < \infty$. Provided that the policy selects all actions infinitely often, that is, $\pi^{\mathsf{b}}(a \mid o) > 0$, then Q-learning converges to the optimal policy of the MDP, in the limit. Since Q-learning does not require that the policy being optimised be the one used to select actions, we say it is an *off-policy* algorithm. For a summary of the different variants of Q-learning and some *on-policy* algorithms, the reader might refer to Sutton and Andrew G. Barto (2018). Despite its simplicity, Q-learning with generic Q-function initializations and exploration policies could require an exponential number of interactions with respect to the number of MDP observations, even for simple tasks (Koenig and Simmons 1996). In fact, the main challenge posed by Reinforcement Learning in MDPs is the identification of an effective exploration strategy. This is commonly referred to as the “exploration–exploitation” trade-off, because algorithms need to carefully balance these two behaviours. In fact, while persistent exploration is inefficient, premature exploitation can lead to greedy and suboptimal policies.

The first RL algorithm to solve this trade-off and provide polynomial sample complexity guarantees was E^3 (Kearns and S. Singh 2002), which was later expanded to a more general and simple algorithm, called R-MAX (Brafman and Tennenholtz 2003). Unlike Q-learning, which is *model-free*, R-MAX is a *model-based* algorithm. This means that the algorithm internally estimates an approximate model of the environment. The core idea of R-MAX is the distinction between the sets of known and unknown states. Unknown states are those in which there is an action that has been tried an insufficient number of times. Since R-MAX assumes that unknown states are maximally rewarding, the policy obtained when planning is a compromise between the exploitation of the known region and exploration of the unknown part. This solution is an instantiation of the general principle called “optimism in the face of uncertainty” that has been implemented by many online learning algorithms. In a similar spirit to R-MAX, Delayed Q-learning implements the same principle in a model-free way. In particular, without estimating any transition function, Delayed Q-learning initialises the value functions to their maximum, then, it updates them using a finite number of samples. Both R-max and Delayed Q-learning are PAC for MDPs with polynomial sample complexity (Strehl, L. Li, et al. 2006).

The online RL algorithms with regret guarantees follow slightly different techniques, but they also implement the same optimism principle. Algorithms that directly target regret as their optimisation objective are Jaksch, Ortner, et al. (2010), Azar, Osband, et al. (2017), and Dann, Lattimore, et al. (2017).

Some lower bounds are also available. Regarding PAC lower-bounds, Dann and Brunskill (2015) show that any (ε, δ)-PAC RL algorithm for stationary finite-horizon MDPs, with ε and δ sufficiently small, must incur in an expected sample complexity of

$$\tilde{\Omega}\left(\frac{|\mathcal{O}\mathcal{A}|H^2}{\varepsilon^2}\log\left(\frac{c_1}{\delta + c_2}\right)\right) \tag{2.21}$$

steps, where $c_1, c_2 \in \mathbb{R}^+$ are appropriate constants.

All the previous results and algorithms apply to tabular MDPs. More generally, a decision process is tabular if its observation, state, and action spaces are finite, and they may be enumerated. This informal requirement implies that any algorithm for nontabular decision processes may not rely on counting arguments since each space may be too large to cover exhaustively. The most common approaches for non-tabular MDPs are Deep RL algorithms, which adopt Neural Networks (NN) as general function approximators of their value functions and/or policies. Notable Deep RL algorithms are DQN (Mnih, Kavukcuoglu, et al. 2015), DDQN (van Hasselt, Guez, et al. 2016), Dueling DQN (Z. Wang, Schaul, et al. 2016), Distributional DQN (Bellemare, Dabney, et al. 2017), Rainbow (Hessel, Modayil, et al. 2018), with respect to approaches based on Q-learning, and TRPO (Schulman, Levine, et al. 2015), PPO (Schulman, Wolski, et al. 2017), DDPG (Lillicrap, Hunt, et al. 2016), SAC (Haarnoja, Zhou, et al. 2018), regarding policy gradient and actor-critic methods. These algorithms are only some of the most effective RL methods that have been validated with extensive experimentation. Most of them are heavily inspired by the RL theory. However, since their aim is to be usable in practice, due to their approximations and the use of Deep NNs, they do not provide explicit theoretical guarantees.

Another long line of work for nontabular MDPs is focused on obtaining sample efficiency guarantees under a pre-learnt feature network (L. Yang and M. Wang 2019; C. Jin, Z. Yang, et al. 2020). Such algorithms usually provide strong formal guarantees at the cost of increased computational complexity or additional technical complexity in their practical implementations.

Part II

Learning With MDP Abstractions

Introduction to part II

As anticipated, this thesis explores two aspects of learning in complex environments: learning in non-Markovian decision processes and learning with MDP abstractions. Since the classes of environments considered in the two cases differ, the background and techniques are partly specific to each of the two problems. However, as we will see, both directions share one common need: respecting the Markov assumptions on the constructed state space. This is something to achieve, in the former case, and to preserve, so as to avoid nonstationarity, in the latter. In this part of the thesis, we explore this second direction.

Hierarchical Reinforcement Learning (HRL) is the large subfield of RL that studies abstractions of decision processes, and how they can be used to improve the efficiency and the compositionality of RL algorithms. As in other research fields, HRL is only a loose grouping of a series of works from the literature. The main ambiguity in defining it comes from the fact that there is no universally shared notion of what exactly an MDP abstraction is. In fact, defining abstractions is even one of the main objectives of HRL. Rather, HRL is best described by the objectives it aims to achieve. As well summarised by Abel (2020), abstractions have the following desiderata:

- Efficient decision-making: planning or learning with a good abstraction should be much faster than planning or learning without it;
- Near-optimality: an abstraction should enable agents to discover policies that solve the original problem to a satisfactory degree.

However, we should note that the third desiderata reported in Abel (2020) has not been shared here. Namely, this thesis does not require that abstractions should be efficient to learn from experience. In fact, whether it is feasible to efficiently learn an abstraction and effectively use it in the same learning routine still remains an open research question, which might not admit a positive answer, outside continual RL or multi-task RL.

This thesis defines MDP abstractions that satisfy the two properties above, also improving on the definitions that can be found in the literature. As a whole, the purpose of part II is to answer two important questions for Reinforcement Learning: (i) What should be regarded as "good" MDP abstractions? (ii) How can abstractions be used to improve the sample efficiency and compositionality of RL algorithms? The two chapters of part II, 3 and 4, both address these two questions and provide incremental results on the topic.

Chapter 3

Exploiting MDP Abstractions

The content of this chapter is based on the work: Roberto Cipollone, Giuseppe De Giacomo, Marco Favorito, Luca Iocchi, and Fabio Patrizi (2023a). "Exploiting Multiple Abstractions in Episodic RL via Reward Shaping". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 37, pp. 7227–7234.

3.1 Introduction

Among the many notions of MDP abstractions that can be found in the literature, this work assumes that an abstraction essentially consists of a separate decision process, with its own dynamics, which acts as an abstract representation of the original domain. As we will see in the related work section at page 31, many works struggle in defining explicit abstractions that involve both the states and the actions, especially without incurring in inapplicable constraints or undesirable non-stationarity effects. This chapter proposes a simple technique that does not incur in these issues. The purpose is to define very flexible abstractions while improving in sample complexity and retaining training stability towards the near-optimal policy.

Consider a generic MDP that we aim to solve efficiently. As in L. Li, Walsh, et al. (2006), we refer to this decision process as *ground MDP*. This might be a domain with a large or continuous state–action space, possibly with a complex dynamics. This work considers a linear hierarchy of abstractions of the ground MDP underlying the target domain. Each layer in this hierarchy is a simplified model, still represented as an MDP, of the one immediately below. A simple example is that of an agent moving in a map. The states of the ground MDP may capture the real pose of the agent, such as continuous coordinates, orientation, and other features. The states of its abstraction, instead, may provide a coarser description, obtained by coordinate discretization, semantic map labelling (that is, associating semantic labels to metric poses), or by projecting out state variables. Ultimately, such a compression corresponds to partitioning the ground state space into abstract states, implicitly defining a mapping from the former to the latter. The actions of the two models can also differ, in general, because they would include the actions that are most appropriate for each representation. In fact, a single abstract action will generally correspond to a sequence of actions in the ground domain. Thus, with reference to the related work section on page 31, the approach proposed in this chapter is best regarded as a state-action abstraction.

This work assumes that the abstractions are available in the form of one or more MDPs simulators. Simulators are commonly used in RL and robotics. Often, simply through a different configuration of the same software, such as noiseless or ideal movement actions, it is possible to obtain a simplified environment which acts as an abstraction. Importantly, each simplified model also applies to a variety of tasks that may be defined over the same domain.

Taking advantage of the abstraction hierarchy, we devise an approach that allows any off-policy RL algorithm to efficiently explore the ground environment while guaranteeing optimal convergence. The core intuition is that the value function of the abstract MDP is a good proxy for the expected return of each group of ground states. This work proposes a variant of Reward Shaping (RS), whose potential is generated from the abstract value function. In this way, when learning in the ground MDP, the agent is *biased* to first visit the states that correspond in the abstraction to those that are preferred by the abstract policy, thus trying, in a sense, to replicate its behaviour in the ground domain.

In order to characterise the effectiveness of the exploration bias, it is essential that the transitions of the abstraction are good proxies for the dynamics of the ground MDP. This relationship will be characterised by the identification of conditions under which the abstraction induces an exploration policy that is consistent with the ground optimal policy. To obtain this result, in theorem 3.4, we develop a general result for comparing ground MDP policies in the presence of state partitions and abstractions.

The chapter, together with the individual contributions, is summarised below.

- In section 3.4, we define an original Reward Shaping schema that allows for transferring the acquired experience from coarser models to the more concrete domains in the abstraction hierarchy. The algorithm is simple and can be combined with generic offline RL algorithms for MDPs. Importantly, the link between abstract and ground actions remains implicit. Although we choose to provide a persistent bias to exploration, the algorithm guarantees convergence towards the original policy.

- In section 3.5, we derive a relation between abstractions and ground MDPs, obtained through the induced exploration policy in the lower domain. To derive this result, we prove a more general statement in theorem 3.4, which can be used to evaluate the suboptimality of generic policies in the presence of state partitions. This bound strictly improves over analogous results in the literature.
- To obtain this result, we identify two new parameters that characterise the quality of an abstraction with respect to the ground MDP. In particular, "abstract value approximation", arises as a new condition to evaluate when abstract states can be good representatives of the ground MDP values.
- In section 3.6, we conclude the chapter showing that our approach has a positive impact on sample efficiency and that modelling errors yield only a limited degradation in performance.

3.2 Hierarchical Reinforcement Learning

After the quick introduction to part II, which briefly introduces the main topics of Hierarchical RL, we now need to describe more in detail what techniques and methods are commonly used in the literature, in order to better understand how the approaches described in this thesis relate to other works. This context will also be useful for chapter 4. In this thesis, we only consider abstractions of MDPs, which is by far the most common case in the literature.

Motivation We first want to draw the reader's attention to the core motivation for this research field. In other words, why should we study the role of MDP abstractions in RL? After all, section 2.4 already anticipated some sample-efficient RL algorithm for MDPs with near-optimality guarantees. HRL has been heavily influenced by hierarchical methods for classical planning, and together, they share the same core intuitions and motivations. In fact, HRL can be seen as a generalisation of hierarchical planning to stochastic decision processes. For historical references on hierarchical planning, the reader might refer to Russell and Norvig (2009). The common intuition can be explained with the following example.

Suppose that an agent starts from a specific room of a house and its task is to open the door in the entrance corridor. This objective requires the agent to learn and plan for navigating toward the corridor and, only then, find a policy that allows to grab the handle and pull. For an autonomous agent, this behaviour clearly requires precise control of the force applied by each motor to achieve an accurate manipulation of the door handle. However, the same level of modelling granularity is not reasonable for navigating large distances. In the same way, a person would first navigate through the house, planning at an abstract level, reasoning in terms of rooms and changes between them, while individual muscle inputs would not matter. On the other hand, when approaching the specific manipulation task, reasoning must take place at a much more granular level (Tenenbaum, Kemp, et al. 2011; Eckstein and Anne G. E. Collins 2020).

In the same way, HRL usually considers at least two levels: a detailed description of the environment dynamics, which is represented by the "ground" MDP, and a more coarse representation, the abstraction. The impact of such an abstraction depends on the specific technique, but, generally, MDP abstractions allow for three important benefits: efficiency, policy reuse, and interpretability. In fact, as for the example above, efficient planning and learning consists of reasoning in a compositional way, avoiding a multitude of small-scale decisions. Compositionality is also a step toward policy reuse, which is a very relevant topic for RL, due to its specificity toward the reward function used for learning. Finally, high-level descriptions are much more interpretable to a human observer. Simply because "the agent is in room number 3" and "the agent is trying to open the door" are much easier to understand than "the agent configuration is $(x, y, \theta, \dots)$" and the action is "$(0.3, 0, 0.1, \dots)$".

Related Work The following paragraphs summarise the main achievements in the Hierarchical RL literature. Some of these results and techniques will also be studied in greater detail in the next sections, whenever they are needed as preliminaries for the techniques proposed in this thesis.

One of the first works in HRL is Feudal Learning from (Dayan and Hinton 1992). Although informally, this work has set the fundamental concepts of subtask decomposition, high-level learning, and multiple levels of abstractions. They considered a discrete navigation scenario and a linear abstraction hierarchy, where each level in the hierarchy is optimised with Q-learning, while ignoring the details of the one below. In a later work, Ravindran and Andrew G. Barto (2002) proposed that abstractions should also be MDPs and these should be related to the ground MDP via homomorphisms. This idea has also been extended to approximate homomorphisms in Ravindran and Andrew G Barto (2004). As we will see, homomorphisms are excellent for capturing symmetries in the ground MDP, and eliminate those regularities by representing them only once in the abstraction. An influential work in this line of research is L. Li, Walsh, et al. (2006), which analysed some state abstraction formalisms that could be found in the literature. In particular, they considered MDP compressions that preserved either the dynamics, the values, or the optimal actions, and they showed the guarantees that could be derived with each formalism. Moreover, their simple framework for presenting state abstractions was later used by

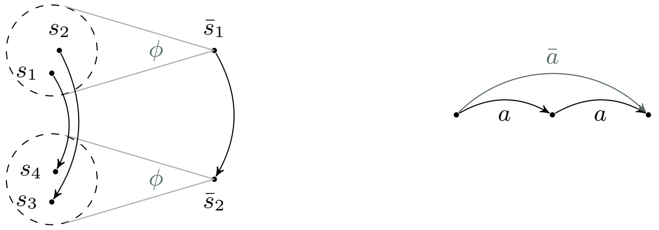


Figure 3.1. State abstraction (left). Multiple states in $\mathcal{S}$ map to one state in the abstract $\bar{\mathcal{S}}$. The transitions are symmetric at both levels. Action abstraction (right). The state space remains the same, but multiple actions in $\mathcal{A}$ are related to a single abstract action in $\bar{\mathcal{A}}$.

other papers (Abel, Umbanhowar, et al. 2020), and will also be adopted in this thesis.

These early approaches are mostly considered *state abstractions*. In other words, if $\mathcal{S}$ is the state space of the ground MDP[1] and $\bar{\mathcal{S}}$ is a set representing an abstract state space, these works assume that there exists a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ which preserves some important property of the ground MDP. In these early works, the property to preserve was often related to some immediate behaviour, as in homomorphisms. The intuition behind state-abstraction mappings is illustrated in fig. 3.1 (left).

In parallel to state abstractions, a second line of research developed an alternative view of MDP abstractions that we generally consider *action abstractions*. Unlike state abstractions, these place a major emphasis on defining what is a high-level action. The theory of *options* is a very influential framework in this topic. Some papers, such as Precup and Sutton (1997), Sutton, Precup, and S. P. Singh (1998), and Sutton, Precup, and S. Singh (1999), successfully formalised a first notion of high-level actions and behaviours, showing that these can be optimised with a Q-learning style of update. Options will be formally defined in section 3.3, as this thesis makes extensive use of options to define abstract actions. For now, they can be regarded as partial policies with a termination condition. Clearly, depending on which options are available to the learning agent, the utility of options to speed up the learning process can change drastically (Jong, Hester, et al. 2008). The options framework has been very influential in the field of HRL, and they have been extended in multiple ways. Their internal policy can be learnt from experience with policy gradient methods (Levy and Shimkin 2011), as well as their termination condition (Bacon, Harb, et al. 2017), or the initiation sets (Khetarpal, Klissarov, et al. 2020). This means that learning options is reasonably effective, even in large or continuous state spaces, similarly to what happens when learning classic policies in MDPs.

Options can be very effective in reducing the effective planning horizon that is needed to solve composite tasks. However, these techniques may not generally

[1] Throughout part II, $\mathcal{S}$ is used to represent the state/observation space of k-MDPs. Since these chapters do not treat partial observability, this is done to better match the notation used in the literature.

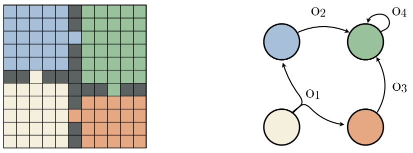


Figure 3.2. Intuition of state partitioning and ϕ-relative options. Graphic representation of ground MDP (left) and abstraction (right). Figure from Abel, Umbanhowar, et al. (2020).

correspond to a strong decomposition of the original MDP into subtasks because the state space is not compressed. For this reason, more recent works study how abstractions can involve both the state and the action spaces (Abel, Umbanhowar, et al. 2020; Abel 2020). Here, abstract actions are interpreted as a specific class of options that terminate only when the abstract state changes. These will be called ϕ-relative options, and their intuition is depicted in fig. 3.2.

Other excellent papers do not clearly fit this ternary classification of related work. Some of these papers are: G. D. Konidaris, Kaelbling, et al. (2018) and J. Lee, Katz, et al. (2022), using classical planning domains as abstractions for RL; Jong and Stone (2008), combining R-max with MAXQ (Dietterich 2000), a classical HRL algorithm; Ravindran and Andrew G. Barto (2003) that aimed to combine options and homomorphisms; Infante, Jonsson, et al. (2022) that develop HRL for "Linear MDPs"; García, Visús, et al. (2022) that study how different MDPs can be related via various metrics;

The biggest challenges of HRL are related to the following important question. How can multiple states and multiple actions be related to one abstract state and action in the abstraction? Somehow, being in a state in which a transition or reward is very probable should be modelled very differently from other neighbouring states with different probabilities. The grouping of states may be easy to decide for simple domains with bottlenecks such as fig. 3.2 (although it remains complex with respect to actions). However, in more general cases, the grouping provided by the abstraction might introduce undesired discrepancies. This modelling issue could be very effectively modelled with POMDPs, according to the observation function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$. However, this may render the problem intractable. So, most authors in HRL prefer to regard the lack of Markovianity in the abstraction as a generic nonstationarity (Gürtler, Büchler, et al. 2021; Jothimurugan, Bastani, et al. 2021). Nevertheless, as a result of this choice, stability and optimality guarantees might be lost in the general case.

An excellent work in HRL is Wen, Precup, et al. (2020). Their main contribution is to provide an algorithm with a formal efficiency guarantee, in the form of a Bayesian regret, which depends on some features often encountered in HRL, such as the maximum cardinality of the sets in the partition. Another feature they find significant is a measure of the quality of the "exit profiles", which quantifies the extent and value of the boundaries between abstract states. This thesis also studies the impact of "exit states" in chapter 4, and improves these formalizations.

We conclude this related work section by reviewing the papers that focus on learning abstract representations. As for the papers presented up to this point, the abstraction of an MDP is interpreted quite differently by each paper. So, in general, they do not have the same learning objective. The intuition that learning in HRL should target subtasks and bottleneck states was shared since the early works (Simsek and Andrew G. Barto 2004). Based on these ideas, options discovery has been at the core of learning abstractions ever since (P. S. Castro and Precup 2011; Machado, Bellemare, et al. 2017). However, the target for learning such options might also differ, as they may try to address cover time (Jinnai, Park, Abel, et al. 2019), goal states (Nachum, Gu, et al. 2019), state space covering (Jinnai, Park, Machado, et al. 2020; S. Lee, Kim, et al. 2022), and information measures (Y. Jiang, E. Z. Liu, et al. 2022). Finally, instead of options, some works directly optimise over different state partitions (Biza and Jr. 2019; Steccanella, Totaro, et al. 2021; Steccanella 2023).

3.3 Preliminaries and Formulation

As in the other chapters, all the global conventions that have been set in section 2.1 are also valid here. The additional notation and background that is needed is discussed here below. The notation used in this chapter is for finite sets and distributions because, for simplicity, we use sums instead of integrals, and we avoid measure-theoretic quantities. However, the approach is not limited to finite MDPs. As we shall see, only the abstraction must be represented by a finite MDP, but no such limitation is required on the ground MDP.

As anticipated in section 2.2, the last observation of any MDP is a complete description of the state of the environment, since it satisfies the Markov properties. Therefore, observations are usually called states and the notation $\mathcal{S}$ is very common, instead of $\mathcal{O}$. This convention will be followed in both chapters 3 and 4. Regarding rewards, instead, this chapter allows them to be stochastic in the last state and action, under the condition that rewards are deterministic if the next state is also given. Thus, as a recap of the global definitions and these specific choices, in this chapter, an MDP is intended as $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, with $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, and $\gamma \in (0, 1)$. Since rewards are deterministic and contained in $[0, 1]$, the set of output rewards $\mathcal{R}$ has been omitted from the MDP tuple.

With respect to notation, anything related to the abstraction is written with a top bar. So, if $\mathcal{S}$ is the ground state space, $\bar{\mathcal{S}}$ would refer to some abstract state space defined previously. Similarly to L. Li, Walsh, et al. (2006), we will use $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ for the state-abstraction function, that we simply call *mapping function.* For any function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, we use $\lfloor \bar{x} \rfloor_\phi$ as an abbreviation for the set of ground states that map to $\bar{x}$. That is, for $\bar{x} \in \bar{\mathcal{S}}$, $\lfloor \bar{x} \rfloor_\phi \coloneqq \{x \in \mathcal{S} \mid \phi(x) = \bar{x}\}$. Since the mapping function will often be clear from the context, we will also use $\lfloor \bar{x} \rfloor$. So, any ϕ immediately induces a partition of the ground state space, $\{\lfloor \bar{s} \rfloor\}_{\bar{s} \in \bar{\mathcal{S}}}$. We will informally refer to each set in this partition as a *block* of states. In the following, we will always assume that ϕ is surjective.

The subscript notation $s_{i:j}$, for integers i, j, is an abbreviation of $s_i, s_{i+1}, \dots, s_j$, if $i \leq j$, or the empty sequence, if $i > j$. Similarly, $\phi(s_{i:j})$ represents $\phi(s_i) \dots \phi(s_j)$, if $i \leq j$, or the empty sequence, otherwise.

Options (Sutton, Precup, and S. P. Singh 1998) An *option* for an MDP $\mathbf{M}$, is a temporally extended action, defined as $o = \langle \mathcal{I}_o, \pi_o, \beta_o \rangle$, where $\mathcal{I}_o \subseteq \mathcal{S}$ is an initiation set, $\pi_o \in \Pi$ is the policy that o executes, and $\beta_o : \mathcal{S} \to \{0, 1\}$ is a termination condition. With respect to Sutton, Precup, and S. Singh (1999), we take the terminating condition to be stationary and deterministic. Relaxing the usual notation, $Q^\pi(s, a)$, we write $Q^\pi(s, o)$, for the expected return of executing the option o until termination, then following policy π afterwards. We will use the letter Ω to denote sets of options and write $o \in \Omega$. There should be no confusion between the letter o and observations, since in MDPs the observations are represented as states $s \in \mathcal{S}$.

ϕ-Relative Options (Abel, Umbanhowar, et al. 2020) Given an MDP $\mathbf{M}$ and a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, an option $o = \langle \mathcal{I}_o, \pi_o, \beta_o \rangle$ of $\mathbf{M}$ is said to be *ϕ-relative* if there exists some $\bar{s} \in \bar{\mathcal{S}}$ such that

$$\mathcal{I}_o = \lfloor \bar{s} \rfloor_\phi, \qquad \beta_o(s) = \mathbb{I}(s \notin \lfloor \bar{s} \rfloor_\phi), \qquad \pi_o \in \Pi_{\bar{s}} \tag{3.1}$$

where $\Pi_{\bar{s}} : \lfloor \bar{s} \rfloor_\phi \to \Delta(\mathcal{A})$ is the set of partial policies defined for the relevant block. Some set of options Ω is ϕ-relative iff all of its options are. In the following, we only consider ϕ-relative options $\Omega = \cup_{\bar{s} \in \bar{\mathcal{S}}} \Omega_{\bar{s}}$, where $\Omega_{\bar{s}}$ is the set of options that satisfies eq. (3.1) with respect to $\bar{s}$. Any high-level deterministic policy on ϕ-relative options $\bar{\mathcal{S}} \to \Omega$ corresponds to a unique subset $\Omega' \subseteq \Omega$, which contains one option per block. We call Ω' a *policy of options*, as it can be fully treated as a policy. For example, $V^{\Omega'}$ is the value of the policy that always executes the only applicable option in Ω' until termination.

Reward Shaping *Reward Shaping* (RS) is a technique for learning in MDPs with sparse rewards, which rarely occur during exploration. The purpose of RS is to guide the agent by exploiting some prior knowledge in the form of additional rewards. Namely, the original reward function R is replaced by $R^{\mathrm{s}}(s,a,s') \coloneqq R(s,a,s') + F(s,a,s')$, where F is some *shaping* function. In the most classic approach, called *potential-based RS* (A. Y. Ng, Harada, et al. 1999), the shaping function for an MDP $\mathbf{M}$ is defined in terms of a potential function, $\Phi : \mathcal{S} \to \mathbb{R}$, as:

$$F(s,a,s') \coloneqq \gamma\,\Phi(s') - \Phi(s) \tag{3.2}$$

From now on we assume that Reward Shaping is always potential-based. If an infinite horizon is considered, this definition and its variants (Eric Wiewiora, Cottrell, et al. 2003; Devlin and Kudenko 2012) guarantee that the set of optimal policies of $\mathbf{M}$ and $\mathbf{M}^{\mathrm{s}} \coloneqq \langle \mathcal{S}, \mathcal{A}, T, R^{\mathrm{s}}, \gamma \rangle$ coincide. In fact, as shown by E. Wiewiora (2003), the Q-learning algorithm executed on $\mathbf{M}^{\mathrm{s}}$ performs the same updates as Q-learning on $\mathbf{M}$, with this modified initialisation: $Q'_0(s,a) \coloneqq Q_0(s,a) + \Phi(s)$.

Problem Formulation

Consider an environment in which experience is costly to obtain. This might be a complex simulation or an actual environment in which a physical robot is acting. This is our *ground* MDP $\mathbf{M}_0$ that we aim to solve while reducing the number of interactions with the environment. Instead of learning on this MDP directly, we choose to solve a simplified, related problem that we call the *abstract* MDP. This idea is not limited to a single abstraction. In fact, we consider a linear hierarchy of related MDPs $\mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_n$, with decreasing difficulty, where the experience acquired by an expert acting in $\mathbf{M}_i$ can be exploited to accelerate learning in the previous one, $\mathbf{M}_{i-1}$.

Associated to each MDP abstraction $\mathbf{M}_i$, we also assume the existence of a *mapping* function $\phi_i : \mathcal{S}_i \to \mathcal{S}_{i+1}$, which projects states of $\mathbf{M}_i$ to states of its direct abstraction $\mathbf{M}_{i+1}$. This induces a partition of $\mathcal{S}_i$, where each block contains all states that are mapped through ϕ_i to a single state in $\mathbf{M}_{i+1}$. The existence of state mappings is a classic assumption in Hierarchical RL (Ravindran and Andrew G. Barto 2002; Abel, Hershkowitz, et al. 2016; Abel, Umbanhowar, et al. 2020). Unlike other works, instead, we do not require any mapping between the ground and the abstract actions. This relationship will remain implicit. This feature leaves a great flexibility to designers in defining the abstraction hierarchy.

An abstract model is a suitable relaxation of the environment dynamics. For example, in a navigation scenario, the ground MDP might model the little stochastic effects of low-level controls. An abstraction, instead, could contain actions that allow

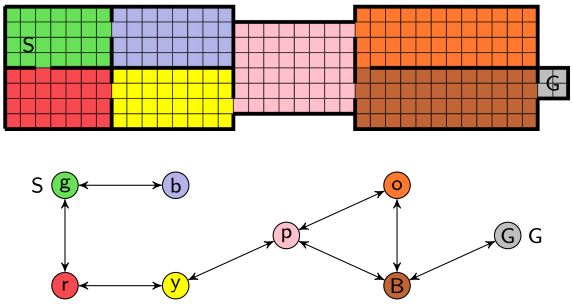


Figure 3.3. A grid world domain (top) and an abstraction (bottom). The colours encode the mapping function, G is the goal.

to just “leave the room”. In section 3.5, we formalise this intuition by deriving a measure that quantifies the accuracy of an abstraction with respect to the lower domain. However, as an even simpler example of abstractions, consider the grid world domain, the abstract MDP, and the mapping in fig. 3.3. Thanks to the abstraction, we can inform the agent that exploration should avoid the blue “room” (b) and only learn options for moving to and within the other rooms. Note that the same does not necessarily hold for the orange block (o), instead, since the optimal path depends on the specific transition probabilities in each arc.

3.4 Exploiting Abstractions With Reward Shaping

This section presents our contribution, that is, a method to exploit abstract models by using a specific form of reward shaping, in the context of episodic RL. Consider a hierarchy of abstractions $\mathbf{M}_0, \dots, \mathbf{M}_n$, together with the functions $\phi_0, \dots, \phi_{n-1}$. So, the abstraction of $\mathbf{M}_i$ is $\langle \mathbf{M}_{i+1}, \phi_i \rangle$.

The learning process proceeds incrementally, training in order from the easiest to the hardest model. When learning in $\mathbf{M}_i$, our method takes advantage of the knowledge acquired from its abstraction by applying a form of reward shaping, constructed from the estimated solution for $\mathbf{M}_{i+1}$. In particular, we recognise that the optimal value function V^*_{i+1} of $\mathbf{M}_{i+1}$ is a helpful heuristic that can be used to evaluate how desirable a group of states is according to the abstraction. We formalise our application of RS in the following definition.

Definition 3.1. Let $\mathbf{M}_i$ be an MDP and $\langle \mathbf{M}_{i+1}, \phi_i \rangle$ its abstraction. We define the *biased MDP* of $\mathbf{M}_i$ with respect to $\langle \mathbf{M}_{i+1}, \phi_i \rangle$ as the MDP $\mathbf{M}^{\mathsf{b}}_i$, resulting from the application of reward shaping to $\mathbf{M}_i$, using the potential:

$$\Phi(s) \coloneqq V^*_{i+1}(\phi_i(s)) \tag{3.3}$$

where V^*_{i+1} is the optimal value function of $\mathbf{M}_{i+1}$.

This choice allows evaluating each state according to how much “desirable” the corresponding abstract state is, according to the optimal policy for $\mathbf{M}_{i+1}$. This is beneficial, as high potentials are associated with high initializations of the Q-function (E. Wiewiora 2003). In fact, already A. Y. Ng, Harada, et al. (1999) was the first to notice that the MDP’s own optimal value function is a very natural potential for reward shaping. We extend this idea by using the value function of the abstract MDP instead.

3.4.1 Reward Shaping for Episodic RL

Potential-based RS has been explicitly designed not to alter optimal policies. In fact, regardless of potential, in the infinite-horizon setting, or if the episodes always terminate in an absorbing state with zero potential, this is always guaranteed (A. Y. Ng, Harada, et al. 1999). However, in RL, it is extremely common to diversify the agent’s experiences by breaking up exploration into episodes of finite length. This means that we are still optimising for values computed over infinite horizons, but learning is carried out in an episodic manner. As a consequence, these guarantees do not hold anymore, as episodes might end in states with arbitrary potential and the optimal policy will be altered (Grzes 2017).

To see this, given an MDP $\mathbf{M}$, consider the trace of an episode $r_0 s_0 a_1 \dots r_n s_n a_{n+1}$ and the associated trace $r'_0 s_0 a_1 \dots r'_n s_n a_{n+1}$, where rewards r'_i have been modified via reward shaping. Let g and g' be the discounted cumulative returns in the two traces, respectively. Then, these are related as follows (Grzes 2017):

$$g' \coloneqq \sum_{t=1}^{n} \gamma^{n-1} \, r'_t = g + \gamma^n \, \Phi(s_n) - \Phi(s_0) \tag{3.4}$$

because all intermediated potentials cancel out in the telescopic sum. Now, since $\Phi(s_0)$ is only a constant shift, the term $\gamma^n \, \Phi(s_n)$ is the one responsible for modifying the optimal policies, as it depends on the state reached at the end of the episode. The solution proposed by Grzes (2017) for this problem is to assume the null potential for every terminal state. Concretely, this means to set $\Phi(s_n) = 0$, each time an episode is interrupted. We call this RS technique *return-invariant*, as this would, in fact, preserve the total returns and the original optimal policies. However, this is not always the only desirable solution. In fact, we might be interested in relaxing the convergence guarantee to an identical policy in favour of a stronger impact on learning speed. The same need has also been identified by Schubert, Oguz, et al. (2021).

As an example, let us consider an MDP with a null reward function everywhere, except when transitioning to a distinct goal state. As a consequence of eq. (3.4) and the choice $\Phi(s_n) = 0$, all finite trajectories that do not contain the goal state are associated to the same return, regardless of how close they arrive to the goal state.

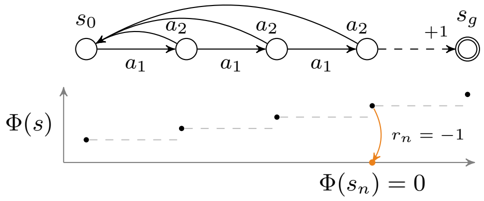


Figure 3.4. With the solution of Grzes (2017), any trajectory that does not reach the goal receives a final reward that cancels out the accumulated return.

Moreover, this is achieved by a drop in potential, causing a negative reward to be generated for that trajectory. This is illustrated in fig. 3.4. Since the agent cannot estimate its distance to the goal through differences in return, return-invariant RS of Grzes (2017) does not provide a persistent exploration bias to the agent. The form of reward shaping adopted in this work, which is formulated in definition 3.1, does not assign null potentials to terminal states. Therefore, we say that it is *non return-invariant*. This explains why the MDP of definition 3.1 has been called "biased": optimal policies of $\mathbf{M}_i^{\mathsf{b}}$ and $\mathbf{M}_i$ do not necessarily correspond. This is addressed in the next section, where we show that the complete procedure recovers optimal convergence.

3.4.2 The Algorithm

Since we deliberately adopted a form of RS which is not return invariant in the episodic setting, we devised a technique to recover optimality. The present section illustrates the proposed method and proves that it converges to the optimal policy of the original MDP, when coupled with any off-policy algorithm.

The procedure is presented in detail in algorithm algorithm 3.1. Learning proceeds sequentially, training from the most abstract model to the ground domain. When learning in the i-th MDP, the estimated optimal value function $\hat{V}_{i+1}^*$ of the previous model is used to obtain a reward shaping function (line 4). Experience is collected by sampling actions according to a stochastic exploration policy, as determined by the specific learning algorithm $\mathfrak{A}$. This policy may be derived from the current optimal policy estimate for $\mathbf{M}_i^{\mathsf{b}}$, such as an ϵ-greedy exploration policy in $\hat{Q}_i^{\mathsf{b}*}$ (line 9). For completeness, we recall that an ϵ-greedy policy with respect to some Q is uniform with probability ϵ, and it returns the greedy action, $\arg\max_a Q(s,a)$, with probability $1-\epsilon$. Being ϵ-greedy with respect to $\hat{Q}_i^{\mathsf{b}*}$ means using the biased value function for action selection and exploration. Finally, the output of each learning phase is the estimate $\hat{\pi}_i^*$ and $\hat{V}_i^*$ for the original MDP $\mathbf{M}_i$. This allows iterating the process with an unbiased value estimate, or closing the procedure with the final learning objective $\hat{\pi}_0^*$.

Algorithm 3.1: Main algorithm

```
Input: Off-policy RL algorithm $\mathfrak{A}$
Input: $\mathbf{M}_0, \dots, \mathbf{M}_n$, $\phi_0, \dots, \phi_{n-1}$
Output: $\hat{\pi}_0^*$, ground MDP estimated policy
$\hat{V}_{n+1}^* : s \mapsto 0$
$\phi_n : s \mapsto s$
foreach $i \in \{n, \dots, 0\}$ do
    $F_i \leftarrow \textsc{Shaping}(\gamma_i, \phi_i, \hat{V}_{i+1}^*)$
    $Learner_i \leftarrow \mathfrak{A}(\mathbf{M}_i)$
    $Learner_i^{\mathsf{b}} \leftarrow \mathfrak{A}(\mathbf{M}_i)$
    while not $Learner_i.\textsc{Stop}()$ do
        $s \leftarrow \mathbf{M}_i.\textsc{State}()$
        $a \leftarrow Learner_i^{\mathsf{b}}.\textsc{Action}(s)$
        $r, s' \leftarrow \mathbf{M}_i.\textsc{Act}(a)$
        $r^b \leftarrow r + F_i(s, a, s')$
        $Learner_i^{\mathsf{b}}.\textsc{Update}(s, a, r^b, s')$
        $Learner_i.\textsc{Update}(s, a, r, s')$
    end
    $\hat{\pi}_i^* \leftarrow Learner_i.\textsc{Output}()$
    $\hat{V}_i^* \leftarrow \textsc{ComputeValue}(\hat{\pi}_i^*, \mathbf{M}_i)$
end
```

An RL algorithm for MDPs, $\mathfrak{A}$, is off-policy if, for any MDP, the sequence of policy updates it generates always converges to the optimal policy, as long as the transitions are produced with a sequence of policies $\{\pi_l\}_{l \in \mathbb{N}}$ satisfying $\pi_l(a \mid s) > 0$, at every s, a, and global learning time $l \in \mathbb{N}$. As a consequence of off-policy learning, algorithm 3.1 converges to the optimal policy, as stated below.

Proposition 3.1. *Consider MDPs $\mathbf{M}_0, \dots, \mathbf{M}_n$ and their associated mapping functions $\phi_0, \dots, \phi_{n-1}$. If $\mathfrak{A}$ is an off-policy RL algorithm, then, in every i-th iteration of algorithm 3.1, $\hat{\pi}_i^*$ converges to π_i^*, as the number of environment interactions increases.*

Proof. See page 52. □

3.5 Abstraction Quality

Our approach gives great flexibility in selecting an abstraction. Still, given some ground MDP, not all models are equally helpful and beneficial, when used for reward shaping or selected as abstractions. This section serves to define what properties good abstractions should possess. As we can see from algorithm 3.1, they are used to construct effective exploration policies (in row 9). Therefore, each abstraction should induce a biased MDP that assigns higher rewards to regions of the state space from which the optimal policy of the original problem can be easily estimated. The exploration loss of an abstraction, introduced in definition 3.3, will capture this idea.

Although we may apply the proposed method to generic MDPs, our analysis focuses on a wide class of tasks that can be described with goal states.

Definition 3.2. We say that an MDP $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ is a *goal MDP* if there exists a set of goal states $\mathcal{G} \subseteq \mathcal{S}$ such that:

$$R(s, a, s') = 1 \quad \text{if } s \notin \mathcal{G} \text{ and } s' \in \mathcal{G}, \quad 0 \text{ otherwise} \tag{3.5}$$

$$V^*(s) = 0 \qquad \forall s \in \mathcal{G} \tag{3.6}$$

Equation (3.6) simply requires that from any goal state, it is not possible to re-enter any other goal and collect an additional reward. The task consists only in reaching any goal state efficiently. Goal MDPs are very straightforward definitions of tasks, but they are also sufficiently general, as we will see in the experimental section. In fact, many composite and temporal tasks can be captured by goal MDPs over an appropriately extended state space (Bacchus, Boutilier, et al. 1996; Brafman, De Giacomo, and Patrizi 2018; Icarte, T. Q. Klassen, et al. 2022).

Assumption 3.1. The ground MDP $\mathbf{M}_0$ is a goal MDP.

Assumption 3.2. Given each goal MDP $\mathbf{M}_i$, with goal states $\mathcal{G}_i$, and abstraction $\langle \mathbf{M}_{i+1}, \phi_i \rangle$, we assume $\mathbf{M}_{i+1}$ is a goal MDP with $\mathcal{G}_{i+1}$ satisfying:

$$\mathcal{G}_i = \cup_{s \in \mathcal{G}_{i+1}} \lfloor s \rfloor \tag{3.7}$$

In other words, the abstract goals should correspond through the mapping to all and only the goal states in the ground domain. In the example of fig. 3.3, the grey cells in the ground MDP are mapped to all and only the abstract goals labelled as G. We start our analysis with two observations. First, due to the way the framework is designed, convergence on any model $\mathbf{M}_i$, does depend on its abstraction, $\mathbf{M}_{i+1}$, but not on any other model in the hierarchy. Therefore, when discussing convergence properties, it suffices to talk about a generic MDP $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ and its direct abstraction $\bar{\mathbf{M}} = \langle \bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{T}, \bar{R}, \bar{\gamma} \rangle$. Let $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ denote the relevant mapping.

Second, while a goal MDP $\mathbf{M}$ has sparse rewards, the reward function of the biased MDP $\mathbf{M}^{\mathsf{b}}$ is no longer sparse. Depending on the abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$, from which it is defined, the rewards of $\mathbf{M}^{\mathsf{b}}$ can be as dense as needed. As confirmed empirically, this allows to achieve a faster convergence on the biased MDP. However, apart from convergence speed on $\mathbf{M}^{\mathsf{b}}$, the biased optimal policy should also be a good exploration policy for the original domain. Therefore, we measure how similar $\pi^{\mathsf{b}*}$, the optimal policy for the biased MDP, is to some optimal policy π^* of $\mathbf{M}$.

Definition 3.3. Given an MDP $\mathbf{M}$, the *exploration loss* of an abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$ is the value loss of executing $\pi^{\mathsf{b}*}$ in $\mathbf{M}$, that is the optimal policy of the biased MDP:

$$L(\mathbf{M}, \langle \bar{\mathbf{M}}, \phi \rangle) \coloneqq V^*_\mu - V^{\pi^{\mathsf{b}*}}_\mu \tag{3.8}$$

In order to bound this quantity in terms of the abstraction, we provide a more general result in theorem 3.4, which compares generic policies of ϕ-relative options. This is an independent result, and it will be applied to specific policies to obtain the final result in corollary 3.5.

To start the analysis, we first observe that each abstract state $\bar{s} \in \bar{\mathcal{S}}$ is related to the sets of states $\lfloor \bar{s} \rfloor \subseteq \mathcal{S}$ in the ground MDP. Similarly, abstract actions $\bar{a} \in \bar{\mathcal{A}}$ correspond to non-interruptible policies that only terminate when leaving the current block. So, a more appropriate correspondence can be identified between abstract actions $\bar{\mathcal{A}}$ and ϕ-relative options in $\mathbf{M}$. We start by deriving, in eq. (3.9), the multistep value of a ϕ-relative option in goal MDPs.

Multistep Value of Options By combining the classic multistep return of options (Sutton, Precup, and S. Singh 1999), ϕ-relative options (Abel, Umbanhowar, et al. 2020), and goal MDPs of definition 3.2, we obtain the following.

Lemma 3.2. *Given a goal MDP $\mathbf{M}$ and a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, for any $s \in \mathcal{S}$ and ϕ-relative option $o \in \Omega_{\phi(s)}$, the optimal value of o is:*

$$Q^*(s, o) = \sum_{k=0}^{\infty} \gamma^k \sum_{s_{1:k} \in \lfloor \phi(s) \rfloor^k} \sum_{s' \notin \lfloor \phi(s) \rfloor} \mathbb{P}(s_{1:k} s' \mid s, o) \left(\mathbb{I}(s' \in \mathcal{G}) + \gamma V^*(s') \right) \tag{3.9}$$

Proof. See page 53. □

This expression sums over any sequence of states $s_1 \dots s_k$ that remain within $\lfloor \phi(s) \rfloor$ for k steps and leave the block to reach some s'. A similar result was derived by Abel, Umbanhowar, et al. (2020) for a slightly different definition of goal MDPs. However, eq. (3.9) is not yet an expression about abstract states, because it depends on the specific ground state s' that is reached at the end of the option. Therefore, in the following definition, we introduce a parameter, ν, which quantifies how much the reachable states s' are dissimilar in value. This allows us to jointly talk about the value of each group of states as a whole and only refer to blocks. Specifically, we define a function $W_\nu : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}$, which, given a pair of abstract states $\bar{s}, \bar{s}'$, predicts, with ν-approximation error, the value of any successor ground state $s' \in \lfloor \bar{s}' \rfloor$ that can be reached from some $s \in \lfloor \bar{s} \rfloor$.

Definition 3.4. Consider an MDP $\mathbf{M}$ and a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$. We define the *abstract value approximation* as the smallest $\nu \geq 0$ such that there exists a function $W_\nu : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}$ which, for all distinct $\bar{s}, \bar{s}' \in \bar{\mathcal{S}}$, satisfies $\forall s \in \lfloor \bar{s} \rfloor$, $\forall s' \in \lfloor \bar{s}' \rfloor$, $\forall a \in \mathcal{A}$,

$$T(s' \mid s, a) > 0 \implies |W_\nu(\bar{s}, \bar{s}') - V^*(s')| \leq \nu \tag{3.10}$$

According to this definition, the frontier separating any two sets of states in the partition induced by ϕ must lie in ground states that can be approximated with the same optimal value, with a maximum error ν. This implies that any small ν places a constraint on the mapping function ϕ. In the example of fig. 3.3, each room is connected to each other through a single location, so this condition is simply satisfied for $\nu = 0$. However, this definition can be applied in the general case. In fig. 3.9, for example, a small ν means that every frontier state should have approximately the same optimal value.

Thanks to definition 3.4, it will be possible to bound the value of options, only taking future abstract states into consideration. For this purpose, when starting from some $s \in \mathcal{S}$ with a ϕ-relative option o, we use $\mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o)$ to denote the probability of the event of remaining for k steps in the same block as s, then reaching any $s' \in \lfloor \bar{s}' \rfloor$ in the next transition.

Lemma 3.3. *Consider a goal* $\mathbf{M}$ *and a function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$. *The value of any option* $o \in \Omega_{\phi(s)}$ *in any* $s \in \mathcal{S}$ *admits the following lower bound:*

$$Q^*(s, o) \geq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') - \nu \right) \right) \tag{3.11}$$

where, ν *and* W_ν *follow definition 3.4.*

Proof. See page 54. □

This lemma provides a different characterisation of options, in terms of abstract states, so that it can be exploited to obtain theorem 3.4.

Exploration Loss of Abstractions Thanks to the results of the previous section, we can now provide a bound for the exploration loss of definition 3.3, for any abstraction. We expand the results of Abel (2020) to limit this quantity.

First, we observe that, for any mapping function, any policy can also be regarded as a policy of options, in the specific sense of section 3.3. Then, from lemma 3.3, we know that an approximation for the value of options only depends on the k-step transition probability to each abstract state. So, we assume that this quantity is bounded as follows.

Definition 3.5. Given an MDP $\mathbf{M}$, a function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ and two policies of options Ω, Ω', we say that Ω and Ω' have *abstract similarity* ξ if

$$\forall s \in \mathcal{S}, \quad \forall \bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}, \quad \forall k \in \mathbb{N}$$
$$|\,\mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) - \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o')\,| \leq \xi \tag{3.12}$$

where the options o and o' are the only applicable in $\lfloor \phi(s) \rfloor$, meaning, $o \in \Omega \cap \Omega_{\phi(s)}$ and $o' \in \Omega' \cap \Omega_{\phi(s)}$.

Intuitively, abstract similarity measures the difference between the two abstract actions described by each policy, as it only depends on the probability of the next abstract state that is reached, regardless of the single trajectories and the specific final ground state. In the running example of fig. 3.3, two policies with low ξ, after the same number of steps, would reach the same adjacent room with similar probability. It is now finally possible to state the general result.

Theorem 3.4. *Consider a goal MDP* $\mathbf{M}$*, its optimal policy* Ω^**, a function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$*, and a policy of* ϕ*-relative options,* Ω*. If* ϵ *is the abstract similarity of* Ω^* *and* Ω*, and the abstract value approximation is* ν*, then,* Ω *is* ε*-optimal, with*

$$\varepsilon = \frac{2|\bar{\mathcal{S}}|(\xi + \gamma\,\nu)}{(1-\gamma)^2} \tag{3.13}$$

Proof. See page 55. □

The result shows that, provided that the abstraction induces a partition of states whose frontiers have some homogeneity in value (definition 3.4), it is possible to reason in terms of abstract transitions. Only for a $\nu = 0$, this bound has similarities with the inequality n. 5 in Abel, Umbanhowar, et al. (2020). Notice, however, that the one stated here is expressed in terms of the size of the abstract state space, which can usually be assumed to be much smaller than the ground state space, which might be potentially even infinite.

To conclude this section, we apply the general result just derived to two policies, π^*, the optimal policy of $\mathbf{M}$, and $\pi^{\mathsf{b}*}$, the optimal policy of the biased MDP. The corollary below shows under which conditions an abstraction induces an exploration policy that is similar to some optimal policy of the original domain. However, we recall that optimal convergence is guaranteed regardless of the abstraction quality, because the stochastic exploration policy satisfies the mild conditions posed by the adopted off-policy learning algorithm.

Corollary 3.5. *Consider an* $\mathbf{M}$ *and an abstraction* $\langle \bar{\mathbf{M}}, \phi \rangle$*, both satisfying assumptions 3.1 and 3.2. Let* ϵ *be the abstract similarity of* Ω^* *and* $\Omega^{\mathsf{b}*}$*, the optimal policies of options in* $\mathbf{M}$ *and* $\mathbf{M}^{\mathsf{b}}$*, and let* ν *be the abstract value approximation. Then, the exploration loss of the abstraction satisfies:*

$$L(\mathbf{M}, \langle \bar{\mathbf{M}}, \phi \rangle) \leq \frac{2|\bar{\mathcal{S}}|(\xi + \gamma\,\nu)}{(1-\gamma)^2} \tag{3.14}$$

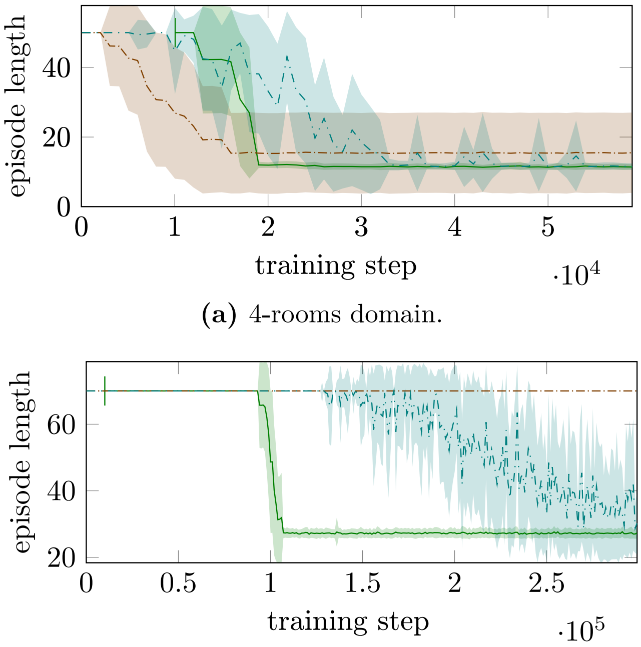
3.6 Validation

To verify the effectiveness of our reward shaping technique, we implemented the approach in a public repository at `https://github.com/cipollone/multinav2`. The instructions for executing the software and reproducing each plot are available in Appendix B and C of Cipollone, De Giacomo, et al. (2023b).

Environments We initially consider a navigation scenario, where some locations in a map are selected as goal states. We start with two levels of abstractions, $\mathbf{M}_1$ and $\mathbf{M}_2$. The ground MDP $\mathbf{M}_1$ consists of a finite state space $\mathcal{S}_1$, containing a set of locations, and a finite set of actions $\mathcal{A}_1$ that allows to move between neighbouring states, with some small failure probability at each transition. Following the idea of fig. 3.3, we also define an abstract MDP $\mathbf{M}_2$, whose states correspond to contiguous regions of $\mathcal{S}_1$. Actions $\mathcal{A}_2$ allow moving, with high probability, from any region to any other, only if there is a direct connection in $\mathbf{M}_1$. We instantiate this idea in two domains. In the first, we consider a map as the one in the classic "4-rooms" environment from Sutton, Precup, and S. Singh (1999) and fig. 3.2. The second is the "8-rooms" environment shown in fig. 3.3.

Training Results In the plots of figs. 3.5a and 3.5b, for each of the two ground MDPs, we compare the performance of the following algorithms:
- - - Q-learning (Watkins and Dayan 1992);
-·-· Delayed Q-learning (Strehl, L. Li, et al. 2006);
—— Algorithm 3.1 (our approach) with Q-learning.

Each episode is terminated after a fixed timeout or when the agent reaches a goal state. Therefore, shorter episode lengths are associated with higher cumulative discounted returns. The horizontal axis spans the number of sampled transitions. Each point in these graphs shows the average and standard deviation of the evaluations of 10 different runs. The solid green line of our approach is shifted to the right, to account for the number of time steps that were spent training the abstraction. Further training details, together with all instructions for reproducing each run, can be found in the appendix of Cipollone, De Giacomo, et al. (2023b).



(a) 4-rooms domain.

(b) 8-rooms domain.

Figure 3.5. Results on the two navigation tasks: -·-· Q-learning; -·-·· Delayed Q-learning; —— our approach.

As we can see from fig. 3.5a, all algorithms converge relatively easily in the small 4-rooms domain. In fig. 3.5b, as the state space increases and it becomes more difficult to explore, a naive exploration policy does not allow Q-learning to converge in reasonable time. Our agent, on the other hand, steadily converges to optimum, even faster than Delayed Q-learning which has polynomial-time guarantees and a more careful exploration strategy.

3.6.1 Return-Invariant Shaping

As discussed in section 3.4.1, when applying RS in the episodic setting, there is a technical but delicate distinction to make between:

---- Return-invariant RS (null potentials at terminal states);

—— Non return-invariant RS (our approach).

In fig. 3.6 (top), we compare the two RS variants in the 8-rooms domain. Although both agents receive RS from the same potential function, this minor modification suffices to produce this noticeable difference. The reason lies in the returns the two agents observe (bottom). Although they are incomparable in magnitude, in the early learning phase, we can see that only our reward shaping is able to reward each episode differently, depending on their estimated distance to the goal. Invariant RS, on the other hand, appears as a flat line.

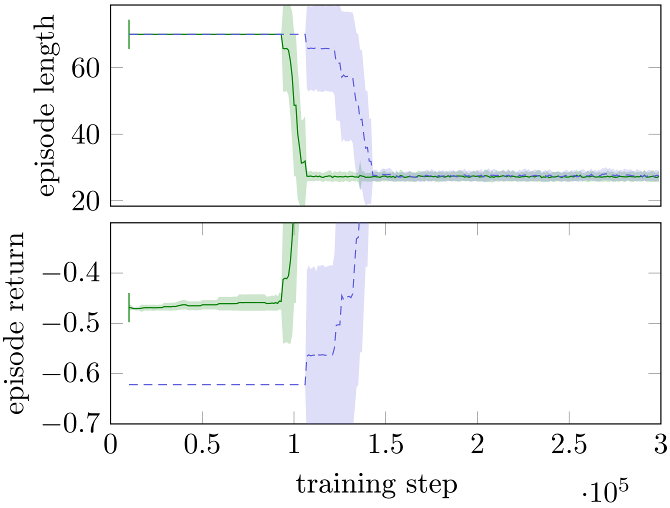


Figure 3.6. Return-invariant reward shaping - - - and our approach ——.

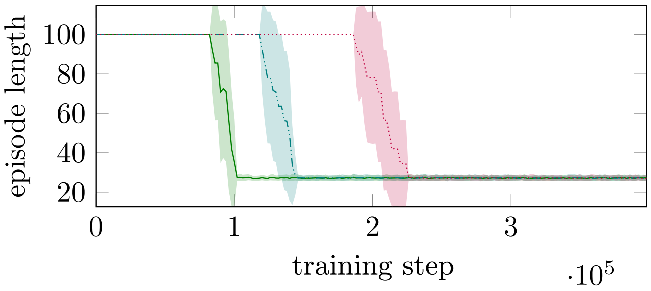


Figure 3.7. Training in presence of errors: consistent abstraction ——, one large mismatch -·-·-, two large mismatches ·········.

3.6.2 Robustness to Modelling Errors

We also considered the effect of significant modelling errors in the abstraction. In fig. 3.7, we report the performance of our agent on the 8-rooms domain, when driven by three different abstractions:

—— $\mathbf{M}_2$: is the same abstraction used in fig. 3.5b;

-·-·- $\mathbf{M}_2^{(b)}$: is $\mathbf{M}_2$ with an additional transition from the pink states (p) to the goal (G), not achievable in the ground MDP $\mathbf{M}_1$.

········· $\mathbf{M}_2^{(c)}$: is $\mathbf{M}_2^{(b)}$ with an additional transition from the blue (b) to the pink region (p), not achievable in the ground MDP $\mathbf{M}_1$.

Clearly, abstractions with larger differences with respect to the underlying domain cause the learning process to slow down. However, with any of these, Q-learning converges to the desired policy and the performance degrades gracefully. Interestingly, even in the presence of severe modelling errors, the abstraction still provides useful information with respect to uninformed exploration.

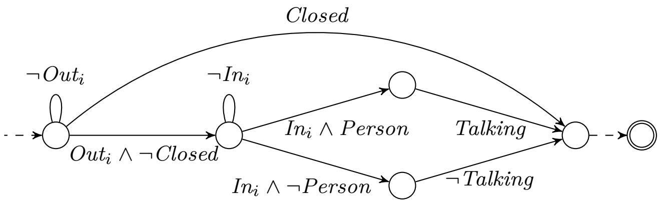


Figure 3.8. A temporally-extended task, repeated in series for $i = 1, 2$. The missing transitions go to a sink state representing irreparable failure.

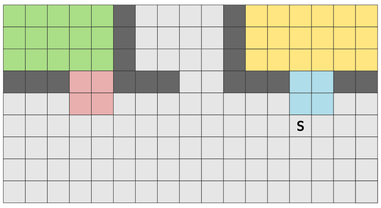


Figure 3.9. Two rooms environment with doors.

3.6.3 Interaction Task

In this section, we demonstrate that the proposed method applies to a wide range of algorithms, dynamics, and tasks. With respect to variability in tasks, we emphasise that goal MDPs can capture many interesting problems. For this purpose, instead of just reaching a location, we consider a complex temporally extended behaviour such as: "reach the entrance of the two rooms in sequence and, if each door is open, enter and interact with the person inside, if present". This task is summarised by the DFA $\mathbf{A}$ of fig. 3.8, whose structure should be replicated sequentially for each room (2 in this case). Note that there is a single accepting state, and arcs are associated to environment events. Events refer to the conditions and locations for the map in fig. 3.9. In particular, the two doors, which might be closed or open, are in the locations coloured as pink and light blue. The rooms behind them are green and yellow, and a person might only be found in those locations. Depending on the transition dynamics, we observe that this is a very challenging task: The optimal policy starts in the initial state "S", then proceeds towards the left door, checks if that is open, enters and interacts if appropriate, leaves the room, and repeats the process for the rightmost room. Apart from the reward being very sparse due to the whole sequence, trying to enter with a closed door or interacting when is not appropriate leads to irreparable failure of the episode.

Regarding generalisation with respect to environment dynamics, instead, we consider as ground MDP one with an infinite state space and continuous features. Specifically, let $\mathbf{M}_{2,d}$ and $\mathbf{M}_{1,d}$ be the tabular MDPs dynamics, structured as we have seen so far. Now we introduce a ground MDP $\mathbf{M}_{0,d}$ at which the robot

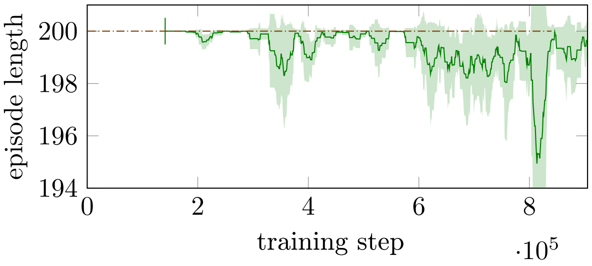


Figure 3.10. Dueling DQN algorithm with our RS —— and without -·-·-. Training episode lengths, averaged over 5 runs. Shorter episodes have higher returns.

movements are modelled using continuous features. The state space $\mathcal{S}_0$ now contains continuous vectors $(x, y, \theta, v) \in \mathrm{SE}(2) \times \mathbb{R}$, representing the pose and velocity of the agent's mobile base on the plane. The discrete set of actions $\mathcal{A}_0$ allows accelerating, decelerating, rotating, and a special action denotes the initiation of an interaction with a person. The specific navigation environment is shown in fig. 3.9, with colours representing the mapping function.

Using some results from the literature (Brafman, De Giacomo, and Patrizi 2018; Icarte, T. Klassen, et al. 2018; De Giacomo, Iocchi, et al. 2019; Icarte, T. Q. Klassen, et al. 2022), we can automatically construct goal MDPs, $\mathbf{M}_2, \mathbf{M}_1, \mathbf{M}_0$ that capture both the dynamics and the task defined above, which can be obtained through a suitable composition of each $\mathbf{M}_{i,d}$ and the DFA $\mathbf{A}$ that describes the complex task. Therefore, we can still apply our technique to the composed goal MDP.

Since $\mathbf{M}_0$ now includes continuous features, we combine our approach with Dueling DQN (Z. Wang, Schaul, et al. 2016), a Deep RL algorithm. The plot in fig. 3.10 shows a training comparison between the Dueling DQN agent alone (dot-dashed brown) and Dueling DQN receiving rewards from the grid-world abstraction (green). As we can see, our method provides a useful exploration bias even in case of extremely sparse goal states, as in this case. What we observe is not a full training, up to convergence. This only serves to demonstrate that uninformed Deep RL struggles to sample the optimum even once, whereas the biased exploration policy is able to achieve some complete rewarding sequences, which are essential for the rest of the learning.

3.7 Discussion

The principles of Hierarchical Reinforcement Learning and some important references have already been reviewed in section 3.2. In this small paragraph, we only want to draw some specific connections of this work with the relevant literature.

The use of multiple abstractions, organised as a linear hierarchy, is a broad idea that can be traced back to the origin of HRL, at Dayan and Hinton (1992). In this work, we take this intuition and extend it in the context of multiple MDPs. The simple algorithm that we obtain is guaranteed to converge to the optimum and the estimation errors obtained at one abstraction level do get propagated to the next below, unlike in other analysis (Abel, Umbanhowar, et al. 2020).

The use of Reward Shaping is extensive in the RL literature. In relation to its employment in HRL, we recall Gao and Toni (2015), whose technique is specific to the MAXQ algorithm. In the experimental section, we demonstrated that our technique can be coupled with temporally extended tasks and non-Markovian reward specifications (Brafman, De Giacomo, and Patrizi 2018; Icarte, T. Klassen, et al. 2018; De Giacomo, Iocchi, et al. 2019; Icarte, T. Q. Klassen, et al. 2022). Our use of RS, specifically for the application to non-Markovian rewards, can be compared with Camacho, O. Chen, et al. (2017). However, in this last work, the reward specification, in the form of a DFA, is regarded as a deterministic MDP in which every event is possible. Our formulation captures this reward shaping, as well as others, in which it is possible to represent that some events may be very unlikely to happen, and as a consequence, they should be associated to very low potential values. Finally, for abstractions with respect to task complexity, not domain, we recall Furelos-Blanco, Law, et al. (2022).

As we have already discussed in the text, the theoretical analysis of this work has been influenced by Abel, Umbanhowar, et al. (2020) and Abel (2020). Similarly to these works, we adopt ϕ-relative options to describe the object that corresponds to abstract actions. However, these works only use ϕ-relative options to describe policies in the ground MDP, but they do not consider any real dynamics over the abstract state space. In this work, on the other hand, we go beyond a simple state partitioning and consider full MDPs as abstractions. Their purpose, in fact, was not to show how effective HRL abstractions can be.

Our notion of abstract value approximation has interesting connections with the "suboptimality of exit profiles" of Wen, Precup, et al. (2020), especially in the case where the exit profile has a constant value for each neighbouring abstract state. The role of exit states will be studied in more detail in the next chapter. The second parameter, abstract similarity, is analogous to the one found in Abel, Umbanhowar, et al. (2020), with the significant difference that only abstract states are considered, not ground ones. This is important since assumptions regarding ground states are very restrictive, and they can hardly be understood and verified in practice. The abstract state space, which is always assumed to be discrete, is a more appropriate target for such assumptions.

Future Work After some significant advantages, it is also worth emphasising the limitations of this work and the opportunities for future development. Regarding the experimental validation, we tested robustness to modelling errors, compared against return-invariant shaping, and how our approach compares to other RL algorithms. However, a more complete validation should also include a comparison with other HRL approaches, especially those that utilise a similar amount of prior knowledge from the abstraction.

We may also discuss the limitations of the theoretical treatment in section 3.5. With this work, we provided one possible answer to question 2 at page 27, namely, how can abstraction be used to improve sample efficiency of RL. Moreover, the guarantee that we obtained in corollary 3.5 is also a possible answer for question 1, that asks what is a "good" abstraction of an MDP. However, this last answer to question 1 has two limitations: first, this definition of "good" abstractions is closely related to the specific algorithm that we described, but it lacks a more generic applicability and insight for this important HRL question; secondly, the two parameters that we used to describe abstractions are hard to evaluate or estimate for generic pairs of ground MDPs $\mathbf{M}$ and abstractions $\langle \bar{\mathbf{M}}, \phi \rangle$. The exact purpose of chapter 4 is to address all of these critiques in relation to the theoretical treatment of abstractions in this chapter.

3.8 Proofs

This section contains all the proofs for this chapter. The reader may skip this section and refer to it as needed.

Proposition 3.1. *Consider MDPs $\mathbf{M}_0, \dots, \mathbf{M}_n$ and their associated mapping functions $\phi_0, \dots, \phi_{n-1}$. If $\mathfrak{A}$ is an off-policy RL algorithm, then, in every i-th iteration of algorithm 3.1, $\hat{\pi}_i^*$ converges to π_i^*, as the number of environment interactions increases.*

Proof. At any iteration $i \in \{n, \dots, 0\}$ of algorithm 3.1, we are given $\mathbf{M}_i$, ϕ_i and $\hat{V}_{i+1}^*$. By construction, the two instantiations of $\mathfrak{A}$, *Learner*$_i$ and *Learner*$_i^{\mathsf{b}}$, perform updates from transitions generated from $\mathbf{M}_i$ and $\mathbf{M}_i^{\mathsf{b}}$, respectively. Actions are selected according to *Learner*$_i^{\mathsf{b}}$ which follows some exploration policies $\{\pi_l^{\mathsf{b}}\}_l$. Since $\mathbf{M}_i$ and $\mathbf{M}_i^{\mathsf{b}}$ share the same state and action spaces, the off-policy algorithm $\mathfrak{A}$ also converges in $\mathbf{M}_i$, under the same policies. Therefore, the output of *Learner*$_i$ also converges to π_i^*, as the number of environment interactions $t \to \infty$. □

Lemma 3.2. *Given a goal MDP* $\mathbf{M}$ *and a function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$*, for any* $s \in \mathcal{S}$ *and* ϕ*-relative option* $o \in \Omega_{\phi(s)}$*, the optimal value of* o *is:*

$$Q^*(s,o) = \sum_{k=0}^{\infty} \gamma^k \sum_{s_{1:k} \in \lfloor \phi(s) \rfloor^k} \sum_{s' \notin \lfloor \phi(s) \rfloor} \mathbb{P}(s_{1:k}s' \mid s, o) \left(\mathbb{I}(s' \in \mathcal{G}) + \gamma\, V^*(s') \right) \tag{3.9}$$

Proof. By assumption, $\mathbf{M}$ is a goal MDP over some $\mathcal{G} \subseteq \mathcal{S}$. For $s \in \mathcal{G}$, we know that $Q^*(s,o) = 0$. We consider $s \notin \mathcal{G}$. Since the MDP $\mathbf{M}$ is clear from the context, we use $\mathbb{P}(s' \mid s, a)$ to mean $T(s' \mid s, a)$. Actions will be selected according to the option policy π_o. Following a similar procedure as Abel, Umbanhowar, et al. (2020), for our definition of goal MDPs:

$$\begin{aligned} Q^*(s,o) &\coloneqq \mathbb{E}_{s'|s,o}[R(s, \pi_o(s), s') \; + \gamma\, (\\ &\qquad \mathbb{I}(s' \in \lfloor \phi(s) \rfloor)\, Q^*(s', o) + \mathbb{I}(s' \notin \lfloor \phi(s) \rfloor)\, V^*(s'))] && (3.15) \\ &= \sum_{s' \in \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s))[\,\cdot\,] + \sum_{s' \notin \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s))[\,\cdot\,] && (3.16) \\ &= \sum_{s' \in \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s))\, \gamma\, Q^*(s', o) + \\ &\qquad \sum_{s' \notin \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s)) (\mathbb{I}(s' \in \mathcal{G}) + \gamma\, V^*(s')) && (3.17) \end{aligned}$$

We abbreviate the second term of eq. (3.17) with Ψ and let $s_0 = s$. Then, similarly to the classic multistep value of options (Sutton, Precup, and S. Singh 1999), we can expand over time.

$$\begin{aligned} Q^*(s,o) = &\sum_{s' \in \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s))\, \gamma\, Q^*(s', o) \; + \Psi(s, o) && (3.18) \\ &= \Psi(s,o) + \gamma \sum_{s' \in \lfloor \phi(s) \rfloor} \mathbb{P}(s' \mid s, \pi_o(s))\, \Psi(s', o) + \\ &\qquad \gamma^2 \sum_{s', s'' \in \lfloor \phi(s) \rfloor^2} \mathbb{P}(s'\, s'' \mid s, \pi_o(s)\, \pi_o(s'))\, Q^*(s'', o) && (3.19) \\ &= \sum_{k=0}^{\infty} \gamma^k \sum_{s_{1:k} \in \lfloor \phi(s) \rfloor^k} \mathbb{P}(s_{1:k} \mid s, \pi_o)\, \Psi(s_k, o) && (3.20) \\ &= \sum_{k=0}^{\infty} \gamma^k \sum_{s_{1:k} \in \lfloor \phi(s) \rfloor^k} \sum_{s' \notin \lfloor \phi(s) \rfloor} \mathbb{P}(s_{1:k}s' \mid s, \pi_o) \left(\mathbb{I}(s' \in \mathcal{G}) + \gamma\, V^*(s') \right) && (3.21) \end{aligned}$$

□

Lemma 3.3. *Consider a goal* $\mathbf{M}$ *and a function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$. *The value of any option* $o \in \Omega_{\phi(s)}$ *in any* $s \in \mathcal{S}$ *admits the following lower bound:*

$$Q^*(s,o) \geq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') - \nu \right) \right) \tag{3.11}$$

where, ν *and* W_ν *follow definition 3.4.*

Proof. We recall that $\mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o)$ denotes the probability of the event of remaining for k steps within $\lfloor \phi(s) \rfloor$, then reaching $\bar{s}'$ at the next transition, when following option $o \in \Omega_{\phi(s)}$, starting from s. Also, we use $\mathbb{P}(k, s_{k+1} = s' \mid s, o)$ to represent the associated event of terminating in a specific ground state $s' \in \mathcal{S} \setminus \lfloor \phi(s) \rfloor$ after k transitions.

To obtain the result, we marginalize the probabilities appearing in lemma 3.2 over all possible trajectories $s_{1:k}$:

$$Q^*(s,o) = \sum_{s' \in \mathcal{S} \setminus \lfloor \phi(s) \rfloor} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} = s' \mid s, o) \left(\mathbb{I}(s' \in \mathcal{G}) + \gamma \, V^*(s') \right) \tag{3.22}$$

Now, for all s', k such that $\mathbb{P}(k, s_{k+1} = s' \mid s, o) > 0$, there is one state $s_k \in \lfloor \phi(s) \rfloor$, reachable in k steps from s under o, from which $T(s' \mid s_k, \pi_o(s_k)) > 0$. From definition 3.4, we know $|W_\nu(\phi(s_k), \phi(s')) - V^*(s')| \leq \nu$. Therefore, we can provide a lower bound for each term $V^*(s')$ in the sum above:

$$Q^*(s,o) \geq \sum_{s' \in \mathcal{S} \setminus \lfloor \phi(s) \rfloor} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} = s' \mid s, o) \left(\mathbb{I}(s' \in \mathcal{G}) + \gamma \left(W_\nu(\phi(s), \phi(s')) - \nu \right) \right) \tag{3.23}$$

because $\phi(s_k) = \phi(s)$. It is now possible to split the sum $\sum_{s' \in \mathcal{S} \setminus \lfloor \phi(s) \rfloor}$ into $|\bar{\mathcal{S}}| - 1$ sums over future blocks and marginalize among them to obtain:

$$Q^*(s,o) \geq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') - \nu \right) \right) \tag{3.24}$$

since $\mathbb{I}(s \in \mathcal{G}) = \mathbb{I}(\phi(s) \in \bar{\mathcal{G}})$. This proves the lemma. With the same procedure, we also obtain the upper bound:

$$Q^*(s,o) \leq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') + \nu \right) \right) \tag{3.25}$$

□

Theorem 3.4. *Consider a goal MDP* $\mathbf{M}$*, its optimal policy* Ω^**, a function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$*, and a policy of* ϕ*-relative options,* Ω*. If* ϵ *is the abstract similarity of* Ω^* *and* Ω*, and the abstract value approximation is* ν*, then,* Ω *is* ε*-optimal, with*

$$\varepsilon = \frac{2|\bar{\mathcal{S}}|(\xi + \gamma\,\nu)}{(1-\gamma)^2} \tag{3.13}$$

Proof. To prove the result, we first compute the sub-optimality that is caused by executing one option from Ω. This will be extended to multiple options to conclude the proof. In other words, we compute what is the difference in value between executing Ω^* and Ω, for one option each, then following the optimal policy π^* afterwards. For any $s \in \mathcal{S} \setminus \mathcal{G}$, let o^* and o be the relevant options in Ω^* and Ω, respectively. We bound the following difference in value:

$$|Q^*(s, o^*) - Q^*(s, o)| = Q^*(s, o^*) - Q^*(s, o) \tag{3.26}$$

From an application of the upper and lower bound of lemma 3.3,

$$|Q^*(s, o^*) - Q^*(s, o)| \leq \tag{3.27}$$

$$\begin{aligned} \leq & \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o^*) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') + \nu \right) \right) \\ & - \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \, \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \left(W_\nu(\phi(s), \bar{s}') - \nu \right) \right) \end{aligned} \tag{3.28}$$

$$\begin{aligned} = & \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \, W_\nu(\phi(s), \bar{s}') \right) \sum_{k=0}^{\infty} \gamma^k \, (\\ & \quad \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o^*) - \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o)) + \\ & \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^k \left(\mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o^*) + \mathbb{P}(k, s_{k+1} \in \lfloor \bar{s}' \rfloor \mid s, o) \right) \gamma\, \nu \end{aligned} \tag{3.29}$$

Now we apply the abstract similarity of definition 3.5 and bound

$$|Q^*(s, o^*) - Q^*(s, o)| \tag{3.30}$$

$$\leq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \, W_\nu(\phi(s), \bar{s}') \right) \sum_{k=0}^{\infty} \gamma^k \, \xi + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \sum_{k=0}^{\infty} \gamma^{k+1} \, 2\, \nu \tag{3.31}$$

$$= \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \left(\left(\mathbb{I}(\bar{s}' \in \bar{\mathcal{G}}) + \gamma \, W_\nu(\phi(s), \bar{s}') \right) \frac{\xi}{1-\gamma} + \frac{2\,\gamma\,\nu}{1-\gamma} \right) \tag{3.32}$$

Since in a goal MDP the maximum value is 1, we know $W_\nu(\phi(s), \bar{s}') \leq (1 + \nu)$, for all $s \in \mathcal{S}, \bar{s}' \in \bar{\mathcal{S}}$. Moreover, if W_ν satisfies eq. (3.10), the function $W_\nu^{\text{clip}}(\bar{s}, \bar{s}') \coloneqq \min\{1, W_\nu(\bar{s}, \bar{s}')\}$ also satisfies it. Concluding,

$$|Q^*(s, o^*) - Q^*(s, o)| \leq \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\phi(s)\}} \Big((1+\gamma)\frac{\xi}{1-\gamma} + \frac{2\,\gamma\,\nu}{1-\gamma}\Big) \tag{3.33}$$

$$\leq |\bar{\mathcal{S}}|\Big(\frac{2\,\xi}{1-\gamma} + \frac{2\,\gamma\,\nu}{1-\gamma}\Big) \tag{3.34}$$

$$= \frac{2|\bar{\mathcal{S}}|(\xi + \gamma\,\nu)}{1-\gamma} \tag{3.35}$$

This is a bound on the value loss of executing a single option from the set Ω. To this option set, equation (3) in Abel, Umbanhowar, et al. (2020) applies:

$$\max_{s \in \mathcal{S}} V^*(s) - V^\Omega(s) = \frac{2|\bar{\mathcal{S}}|(\xi + \gamma\,\nu)}{1-\gamma} \tag{3.36}$$

Then, the statement is also true, for any initial distribution μ. □

Chapter 4

Realizing MDP decompositions

> The content of this chapter is based on original work, developed in collaboration with Luca Iocchi and Matteo Leonetti, and it will be included as part of a future submission.

The previous chapter proposed a specific approach for utilising MDP abstractions in RL algorithms, and it provided some results regarding how to relate the abstraction and the ground MDP. This chapter is the natural continuation of the previous, since it steps back from specific algorithms and applications, and aims to answer a fundamental question: What is a "good" MDP abstraction? And, consequently, what properties should it possess? These questions are very relevant for the broad HRL community.

This chapter builds on previous material. We expect the reader to be familiar with the background in section 3.2, for a broader picture of Hierarchical RL.

4.1 Introduction

There is a common intuition that drives many authors in HRL. That is, abstract states correspond to sets of ground states, and abstract actions correspond to sequences of ground actions. This was evident since the early work in HRL (Dayan and Hinton 1992), and was largely derived from Hierarchical Planning, which HRL extends. The works that focus on this double correspondence are the ones reported in section 3.2 as state-action abstractions. However, two precise questions remain still unanswered: *which set* of ground states should each abstract state correspond to? Similarly, *which sequence* of actions should each abstract action correspond to? The specific answers for both of these questions have a strong impact on the applicability and the guarantees of the resulting abstractions.

Moreover, there is even no shared consensus on what MDP abstractions should refer to. In the literature, the term "abstraction" is loosely used to refer to a variety of concepts, including state partitions (L. Li, Walsh, et al. 2006; Wen, Precup, et al. 2020), bottleneck states (Jothimurugan, Bastani, et al. 2021), goal states (Nachum, Gu, et al. 2018), options (Precup and Sutton 1997; Khetarpal, Klissarov, et al. 2020), entire MDPs (Ravindran and Andrew G. Barto 2002; Cipollone, De Giacomo, et al. 2023a), or even the natural language (Y. Jiang, Gu, et al. 2019). Because of all these scattered notions, this chapter explicitly and formally defines what MDP abstractions are and what properties they should satisfy. This is not made in an attempt to identify the best single definition of MDP abstractions and to discard all the others. In fact, each work may have different needs and desires to provide a specific amount of prior knowledge to the HRL algorithm they develop. However, as we will see, the abstractions that we propose are general, they allow agents to reason in a truly compositional way, and they satisfy near-optimal properties.

In this work, we say that the abstraction of an MDP is another decision process together with a mapping function that relates the two state spaces. Specifically, an abstraction for some MDP $\mathbf{M}$ is defined as the pair $\langle \bar{\mathbf{M}}, \phi \rangle$, where $\bar{\mathbf{M}}$ is a 2-MDP and $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ is a function connecting the two state spaces. We recall that k-MDPs have been defined on page 18. This choice will be motivated in the next sections, but, generally, 2-MDPs come from the need to capture dependencies on the previous time step. With a minor inaccuracy, we will naturally say that MDPs *are* also 2-MDPs. Specifically, they are 2-MDPs in which the penultimate observation and action have no influence on the transition and reward functions. This is important because some pair $\langle \bar{\mathbf{M}}, \phi \rangle$, where $\bar{\mathbf{M}}$ is an MDP, also fits our definition of MDP abstractions. This makes the abstraction used here consistent with that of chapter 3 and possibly others in the literature, where abstract MDPs are more common (Ravindran and Andrew G. Barto 2002).

Within this generic framework, we will be able to identify criteria that make an MDP abstraction appropriate for some ground MDP. This answers the general question of what is a "good" MDP abstraction and also identifies which ground elements each abstract state or action corresponds to.

We summarise the contributions of this work.

- In section 4.3, we transform the most common intuitions in HRL into specific relations, involving well known elements of MDPs. In particular, the probability of the abstract transitions will be related to the occupancy measures of the ground options. Similarly, the immediate rewards will be related to specific ground values. These relationships are original contributions of this work. Based on this, we define a new notion called *realizable abstractions.*

- In section 4.4, we verify that realizable abstractions allow us to obtain near-optimal policies in the ground MDP. In particular, theorem 4.1 shows that, if an MDP abstraction is approximately realizable, any abstract policy can be implemented as a policy for the ground MDP, achieving approximately the same value obtained in the abstraction. Moreover, the ground policy can be obtained compositionally.

- In section 4.5, we explore two complementary problems. We address how realizable abstractions can be realized into ground policies, and how ground options can be represented as abstract transitions. Both are important steps towards the development of learning algorithms based on realizable abstractions.
- Finally, the analysis of the abstraction process allowed us to identify a specific relationship between the abstract and the ground effective horizons, $(1-\gamma)^{-1}$ and $(1-\bar{\gamma})^{-1}$. It is a shared opinion that $\bar{\gamma}$ may be reduced with respect to γ, if the abstraction allows shortening of the effective horizon. However, we identify a necessary condition, in assumption 4.1, that expresses when this compression is indeed possible.

Although we will generally use mathematical notation for finite spaces, all properties and methods remain well defined for ground MDPs with infinite state spaces.

4.2 Preliminaries

This chapter builds on the conventions set in the previous chapters. Apart from the classic notions related to MDPs, we will need 2-MDPs, defined from k-MDPs on page 18, occupancy measures at page 22, options, ϕ-relative options, and policies of options from section 3.3.

For a uniform notation, in this chapter we use the term "states", and write $s \in \mathcal{S}$, for the observations that are generated by both MDPs and 2-MDPs. In particular, 2-MDPs will be used as models for the abstract decision process. This allows us to model dependencies on the penultimate state. Since the penultimate action will not be used, we only consider 2-MDPs in which transitions and reward functions are constant with respect to it. Also, we take the reward functions to be deterministic in the last state and action. This last choice is only a slight simplification and only for the learning setting, where rewards are sampled. For the most part, deterministic rewards are fully equivalent to expectations of stochastic rewards functions. In summary, in this chapter we work with *2-Markov Decision Processes* defined as tuples $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where the transition and reward functions are $T : \mathcal{S}^2 \to \Delta(\mathcal{S})$ and $R : \mathcal{S}^2 \to \mathcal{R}$. As usual, without loss of generality, we take rewards normalized as $\mathcal{R} \subseteq [0, 1]$. These models evolve as follows: $s_0 \sim \mu \coloneqq T(s_\circ s_\circ a_\circ)$, $s_1 \sim T(s_\circ s_0 a_1)$, and

$s_t \sim T(s_{t-2} s_{t-1} a_t)$, where $s_\circ$ and $a_\circ$ are reserved start symbols. Rewards evolve in a similar way, according to the function R. An MDP is a 1-Markov Decision Process.

Regarding options in MDPs, we extend the classic definition by generalising initiation sets to pairs of states, thus introducing a similar dependency to the one of 2-MDPs. An *option* for an MDP $\mathbf{M}$, is a temporally extended action, defined in this chapter as $o = \langle \mathcal{I}_o, \pi_o, \beta_o \rangle$, where $\mathcal{I}_o \subseteq \mathcal{S}^2$ is an initiation set composed of pairs of states, $\pi_o \in \Pi$ is the policy that o executes, and $\beta_o : \mathcal{S} \to \{0, 1\}$ is a termination condition. Then, any option is applicable at the end of a trajectory $a_{t-1} r_{t-1} s_{t-1} a_t r_t s_t$ iff $s_{t-1} s_t \in \mathcal{I}_o$. Note that this is not meant to be an option for 2-MDPs, but for MDPs, because the policy remains Markovian, only the initiation set is generalised. So, this is only a minor change.

The option $o = \langle \mathcal{I}_o, \pi_o, \beta_o \rangle$ is said to be *ϕ-relative* if there exists two distinct $\bar{s}_p, \bar{s} \in \bar{\mathcal{S}}$ such that

$$\mathcal{I}_o = \lfloor \bar{s}_p \rfloor_\phi \times \lfloor \bar{s} \rfloor_\phi, \qquad \beta_o(s) = \mathbb{I}(s \notin \lfloor \bar{s} \rfloor_\phi), \qquad \pi_o \in \Pi_{\bar{s}} \tag{4.1}$$

where $\Pi_{\bar{s}} : \lfloor \bar{s} \rfloor_\phi \to \Delta(\mathcal{A})$ is the set of partial policies defined for the relevant block. Some set of options Ω is ϕ-relative iff all of its options are. In the following, we only consider sets ϕ-relative options $\Omega = \{o \in \Omega_{\bar{s}_p \bar{s}} \text{ for } \bar{s}_p \bar{s} \in \bar{\mathcal{S}}^2, \bar{s}_p \neq \bar{s}\}$, where $\Omega_{\bar{s}_p \bar{s}}$ is the set of all options satisfying eq. (4.1). In addition, we write generic ϕ-relative options for $\bar{s}$ as $\Omega_{\bar{s}} \coloneqq \cup_{\bar{s}_p} \Omega_{\bar{s}_p \bar{s}}$. Anhigh-level deterministicel policy $\bar{\pi} : \bar{\mathcal{S}}^2 \to \Omega$ over ϕ-relative options Ω corresponds to a unique subset $\Omega' \subseteq \Omega$, containing one option per $\lfloor \bar{s}_p \rfloor \lfloor \bar{s} \rfloor$ pair. We call Ω' a *policy of options*, as it can be fully treated as a policy. In particular, $V^{\Omega'}$ is value of the policy that always executes the only applicable option in Ω' until termination.

Finally, to discuss connections with other methods from the literature, we also introduce the following definition.

MDP Homomorphisms MDP homomorphisms are a classic formalism for MDP minimisation (Ravindran and Andrew G. Barto 2002). A homomorphism from an MDP $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ to another $\bar{\mathbf{M}} = \langle \bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{T}, \bar{R}, \gamma \rangle$ is a pair $\langle f, \{g_s\}_{s \in \mathcal{S}} \rangle$, with a function $f : \mathcal{S} \to \bar{\mathcal{S}}$ and surjections $g_s : \mathcal{A} \to \bar{\mathcal{A}}$, satisfying

$$\bar{T}(f(s') \mid f(s)\, g_s(a)) = \sum_{s'' \in \lfloor f(s') \rfloor} T(s'' \mid s\, a) \tag{4.2}$$

$$\bar{R}(f(s)\, g_s(a)) = R(s\, a) \tag{4.3}$$

for all $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$. For simplicity, here we assumed that all actions are applicable in any state. MDP homomorphisms can also be generalised to be approximate as shown in Ravindran and Andrew G Barto (2004).

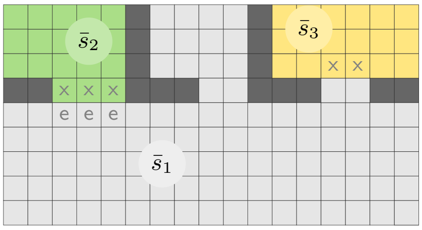


Figure 4.1. In this example, the ground MDP is a grid-world domain, and the abstraction has three states. Entries e and exits × are explained in the paragraph of eq. (4.4).

4.3 Realizable Abstractions

As anticipated in the introduction, we say that the abstraction of an MDP $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ is some pair $\langle \bar{\mathbf{M}}, \phi \rangle$, where $\bar{\mathbf{M}} = \langle \bar{\mathcal{S}}, \bar{\mathcal{A}}, \bar{T}, \bar{R}, \bar{\gamma} \rangle$ is a 2-MDP, and $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ is a surjective function. Any such mapping function ϕ induces a partition over the ground state space as $\{\lfloor \bar{s} \rfloor\}_{\bar{s} \in \bar{\mathcal{S}}}$. As we can see, each component of the tuple that defines the abstract decision process can differ. Thus, it is essential to establish some precise relation between the two models.

We will start this section with some intuitions, examples, and desired properties. All these generic requirements will be then captured by the notion of "realizable abstractions". This will also motivate our use of abstract 2-MDPs instead of the more classic MDPs. Let us set up the following running example of fig. 4.1. This example will be used for illustrating some properties, but abstractions should be very generic, and they should be appropriate for a variety of domains. The ground MDP of the example is a small grid-world domain with actions that allow to move in the four cardinal directions, unless there is a wall. Optionally, we may also consider a failure probability with some unwanted effect. As abstraction, we choose a decision process composed of three states $\bar{\mathcal{S}} = \{\bar{s}_1, \bar{s}_2, \bar{s}_3\}$ and three actions $\bar{\mathcal{A}} = \{\bar{a}_1, \bar{a}_2, \bar{a}_3\}$. The purpose of each action is to represent a "go to" movement for each destination state. In the first discussion, we will say that $\bar{\mathbf{M}}$ is an MDP. Finally, the mapping function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ is chosen as indicated by the colours.

Assuming that all these elements are set, how should the other elements be defined? How should we define the transition function, reward function, and discount factor in the abstract decision process? If we choose to model the abstract decision process as an MDP, we could make the following choices. Intuitively, the probability $\bar{T}(\bar{s}_3 \mid \bar{s}_1\, \bar{a}_3)$ should represent how likely it is, in the ground MDP, to end up in the yellow block, when starting from the grey block, under some policy represented by $\bar{a}_3$. This policy should be some that attempts to reach any state in $\lfloor \bar{s}_3 \rfloor$. So, as an extreme case, $\bar{T}(\bar{s}_3 \mid \bar{s}_2\, \bar{a}_3) = 0$ should mean that no direct path is possible from the green to the yellow block, which is the case in fig. 4.1. Regarding rewards, we say that $\bar{R}(\bar{s}_1\bar{a}_3)$ should represent the total return accumulated while executing the policy represented by $\bar{a}_3$ in $\lfloor\bar{s}_1\rfloor$.

We already identified a correspondence between the probability of abstract transitions and the likelihood of ground paths between the corresponding blocks. The second relationship we find is between abstract probabilities and a notion of time in the ground MDP. In fact, we might expect that the cost (value loss) of the abstract transition $\bar{s}_1 \rightsquigarrow \bar{s}_3$ should also increase if, in the ground MDP, any path between $\lfloor\bar{s}_1\rfloor$ and $\lfloor\bar{s}_3\rfloor$ requires many steps to complete. This relationship only holds when working with discounted values. In fact, $\bar{T}(\bar{s}_3 \mid \bar{s}_1\,\bar{a}_3)$ will also be low if many steps are needed between the two respective blocks. Due to this effect, our abstractions do not force a specific discount factor, but allow $\bar{\gamma} < \gamma$, whenever the ground MDP admits a *uniform compression of the effective horizon.* This choice, whenever possible, allows for transferring some value loss from the abstract transition probabilities to the abstract discount factor.

These intuitions already allow one to make some specific choices. Since abstract transitions should represent *direct paths* in the ground MDP, which do not end until a new block is reached, we choose to associate abstract actions $\bar{\mathcal{A}}$ to ϕ-relative options in $\mathbf{M}$. Also, because of this probability–time duality, abstract transitions will be put in relation with occupancy measures of the ground MDP. Similarly, abstract rewards will be related to specific ground value functions.

Finally, we motivate our choice of 2-MDPs for the abstract decision process. In the above example, if $\bar{\mathbf{M}}$ is an MDP, then a single transition probability $\bar{T}(\bar{s}_3 \mid \bar{s}_1\,\bar{a}_3)$ would be taken as representative of a multitude of initial states in the grey block $\lfloor\bar{s}_1\rfloor$. Since there might be many states in each block, in general, it would be infeasible to represent their dynamics under a single value. Even in the small example of fig. 4.1, depending on the initial grey state, the shortest path from $\lfloor\bar{s}_1\rfloor$ to $\lfloor\bar{s}_3\rfloor$ can take from 1 to 18 steps. This issue is at the heart of what the literature calls the non-stationarity effect of Hierarchical RL representations. To address this issue, we observe that this path length is strongly dependent on whether the penultimate block was $\lfloor\bar{s}_3\rfloor$ or $\lfloor\bar{s}_2\rfloor$ (being very easy to immediately re-enter $\lfloor\bar{s}_3\rfloor$ from $\lfloor\bar{s}_1\rfloor$, just after leaving it). For this reason, we generally consider abstract decision processes as 2-MDPs, in which it is possible to express that $\bar{T}(\bar{s}_3 \mid \bar{s}_3\,\bar{s}_2\,\bar{a}_3)$ should be associated to a much higher value than $\bar{T}(\bar{s}_3 \mid \bar{s}_1\,\bar{s}_2\,\bar{a}_3)$.

We now proceed to make all the statements and intuitions formal. For each two distinct abstract states $\bar{s}_p, \bar{s} \in \bar{\mathcal{S}}$, we define the set of *entry states* of $\bar{s}$ as the set of ground states of $\lfloor\bar{s}\rfloor$, at which it is possible to enter $\lfloor\bar{s}\rfloor$ from $\lfloor\bar{s}_p\rfloor$. Namely,

$$\mathcal{E}_{\bar{s}_p\bar{s}} \coloneqq \{s \in \lfloor\bar{s}\rfloor \mid \exists s_p \in \lfloor\bar{s}_p\rfloor, \exists a \in \mathcal{A},\, T(s \mid s_p\,a) > 0\} \tag{4.4}$$

Also, we define the set of *exit states* of $\lfloor \bar{s} \rceil$ as all the ground states outside the block, reachable in one transition. This is $\mathcal{X}_{\bar{s}} := \cup_{\bar{s}' \neq \bar{s}} \mathcal{E}_{\bar{s}\bar{s}'}$. Exit states are an intuitive way to discuss boundaries in contiguous state partitions and are often found in the HRL literature (Wen, Precup, et al. 2020; Infante, Jonsson, et al. 2022). In fig. 4.1, each entry state in $\mathcal{E}_{\bar{s}_p\bar{s}}$ is marked with an e, and each exit in $\mathcal{X}_{\bar{s}}$ is marked with an x. There is one last possibility of entering a block, that is, through the initial distribution μ. Although this is a technical and less interesting case, to capture this possibility we will also consider the entries $\mathcal{E}_{\bar{s}_\circ\bar{s}}$. In this regard, we assume that the start states are properly mapped as $\lfloor \bar{s}_\circ \rceil_\phi = \{s_\circ\}$.

A careful treatment of exits and entries is essential, as it allows us to develop a truly compositional approach in which each block is treated separately. For this purpose, associated with each abstract state, we define the block MDP as the portion of the original MDP that is restricted to a single block and its exit states.

Definition 4.1. Given an MDP $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ and a surjective function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, we define *block MDP* of some $\bar{s} \in \bar{\mathcal{S}}$ as $\mathbf{M}_{\bar{s}} = \langle \mathcal{S}_{\bar{s}}, \mathcal{A}, T_{\bar{s}}, R_{\bar{s}}, \gamma \rangle$, where: $\mathcal{S}_{\bar{s}} := \lfloor \bar{s} \rceil \cup \mathcal{X}_{\bar{s}} \cup \{s_\perp\}$ includes the interested block, all the reachable states in one transition, and a new sink state $s_\perp$; the transition function is $T_{\bar{s}}(sa) := T(sa)$ if $s \in \lfloor \bar{s} \rceil$ and $T(sa) := \delta_{s_\perp}$ otherwise;[1] the reward function is $R_{\bar{s}}(sa) := R(sa)$ if $s \in \lfloor \bar{s} \rceil$ and 0 otherwise.

Since any ϕ-relative option is a complete policy for the block MDP of $\bar{s}$, we will use $d^o_{\bar{s}}$ to denote the state occupancy measure, as defined in eq. (2.18), computed from policy π_o in $\mathbf{M}_{\bar{s}}$. State occupancies will often be computed at exit states. In fact, the abstraction should only reflect the "external" behaviour of some option, not the path in specific ground states within the block. Thus, we define block occupancies as a simple marginalisation over blocks.

Definition 4.2. For every MDP $\mathbf{M}$, surjective function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$, we define the *block occupancy* measure of $\bar{s} \in \bar{\mathcal{S}}$ and option $o \in \Omega_{\bar{s}}$ at some $s \in \lfloor \bar{s} \rceil$, as the distribution over next blocks: $h^o_{\bar{s}}(s) \in \Delta(\bar{\mathcal{S}} \cup \{s_\perp\})$, with $h^o_{\bar{s}}(\bar{s}' \mid s) := \sum_{s' \in \lfloor \bar{s}' \rceil} d^o_{\bar{s}}(s' \mid s)$, if $\bar{s}' \neq s_\perp$, and $h^o_{\bar{s}}(s_\perp \mid s) := d^o_{\bar{s}}(s_\perp \mid s)$, otherwise.

We emphasize the role of the sink state $s_\perp$. Although some options might spend multiple steps in $\lfloor \bar{s} \rceil$, each exit state is always visited at most once along any trajectory. This is thanks to the sink $s_\perp$, as it will always be reached in the next step. The block occupancy measure will play a major role in defining the abstract transition probabilities. The abstract reward, instead, will be put in relation to the total return accumulated within the block before reaching the exit states. This is exactly captured by $V^o_{\bar{s}}(s)$, that is, the value of the option o in the block MDP of $\bar{s}$.

[1] We recall that $\delta_{s_\perp}$ is the deterministic distribution. For discrete sets, $\delta_{s_\perp}(s) = \mathbb{I}(s = s_\perp)$.

These two elements, $h^o_{\bar{s}}$ and $V^o_{\bar{s}}$, that are relative to the ground MDP, will be related to analogous quantities in the abstraction. Specifically, this work identifies that the block occupancy should be compared with the discounted probability of entering each abstract state. Similarly, the block value should be related to the total reward accumulated in the corresponding abstract state. Expanding these terms for 2-MDPs, we define the respective target values as follows. For each $\bar{s}_p, \bar{s}, \bar{s}' \in \bar{\mathcal{S}}$ and $\bar{a} \in \bar{\mathcal{A}}$, with $\bar{s}_p \neq \bar{s}$, these are:

$$\tilde{h}_{\bar{s}_p\bar{s}\bar{a}}(\bar{s}') := (1-\gamma)(\bar{\gamma}\,\bar{T}(\bar{s}' \mid \bar{s}_p\bar{s}\bar{a}) + \bar{\gamma}^2\,\bar{T}_{\bar{s}_p\bar{s}\bar{a}}\,\bar{T}(\bar{s}' \mid \bar{s}\bar{s}\bar{a})) \tag{4.5}$$

$$\tilde{V}_{\bar{s}_p\bar{s}\bar{a}} := \bar{R}(\bar{s}_p\bar{s}\bar{a}) + \bar{\gamma}\,\bar{T}_{\bar{s}_p\bar{s}\bar{a}}\,\bar{R}(\bar{s}\bar{s}\bar{a}) \tag{4.6}$$

$$\text{with } \bar{T}_{\bar{s}_p\bar{s}\bar{a}} = \frac{\bar{T}(\bar{s} \mid \bar{s}_p\bar{s}\bar{a})}{1-\bar{\gamma}\,\bar{T}(\bar{s} \mid \bar{s}\bar{s}\bar{a})}$$

Their structure is mainly motivated by the fact that these expressions sum all the transition probabilities and rewards, accumulated over an indefinite number of self-loops in $\bar{s}$.

Realizable Abstractions Using the concepts above, we are finally ready to provide a complete description of our MDP abstractions. We say that an abstract action is *realizable* if the behaviour described by the abstract transitions and rewards can be replicated (realized) in the ground MDP. More precisely, realizable actions can be associated to options in the lower MDP which are associated to the required occupancy measure and value.

Definition 4.3. Given an MDP $\mathbf{M}$ and an abstraction $\langle\bar{\mathbf{M}}, \phi\rangle$, an abstract tuple $(\bar{s}_p\bar{s}\bar{a})$, with $\bar{s}_p \neq \bar{s}$, is said (α, β)-*realizable* if there exists a ϕ-relative option $o \in \Omega_{\bar{s}_p\bar{s}}$, such that

$$\tilde{h}_{\bar{s}_p\bar{s}\bar{a}}(\bar{s}') - h^o_{\bar{s}}(\bar{s}' \mid s) \leq \alpha \tag{4.7}$$

$$(1-\gamma)(\tilde{V}_{\bar{s}_p\bar{s}\bar{a}} - V^o_{\bar{s}}(s)) \leq \beta \tag{4.8}$$

for all $\bar{s}' \neq \bar{s}$ and $s \in \mathcal{E}_{\bar{s}_p\bar{s}}$. The option o is called the realization of $(\bar{s}_p\bar{s}\bar{a})$. An abstraction $\langle\bar{\mathbf{M}}, \phi\rangle$ is said (α, β)-realizable in $\mathbf{M}$ if any $(\bar{s}_p\bar{s}\bar{a}) \in (\bar{\mathcal{S}} \cup \{\bar{s}_\circ\}) \times \bar{\mathcal{S}} \times \bar{\mathcal{A}}$ with $\bar{s}_p \neq \bar{s}$ also is. A $(0,0)$-realizable abstraction is *perfectly realizable*.

This definition essentially requires that the desired block occupancy and value, computed from the abstraction, should be similar to the ones that are possible in the ground MDP, from each entry state. This can be interpreted as requiring that the abstraction should not be too optimistic about the transitions and values that are possible in the ground decision process. Thanks to these requirements, we will be able to show that abstract policies for realizable abstractions can be translated

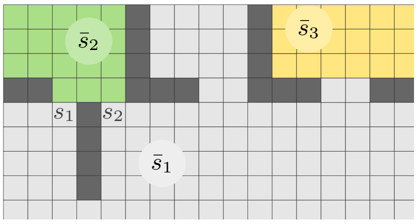


Figure 4.2. If $\bar{T}(\bar{s}_3 \mid \bar{s}_2\bar{s}_1\bar{a}_3)$ is high, relatively to γ and α, the tuple $(\bar{s}_2\bar{s}_1\bar{a}_3)$ becomes not realizable.

into near-optimal policies in the ground MDP. Also, in case the abstraction is an MDP, eqs. (4.5) and (4.6) and the above constraints simplify. Finally, we note that the scale factor of $(1-\gamma)$ was only added to eq. (4.8) to obtain two parameters in the same range: $\alpha, \beta \in [0, 1]$.

Any two given α and β put a restriction on the possible mapping functions. In fact, some partitions of the state space may not admit any satisfying 2-MDP over the induced abstract states, especially if some entry state can achieve very different transition probabilities or rewards. This is the case for the partition of fig. 4.2, if we assume homogeneous probabilities in the 2D plane. In particular, if $\bar{T}(\bar{s}_3 \mid \bar{s}_2\bar{s}_1\bar{a}_3)$ is relatively high, say 0.95, then the state-action $(\bar{s}_2\bar{s}_1\bar{a}_3)$ will likely not be realizable in the grid-world, for small α, β. In fact, this would imply that, for some $o \in \Omega_{\bar{s}_2\bar{s}_1}$, $h^o_{\bar{s}_1}(\bar{s}_3 \mid s_1)$ should be similar to $h^o_{\bar{s}_1}(\bar{s}_3 \mid s_2)$. For sensible values of γ, this is unlikely to be true, due to the many more steps required to reach the yellow block when starting in s_1 instead of s_2. Similar counterexamples can be constructed for rewards, for example, by making only the vertical path from s_1 very rewarding. As we can see, realizability purposefully captures ideas that were already present in the HRL literature. In particular, when the agent is in $\lfloor \bar{s}_2 \rfloor$, the value of reaching $\lfloor \bar{s}_1 \rfloor$ should be treated atomically, regardless of the specific entry state in $\mathcal{E}_{\bar{s}_2\bar{s}_1}$ that is reached.

4.4 Properties

The previous section has defined realizable MDP abstractions and described their meaning. However, the HRL literature already includes many notions of abstractions. A new definition is only valuable if it is applicable, which we have discussed already, and if it possesses strong properties. In this section, we show that any abstract policy for a realizable abstraction can be transformed into a ground near-optimal policy, in a purely compositional way.

If some abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$ is (α, β)-realizable, then it is possible to associate to each $(\bar{s}_p\bar{s}\bar{a}) \in \bar{\mathcal{S}}^2\bar{\mathcal{A}}$ its realizing option. This means that, for every deterministic abstract policy $\bar{\pi}$, there exists some policy of options $\Omega_{\bar{\pi}}$, where, for each $\bar{s}_p\bar{s} \in \bar{\mathcal{S}}^2$, there is an associated option $o \in \Omega_{\bar{\pi}} \cap \Omega_{\bar{s}_p \bar{s}}$ that is a realization of $(\bar{s}_p \bar{s}\, \bar{\pi}(\bar{s}_p \bar{s}))$. We say that $\Omega_{\bar{\pi}}$ is a *realization* of $\bar{\pi}$. We can finally state the following result.

Theorem 4.1. *Let* $\langle \bar{\mathbf{M}}, \phi \rangle$ *be an* (α, β)*-realizable abstraction of an MDP* $\mathbf{M}$*, whose initial distributions satisfy* $\max_{\bar{s}} |\bar{\mu}(\bar{s}) - \sum_{s \in \lfloor \bar{s} \rfloor} \mu(s)| \leq \xi$*. Then, if* Ω' *is the realization of some deterministic abstract policy* $\bar{\pi}$*,*

$$\bar{V}^{\bar{\pi}}_{\bar{\mu}} - V^{\Omega'}_{\mu} \leq \frac{\beta}{(1-\gamma)^2} + \frac{\alpha\, |\bar{\mathcal{S}}|}{(1-\gamma)^2(1-\bar{\gamma})} + \frac{\xi\, |\bar{\mathcal{S}}|}{1-\bar{\gamma}} \tag{4.9}$$

Proof. See page 85. □

As always, all the proofs and necessary lemmas are in a dedicated section (4.7, for this chapter). As we can see, the difference on the left is between the value of $\bar{\pi}$ in the abstract $\bar{\mathbf{M}}$, and the value of the realization Ω for the ground $\mathbf{M}$. So, the value loss is computed between the two decision processes. Essentially, this theorem proves that (α, β)-realizability is enough to have strong guarantees on the minimal value achieved by the realization. This result is generally applicable and of independent interest. Also, if $\bar{\pi}$ is the optimal policy of $\bar{\mathbf{M}}$, we can obtain a near-optimality guarantee for its realization. For this second result, we take inspiration from the role of heuristics in classical planning and say that "admissible" abstractions should be optimistic models of the ground MDP.

Definition 4.4. An abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$ of an MDP $\mathbf{M}$ is *admissible* if $\bar{V}^* \geq V^*$.

Since admissible abstractions are optimistic, an admissible and (α, β)-realizable abstraction can be interpreted as being optimistic by a bounded margin. We obtain the following result.

Corollary 4.2. *The realization of the optimal policy of any admissible and* (α, β)*-realizable abstraction is* ε*-optimal, for*

$$\varepsilon = \frac{\beta}{(1-\gamma)^2} + \frac{\alpha\, |\bar{\mathcal{S}}|}{(1-\gamma)^2(1-\bar{\gamma})} + \frac{\xi\, |\bar{\mathcal{S}}|}{1-\bar{\gamma}} \tag{4.10}$$

Proof. If Ω^* is the realization of $\bar{\pi}^*$, the result follows from $V^* \leq \bar{V}^* \leq V^{\Omega^*} + \varepsilon$. □

To fully appreciate the importance of theorem 4.1 and corollary 4.2, we need to remember that realizations can be computed in a compositional way. In other words, there is no global optimisation involved in computing the ground policy of options. In fact, for an option to be part of the realization it is sufficient that it satisfies the constraints in eqs. (4.7) and (4.8). Since these constraints are expressed in terms of the block MDP, their computation is completely independent of all the other blocks and options. In this light, such near-optimality is the cost to be paid for finding a policy as a union of shorter policies, without further computation.

To evaluate the scale of the bound in eq. (4.9), we recall that α and β are in $[0, 1]$, and that $\bar{\gamma} \leq \gamma$. Importantly, the number of abstract states $|\bar{\mathcal{S}}|$ at the numerator is always a finite number, and the size of the ground state space, which is usually very large or infinite, does not appear.

Realizable abstractions are flexible representations because they can encode a variable amount of information content. If the ground domain is a goal MDP, we might consider two extremes. One abstraction can be composed by just two states: a goal and a non-goal state. Although this might be an admissible and perfectly realizable abstraction, for a suitable transition function, it provides no information content. In fact, realizing the only option would be as complex as solving the entire MDP. On the other extreme, any MDP $\mathbf{M}$ can be abstracted by itself as $\langle \mathbf{M}, \mathrm{I} \rangle$, where $\mathrm{I} : x \mapsto x$ is the identity function. However, many intermediate abstractions are also possible, although these will often not be perfectly realizable.

Proposition 4.3. *Any MDP* $\mathbf{M}$ *admits* $\langle \mathbf{M}, \mathrm{I} \rangle$ *as an admissible and perfectly realizable abstraction.*

Proof. See page 79. □

Although our abstractions are able to represent compressions along the time dimension, they are not restricted to those. In fact, as a special case, they can capture any MDP homomorphism in which compression only takes place with respect to parallel symmetries of the states, without impacting the effective horizon.

Proposition 4.4. *If* $\langle f, \{g_s\}_{s \in \mathcal{S}} \rangle$ *is an MDP homomorphism from* $\mathbf{M}$ *and* $\bar{\mathbf{M}}$*, then* $\langle \bar{\mathbf{M}}, f \rangle$ *is an admissible and perfectly realizable abstraction of* $\mathbf{M}$*.*

Proof. See page 80. □

The Value of Policies of Options We conclude this section by studying the value of any policy of options in the ground MDP. What we show here is that for generic state partitions and MDPs, the value of any policy of options can be expressed as a composition of its options. Somehow, these results do talk about realizable abstractions directly, since no abstract dynamics is involved and only ground MDP and the partition induced by the mapping function appear. However, they are preparatory to them. We delay minor lemmas to the proofs' section and only report proposition 4.5 among these intermediate results. Proposition 4.6, instead, is an independent result. What it shows is that different policies can be compared in terms of their respective behaviours in each block MDP. This is an original way to compare the value of generic policies and is applicable even outside of HRL.

The results that follow are stated for deterministic options, for simplicity, but they can be readily generalised to stochastic options by an appropriate expectation over the output action. First, we express the value of a single option as follows.

Proposition 4.5. *Consider any MDP* $\mathbf{M}$ *and surjective function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$*. Then, from any state* $s \in \mathcal{S}$*, the value of any deterministic* ϕ*-relative option* $o \in \Omega_{\phi(s)}$ *and policy* π *is*

$$Q^\pi(s, o) = \sum_{s' \in \mathcal{S}} \frac{d^o_{\phi(s)}(s' \mid s)}{1 - \gamma} \left(\mathbb{I}(s' \in \lfloor \phi(s) \rfloor)\, R(s'\, o(s')) + \mathbb{I}(s' \in \mathcal{X}_{\phi(s)})\, V^\pi(s')\right) \tag{4.11}$$

where $d^o_{\phi(s)}$ *is the state occupancy measure of* π_o *in the block-restricted MDP* $\mathbf{M}_{\phi(s)}$.

Proof. See page 81. □

As we can see, the contribution of a single ϕ-relative option to the total value is made up of two parts. The first half is the expected return accumulated within the block. For generic stochastic options, if $d^o_{\mathsf{sa},\phi(s)}$ is the state-action occupancy distribution of o in $\mathbf{M}_{\phi(s)}$, then, the first half of the expression becomes $V^o_{\phi(s)}$ or, equivalently, $(1-\gamma)^{-1} \langle d^o_{\mathsf{sa},\phi(s)}(s), R_{\phi(s)} \rangle$. The second term, instead, only comprehends the values that are found when leaving the current block in the exit states. Which exits are visited, and how frequently, is still dictated by $d^o_{\phi(s)}(s)$.

As a final result for this section, we show that entire policies can be compared using the formalism of relative options. In fact, given any mapping function, or any partition of $\mathcal{S}$, any policy is equivalent to a unique policy of options whose second component of the initiation sets corresponds to the block in the partition. Therefore, the following result is true for generic policies and MDPs.

Proposition 4.6. *Given a state partition, let* Ω_1*,* Ω_2 *be two deterministic policies of options in* $\mathbf{M}$ *over that partition. For each* $\bar{s} \in \bar{\mathcal{S}}$ *and* $o_1 \in \Omega_{\bar{s}} \cap \Omega_1, o_2 \in \Omega_{\bar{s}} \cap \Omega_2$*, assume that* $\|d^{o_1}_{\bar{s}} - d^{o_2}_{\bar{s}}\|_1 \leq \alpha$ *and* $\max_{s \in \lfloor \bar{s} \rceil} |R(s\, o_1(s)) - R(s\, o_2(s))| \leq \beta$*. Then, for any state* $s \in \mathcal{S}$*,*

$$|V^{\Omega_1}(s) - V^{\Omega_2}(s)| \leq \frac{\alpha + \beta}{(1 - \gamma)^2} + \frac{\alpha}{(1 - \gamma)^3} \tag{4.12}$$

Proof. See page 82. □

This result can be directly compared with Abel (2020, Theorem 1, eq. (4)), since it computes the same value difference as we do here, in presence of ϕ-relative options and state partitions. The improvement we have here is significant, especially because our result does not involve the cardinality of the ground state space $|\mathcal{S}|$. This is critical since we would like to apply abstractions in MDP with large or infinite state spaces.

4.5 Abstracting and Realizing

As we saw in the previous sections, realizable abstractions have very desirable properties for HRL, including compositionality and near-optimality. Following the natural progress of this work, this section aims to answer two questions: how can realizable abstraction be found, and later, used in RL? The first question goes from the ground MDP to the high-level and can be seen as "*abstracting*" the low-level domain. The second process, instead, goes from the high-level to the ground MDP and, thanks to the properties we have identified, we know that it can be solved by "*realizing*" abstract states and actions into ground options. Targeting these two questions will also reveal further insights that are widely applicable in the HRL literature, such as identifying when it is feasible to reduce the planning horizon with abstractions.

Realizability of definition 4.3 quantifies and constrains for each entry state $s \in \mathcal{E}_{\bar{s}_p\bar{s}}$. We now define a slight relaxation where we consider some initial distribution over the entry states, $\mu_{\bar{s}_p\bar{s}} \in \Delta(\mathcal{E}_{\bar{s}_p\bar{s}})$. Depending on the specific approach or algorithm, $\mu_{\bar{s}_p\bar{s}}$ is usually determined by the global policy that was active in $\lfloor \bar{s}_p \rfloor$. This distribution allows us to marginalise with respect to initial states and only express realizability over blocks. This is the notion that will be abstracted or realized in the following text.

Definition 4.5. Given an MDP $\mathbf{M}$ and an abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$, an abstract tuple $(\bar{s}_p\bar{s}\bar{a})$, with $\bar{s}_p \neq \bar{s}$, is (α, β)-*realizable from* a distribution $\nu \in \Delta(\mathcal{E}_{\bar{s}_p\bar{s}})$, if there exists a ϕ-relative option $o \in \Omega_{\bar{s}_p\bar{s}}$, such that

$$\tilde{h}_{\bar{s}_p\bar{s}\bar{a}}(\bar{s}') - h^o_\nu(\bar{s}') \leq \alpha \tag{4.13}$$

$$(1-\gamma)(\tilde{V}_{\bar{s}_p\bar{s}\bar{a}} - V^o_\nu) \leq \beta \tag{4.14}$$

for all $\bar{s}' \neq \bar{s}$, where $h^o_\nu(\bar{s}') := \sum_s h^o_{\bar{s}}(\bar{s}' \mid s)\, \nu(s)$ and $V^o_\nu := \sum_s V^o_{\bar{s}}(s)\, \nu(s)$. The option o is the realization of $(\bar{s}_p\bar{s}\bar{a})$ from ν.

Being a relaxed definition, every realization is also a realization from any distribution, but not vice versa. Note that, due to marginalization, the terms h^o_ν and V^o_ν are no longer dependent on the ground states. This means that realizability from distribution consists exactly of $|\bar{\mathcal{S}}|$ constraints, where $|\bar{\mathcal{S}}| - 1$ comes from the block occupancies in eq. (4.13) and one from the value in eq. (4.14). This will be the notion of realizability to be abstracted and realized.

Abstracting

Learning realizable abstractions from online experience is a very interesting research direction, but it remains outside the scope of this thesis. Instead, what we will discuss here is how to identify a suitable abstract transition and reward dynamics when the ground values and occupancies are known or their estimates are. Here we assume that ground MDP $\mathbf{M}$, abstract states $\bar{\mathcal{S}}$ and actions $\bar{\mathcal{A}}$, and a mapping function $\phi : \mathcal{S} \to \bar{\mathcal{S}}$ are all given. For each block, we aim to find abstract transitions, rewards, and discount factor, such that the constructed abstraction is realizable in the ground domain. Specifically, consider a single abstract state-action tuple $(\bar{s}_p \bar{s} \bar{a})$, with $\bar{s}_p \neq \bar{s}$. We assume that some entry distribution $\nu \in \Delta(\mathcal{E}_{\bar{s}_p \bar{s}})$ and an option $o \in \Omega_{\bar{s}_p \bar{s}}$ for the relevant block are given. Then, we want to set the local probabilities for the abstract transitions and rewards such as o is a realization. If h^o_ν and V^o_ν are the relevant quantities as in definition 4.5, then algorithm 4.1 reaches the objective. The algorithm also receives the abstract MDP in input. Although its initialisation is irrelevant, passing $\bar{\mathbf{M}}$ in successive iterations allows us to preserve the partial assignment computed in previous runs for different state-action tuples. The correctness of this algorithm is stated below.

Algorithm 4.1: AbstractOne

```
Input: Abstract MDP $\bar{\mathbf{M}}$, tuple $(\bar{s}_p \bar{s} \bar{a})$
Input: Target block occupancy $h^o_\nu$, target value $V^o_\nu$
Output: Updated abstract MDP
$\bar{\mathbf{M}}.\bar{T}(\bar{s}' \mid \bar{s}_p \bar{s}\, \bar{a}) \leftarrow \frac{h^o_\nu(\bar{s}')}{(1-\gamma)\bar{\gamma}}$  for each  $\bar{s}' \neq \bar{s}$
$\bar{\mathbf{M}}.\bar{T}(\bar{s} \mid \bar{s}_p \bar{s}\, \bar{a}) \leftarrow 1 - \sum_{\bar{s}'' \neq \bar{s}} \bar{T}(\bar{s}'' \mid \bar{s}_p \bar{s}\, \bar{a})$
$\bar{\mathbf{M}}.\bar{R}(\bar{s}_p \bar{s}\, \bar{a}) \leftarrow \min\{1, V^o_\nu\}$
$\bar{\mathbf{M}}.\bar{R}(\bar{s} \bar{s}\, \bar{a}) \leftarrow \max\{0, V^o_\nu - 1\} \left( \frac{h^o_\nu(\bar{s})}{1-\bar{\gamma}} - 1 \right)^{-1}$
$\bar{\mathbf{M}}.\bar{T}(\bar{s}' \mid \bar{s} \bar{s}\, \bar{a}) \leftarrow 0$  for each  $\bar{s}' \neq \bar{s}$
$\bar{\mathbf{M}}.\bar{T}(\bar{s} \mid \bar{s} \bar{s}\, \bar{a}) \leftarrow 1$
foreach $\bar{s}'_p \in \bar{\mathcal{S}} \setminus \{\bar{s}_p, \bar{s}\}$ do
    $\bar{\mathbf{M}}.\bar{T}(\bar{s}' \mid \bar{s}'_p \bar{s} \bar{a}) \leftarrow \frac{\tilde{h}_{\bar{s}'_p \bar{s} \bar{a}}(\bar{s}')}{(1-\gamma)\bar{\gamma}}$  for each  $\bar{s}' \neq \bar{s}$
    $\bar{\mathbf{M}}.\bar{T}(\bar{s} \mid \bar{s}'_p \bar{s}\, \bar{a}) \leftarrow 1 - \sum_{\bar{s}'' \neq \bar{s}} \bar{T}(\bar{s}'' \mid \bar{s}'_p \bar{s}\, \bar{a})$
    $\bar{\mathbf{M}}.\bar{R}(\bar{s}'_p \bar{s}\, \bar{a}) \leftarrow \min\{1, \tilde{V}_{\bar{s}'_p \bar{s} \bar{a}}\}$
end
return $\bar{\mathbf{M}}$
```

Proposition 4.7. *Consider an MDP $\mathbf{M}$, an abstraction $\langle \bar{\mathbf{M}}, \phi \rangle$, any tuple $(\bar{s}_p \bar{s} \bar{a})$ with $\bar{s}_p \neq \bar{s}$, distribution $\nu \in \Delta(\mathcal{E}_{\bar{s}_p \bar{s}})$, option $o \in \Omega_{\bar{s}_p \bar{s}}$, with associated targets h^o_ν, V^o_ν. Then, under assumption 4.1, the output of* AbstractOne$(\bar{\mathbf{M}}, (\bar{s}_p \bar{s} \bar{a}), h^o_\nu, V^o_\nu)$ *is a* *valid 2-MDP and the option o is a perfect realization of $(\bar{s}_p\bar{s}\bar{a})$ from ν.*

Proof. See page 87. □

Assumption 4.1. Given an MDP $\mathbf{M}$, and an abstraction $\langle\bar{\mathbf{M}}, \phi\rangle$, the abstract discount factor $\bar{\gamma}$ must satisfy, for each $\bar{s}_p, \bar{s} \in \bar{\mathcal{S}}$, option $o \in \Omega_{\bar{s}_p\bar{s}}$, and $s \in \mathcal{E}_{\bar{s}_p\bar{s}}$,

$$h^o_{\bar{s}}(\bar{s} \mid s) \geq 1 - \bar{\gamma} \tag{4.15}$$

$$V^o_{\bar{s}} \leq h^o_{\bar{s}}(\bar{s} \mid s)/(1 - \bar{\gamma}) \tag{4.16}$$

Proposition 4.7 says that the output decision process is a well-formed 2-MDP, and $(\bar{s}_p\bar{s}\bar{a})$ is $(0, 0)$-realizable from ν after the update of $\bar{\mathbf{M}}$. This function is correct if the abstract discount factor satisfies assumption 4.1, which is an interesting fact. Specifically, assumption 4.1 is the condition needed to guarantee a well-formed 2-MDP after the assignment. We give an interpretation of this condition below because, beyond our specific algorithm, this assumption provides more general insights about abstract horizons in HRL.

Assumption 4.1 can be compactly expressed with a single inequality: $h^o_{\bar{s}}(\bar{s} \mid s) \geq (1 - \bar{\gamma}) \max\{1, V^o_{\bar{s}}\}$. However, we want to emphasise that this is actually composed of two independent parts, eq. (4.15) which only constrains occupancy and eq. (4.16) which also involves value. The first says that $\bar{\gamma}$ can only be low if the occupancy in every block is high. In particular, if there exists an option o that leaves some $\lfloor\bar{s}\rfloor$ in one step, then $h^o_{\bar{s}}(\bar{s}' \mid s) = 1 - \gamma$, and eq. (4.15) is only satisfied for $\bar{\gamma} = \gamma$. The second says that $\bar{\gamma}$ can only be low if $V^o_{\bar{s}}$ is also low with respect to $h^o_{\bar{s}}$. In particular, if there exists an option o that collects in $\lfloor\bar{s}\rfloor$ a reward of 1 at each step, then $V^o_{\bar{s}} = h^o_{\bar{s}}(\bar{s} \mid s)/(1 - \gamma)$, and eq. (4.16) is only satisfied for $\bar{\gamma} = \gamma$. This allows us to conclude that a time compression in the abstraction is possible if and only if: (i) the changes between blocks occur at some lower timescale; (ii) rewards are temporally sparse. This confirms some common intuitive understanding of the role of $\bar{\gamma}$ in the HRL literature. In addition, it confirms that sparse rewards are also important and which is the exact relation to satisfy. Note, in fact, that if either of the conditions above are not verified, no time compression is possible. Importantly, $\bar{\gamma} \coloneqq \gamma$ is always a feasible choice.

The assignment in algorithm 4.1 is only one of many ways to construct consistent abstractions. Regardless of the specific assignment, the existence of abstract states introduces inevitable interactions between the transition probabilities of the tuples $(\bar{s}_p\bar{s})$ and $(\bar{s}'_p\bar{s})$. Choosing 2-MDPs instead of MDP strongly alleviated unwanted interactions between different options starting from different blocks. However, some degree of interaction is at the heart of the abstraction process, and it ultimately translates into a constraint for the mapping function ϕ. As a special case, we reconsider fig. 3.3, which has a non-trivial compression of the state space. Since this is a goal MDP, with null rewards in non-goal states, the transition probabilities of each $(\bar{s}_p\bar{s})$ can be independently set to match the desired options' occupancy in eq. (4.7). More precisely, we can say that for any MDP $\mathbf{M}$ whose ground transition function matches the support shown in the drawing of fig. 3.3 (top), there exists a $\langle\bar{\mathbf{M}}, \phi\rangle$ that is perfectly realizable in $\mathbf{M}$, with ϕ matching the block colours.

Realizing

In this conclusive subsection, we study the opposite problem, that is, how to find realizations of abstract actions. If some realizable abstraction for $\mathbf{M}$ is given, our objective is to find a ground policy of options Ω' that is the realization of each abstract state and action of $\bar{\mathbf{M}}$. This is the missing step to allow future researchers to develop efficient RL algorithms for MDPs in the presence of realizable abstractions. Similarly to the "Abstracting" subsection of the previous page, we do not aim to define a single algorithm here. Rather, we will identify the core principles and two generic templates that can be integrated in more complete learning algorithms. Specifically, instead of seeking to realize a complete abstraction, which would translate to a policy of options, we study how a single tuple $(\bar{s}_p\bar{s}\bar{a})$ can be realized as some option $o \in \Omega_{\bar{s}_p\bar{s}}$. From theorem 4.1, we know that groups of these options are sufficient to achieve near-optimal behaviour. Also, as a simplifying choice, instead of seeking the generic realizing options of definition 4.3, we assume some entry distribution $\nu \in \Delta(\mathcal{E}_{\bar{s}_p\bar{s}})$ is given, and we will optimize for realizing options of $(\bar{s}_p\bar{s}\bar{a})$ from ν as defined in definition 4.5.

Let us write cardinalities with uppercase letters, such as $\bar{S} := |\bar{\mathcal{S}}|$. We observe that realizability from initial distributions introduces $\bar{S}-1$ constraints from eq. (4.13) and 1 constraint from eq. (4.14), for a total of $\bar{S}$ constraints for each $(\bar{s}_p\bar{s}\bar{a}) \in \bar{\mathcal{S}}^2\bar{\mathcal{A}}$ with $\bar{s}_p \neq \bar{s}$. Since each $(\bar{s}_p\bar{s}\bar{a})$ comes with $\bar{S}$ transition probabilities and 1 reward, for a total of $\bar{S}$ degrees of freedom, the problem seems to be well constrained. However, note that no constraints are set for the tuples $(\bar{s}\bar{s}\bar{a})$.

In the following, we illustrate the principles for two solution methods: constrained MDP solutions and primal-dual approaches.

Constrained MDPs (Ross 1985; Altman 1999) Constrained MDPs (CMDPs) are an extension of classic MDPs and their associated RL problem to constrained optimisation problems. The original value maximisation problem of RL can be regarded as unconstrained, because the agent could choose any policy in Π. In a CMDP, the feasible set of solutions is restricted to some subset of policies $\Pi_{\mathsf{c}} \subseteq \Pi$. Then, the solution of a CMDP is $\arg\max_{\pi\in\Pi_{\mathsf{c}}} V_\mu^\pi$, as policy maximisation, restricted to Π_{c}. To define the feasible set, CMDPs augment MDPs with auxiliary reward functions $R_1, \dots, R_c$ and minimum associated values $v_1, \dots, v_c$. The feasible set of policies Π_{c} is defined as the union of all policies whose value with respect to the reward function R_i is at least v_i, for each $i = 1, \dots, c$. Using an equivalent formulation, most CMDP papers prefer to formulate constraints with cost function and maximum costs. The two formalisms are equivalent. Unlike standard RL, CMDPs allow encoding of both soft and hard constraints. This field has received attention because of its relevance for RL safety issues and the encoding of hard constraints in safety-critical systems. Constrained RL is a general framework. Some general solution algorithms are already available (Achiam, Held, et al. 2017; Y. Zhang, Vuong, et al. 2020), and more are expected to be developed in the future due to its relevance. By expressing the realizability problem as a CMDP, we do not restrict ourselves to a specific technique. Rather, we could realize abstract actions with any algorithm that may be developed in the future for constrained RL. This is especially relevant since the ground MDP is expected to be non-tabular, in general, thus preventing the application of the most standard techniques.

Among all $\bar{S}$ constraints, we choose to represent the $\bar{S} - 1$ inequalities of (4.13) over transition probabilities as explicit hard constraints and the single inequality of eq. (4.14) as a soft constraint. Since we assumed that $(\bar{s}_p \bar{s} \bar{a})$ is indeed (α, β)-realizable, there will be at least one option $o^* \in \Omega_{\bar{s}_p \bar{s}}$, obtained as the maximization of V_ν^o, which satisfies all the $\bar{S}$ original constraints. Specifically, we rewrite the hard constraints as:

$$h_\nu^o(\bar{s}') \geq \tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha \tag{4.17}$$

$$\frac{1}{1-\gamma} \sum_{s \in \mathcal{S}} h_{\bar{s}}^o(\bar{s}' \mid s)\, \nu(s) \geq \frac{\tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha}{1-\gamma} \tag{4.18}$$

$$\frac{1}{1-\gamma} \sum_{s, s' \in \mathcal{S}} d_{\bar{s}}^o(s' \mid s)\, \mathbb{I}(s' \in \lfloor \bar{s}' \rfloor)\, \nu(s) \geq \frac{\tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha}{1-\gamma} \tag{4.19}$$

$$V_{\nu, \bar{s}'}^o \geq \frac{\tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha}{1-\gamma} \tag{4.20}$$

where $V_{\nu, \bar{s}'}^o$ is the value function of o in the block MDP $\mathbf{M}_{\bar{s}}$ with reward function $\mathbb{I}(s' \in \lfloor \bar{s}' \rfloor)$ instead of $R_{\bar{s}}$. In summary, we have just reformulated the problem of realizing any tuple $(\bar{s}_p \bar{s} \bar{a})$ in some MDP $\mathbf{M}$ as the problem of solving the following CMDP:

$$\arg\max_{\pi \in \Pi} V_{\bar{s}}^\pi \qquad s.t. \quad V_{\nu, \bar{s}'}^\pi \geq \frac{\tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha}{1-\gamma} \tag{4.21}$$

In other words, the auxiliary reward functions are $R_i(s\, a) \coloneqq \mathbb{I}(s' \in \lfloor \bar{s}' \rfloor)$ with limits $v_i \coloneqq (\tilde{h}_{\bar{s}_p \bar{s} \bar{a}}(\bar{s}') - \alpha)/(1-\gamma)$. The output is a policy for the block MDP $\mathbf{M}_{\bar{s}}$, which can be equivalently seen as a ϕ-relative option for the full MDP $\mathbf{M}$.

LP Formulation For this second solution approach, we show that the realizability problem can be formulated as a linear program, and solved using primal-dual techniques. This may come as little surprise, since the Lagrangian formulation is one of the possible solution methods for constrained optimisation problems such as CMDPs. However, we present these two techniques separately, since some CMDP methods may be more closely related to Deep RL algorithms, and they can appear quite different from online stochastic optimisation algorithms for linear programs. An example of this second direction is Y. Zhang, Vuong, et al. (2020). Similarly, primal-dual techniques have also been developed independently of CMDPs, and they are often presented as solution methods for unconstrained RL.

The linear programming (LP) formulation of optimal planning in MDPs dates back to Bertsekas (1995) and, later Puterman (1994). Planning with the tabular formulation was already known. Later research focused on finding optimal policies for non-tabular MDPs, in presence of generative simulators or online, (de Farias and Roy 2003; Mahadevan, B. Liu, et al. 2014; Y. Chen and M. Wang 2016; Tiapkin and Gasnikov 2022; Gabbianelli, Neu, et al. 2023; Neu and Okolo 2023). Currently, research efforts focus on reaching all these objectives in the more demanding online setting. Similarly to the CMDP formulation above, the linear program we propose here may be solved with any feasible algorithm for this setting.

For a generic MDP, the classic LP formulation of optimal values can be compactly expressed by using the vector notation for the reward function $R \in \mathbb{R}^{SA}$, the initial distribution $\nu \in \mathbb{R}^{S}$, and other matrices which will be introduced next. The matrix that copies elements for each action is $E \in \mathbb{R}^{SA \times S}$ with $E(sa, s') \coloneqq \mathbb{I}(s = s')$. Transitions are also written as a matrix $P \in \mathbb{R}^{SA \times S}$ where $P(sa, s') \coloneqq T(s' \mid sa)$. Consider the following linear program:

$$\begin{aligned} \max_{b \in \mathbb{R}^{SA} :\, b \geq 0} \quad & b^T R \\ \text{s.t.} \quad & E^T b - \gamma\, P^T b = (1 - \gamma)\, \nu \end{aligned} \tag{4.22}$$

The constraint expressed here in vector notation is the Bellman flow equation on the state-action occupancy distribution b^*. At the optimum, the solution b^* is the discounted state-action occupancy measure of the optimal policy, and we have $E^T b^* = d_\nu^{\pi^*}$. In addition, the objective is the scaled optimal value $V^* = \langle b^*, R \rangle / (1 - \gamma)$. The dual linear program is

$$\begin{aligned} \min_{V \in \mathbb{R}^{S}} \quad & (1 - \gamma)\, \mu^T V \\ \text{s.t.} \quad & E\, V - \gamma\, P\, V \geq R \end{aligned} \tag{4.23}$$

and the optimum of this problem is V^*, the value of the optimal policy. Solving either the primal or the dual problem is equivalent to solving the given MDP. The references cited above are only some of the works that adopt this linear formulation to find the optimal policy. For generalising to non-tabular MDPs, the linear formulation is often expressed in feature space. Here we keep the tabular equations for simplicity.

The LP formulation above will be now applied for each block MDP and modified to introduce additional constraints. Similarly to our choice for CMDPs, we only express the constraint on occupancy distributions. Due to the equality constraint, the vector b is forced to be a state-action occupancy distribution. Thus, all $\bar{S}-1$ constraints (4.13) can be written in the primal program as $B^T b \geq \tilde{h}_{\bar{s}_p \bar{s}\bar{a}} - \alpha$, where $B^T \in \mathbb{R}^{SA\times(\bar{S}-1)}$ is the matrix that sums all occupancies across states and actions for one block as $B^T(\bar{s}, sa) \coloneqq \mathbb{I}(\bar{s} = \phi(s))$. The linear program becomes

$$\begin{aligned} \max_{b\in\mathbb{R}^{SA}:\, b\geq 0} \quad & b^T R \\ \text{s.t.} \quad & E^T b - \gamma\, P^T b = (1-\gamma)\,\mu \\ & -B^T b \leq \alpha - \tilde{h}_{\bar{s}_p \bar{s}\bar{a}} \end{aligned} \tag{4.24}$$

Computing the dual of this program we have:

$$\begin{aligned} \min_{V\in\mathbb{R}^S,\, y\in\mathbb{R}^{\bar{S}-1},\, y\geq 0} \quad & \begin{pmatrix} (1-\gamma)\,\mu \\ \alpha - \tilde{h}_{\bar{s}_p \bar{s}\bar{a}} \end{pmatrix}^T \begin{pmatrix} V \\ y \end{pmatrix} \\ \text{s.t.} \quad & \begin{pmatrix} E - \gamma P & -B \end{pmatrix} \begin{pmatrix} V \\ y \end{pmatrix} \geq R \end{aligned} \tag{4.25}$$

We do not need to encode the second constraint on reward because, if $(\bar{s}_p \bar{s}\bar{a})$ is realizable, the optimum will satisfy both eqs. (4.13) and (4.14). Finally, the Lagrangian can be written as:

$$\mathcal{L}(V, y; b) = b^T R + V^T(E^T b - \gamma\, P^T b - (1-\gamma)\,\mu) + y^T(-B^T b + \tilde{h}_{\bar{s}_p \bar{s}\bar{a}} - \alpha) \tag{4.26}$$

$$= (1-\gamma)\mu^T V + (\alpha - \tilde{h}_{\bar{s}_p \bar{s}\bar{a}})^T y + b^T(\gamma P V - E V + B y + R) \tag{4.27}$$

and the solution to the linear program expressed as

$$\min_{V,\, y\geq 0}\ \max_{b\geq 0}\ (1-\gamma)\mu^T V + (\alpha - \tilde{h}_{\bar{s}_p \bar{s}\bar{a}})^T y + b^T(\gamma P V - E V + B y + R) \tag{4.28}$$

This is a saddle-point problem whose solution encodes is the realizing option. Once an optimum (V^*, y^*, b^*) is found, the policy can be reconstructed by normalising b^* over states as $\pi_o(a \mid s) \coloneqq b^*(sa)/\sum_{s'} b^*(s'a)$. Although nontrivially, this saddle-point may also approximated iteratively by online approximation methods by performing (projected) gradient descent-ascent along the respective gradients. We do not explore the details of this possibility here.

The dual vector y gives interesting insights about how this formulation works. Looking at the constraint in (4.25), the variables y play the role of artificial rewards, or terminal values, that are placed at exit states. In other words, these variables are excess values that are needed to incentivise an increased state occupancy at exit states. This is consistent with the classic interpretation of slack variables in dual programs. From an HRL perspective, on the other hand, each entry of y is related to the terminal value associated with neighbouring blocks. This is what causes the optimization problem to shift from pure maximization of the block value V_{ν}^{o}, towards a compromise between the current block and future more rewarding blocks. Therefore, if the optimal vector y^* was known in advance, the realizability problem of each abstract state and action could be solved simply by setting the rewards of the block MDP as

$$R_{\bar{s}}(sa) := \begin{cases} R(sa) & \text{if } s \in \lfloor \bar{s} \rfloor \\ y^*(\phi(s)) & \text{if } s \in \mathcal{X}_{\bar{s}} \\ 0 & \text{if } s = s_{\perp} \end{cases} \tag{4.29}$$

and optimizing the classic RL objective over $\mathbf{M}_{\bar{s}}$ with any (Deep) RL technique.

4.6 Discussion

We close this chapter with a conclusive description of the specific contributions. Moreover, at this point of the text, we will be able to do a more detailed comparison with related works from the literature.

This chapter addressed a very general issue in the HRL literature. It introduced a precise class of MDP abstractions, and it showed their properties. These abstractions have been presented independently of any learning algorithm because the properties that we have identified have a more general significance. At the core of the new *realizable abstractions* from definitions 4.3 and 4.5, there is the basic intuition that realizing an abstract transition should aim to replicate both its probability and the accumulated reward. In the presence of discounting, there is a well-understood duality between probability and time. Therefore, these two had been related to their discounted counterparts: occupancy measures and values. Although we have always used the mathematical notation specific for finite sets, our concepts readily generalize in the case where the ground state space is infinite. In particular, we notice that all the relevant notions for ground MDPs, including block partitions, exit states, block MDPs, occupancies, values and expectations, all remain well-defined when the ground state space is infinite. In contrast, abstract states are always assumed to be in finite number. We argue that this is often implicitly assumed in most work in HRL.

In addition to being intuitive, our abstractions allow for obtaining near-optimality guarantees on the ground policies, in corollary 4.2. Since ground policies are obtained as the union of individual options, the learning algorithm for the ground MDP can act in a truly compositional way, as desired. These abstractions also allow for a significant reduction in the abstract effective horizon, whenever possible, as expressed in assumption 4.1. Finally, in section 4.5, we highlighted the main components for developing learning algorithms that are capable both of active abstraction and of realization into ground policies. An important feature of realizable abstractions is that they can incorporate a variable amount of information content. In fact, they may be defined as having an identical state space with approximate probabilities and overestimated rewards, or they may be composed of just two states for goal MDPs. The formulation also adapts to many intermediate representations in between.

This work represents abstract decision processes as 2-MDPs because their second-order dependencies perfectly capture the differences in occupancy measures that depend on the penultimate block visited. A similar beneficial effect can also be observed for rewards. However, because rewards are accumulated within the block, the effect is not as strong. Abstractions could also be modelled as MDPs without any change to our approach. However, for nontrivial compressions of the state space, this choice may result in larger parameters (α, β) and weaker guaranteed realizations. On the opposite side of the spectrum, the advantage of using 3-MDPs instead of 2-MDPs would be marginal.

We will now try to connect our methods with others from the literature. Regarding occupancy measures, our extensive use of state distributions is mostly motivated by their widespread occurrence in MDP planning and learning, but we have also been strongly influenced by the seminal work on Successor Representations (Dayan 1993). These representations have also been used very recently in (Machado, Barreto, et al. 2023). However, their analysis falls short of capturing their prime role for HRL, as well as the exact properties to satisfy.

Although (Abel, Umbanhowar, et al. 2020; Abel 2020) do not treat MDP abstractions as separate decision processes, it is possible to have a comparison specifically for the analyses of ground ϕ-relative options and policies of options. In particular, proposition 4.6 can be compared with Abel, Umbanhowar, et al. (2020, Theorem 1). The bound that we obtain is significantly improved, mainly because the cardinality of the ground state space does not appear in the bound. We remind that this might be very large or even infinite. Apart from this comparison, the two works cannot be easily compared, since our work assumes explicit abstract transitions and reward functions.

With respect to the previous chapter of this thesis, the parameterisation used here, with (α, β), strictly improves over the abstract similarity and the abstract value approximation of the previous work. In fact, the new analysis can consider any policy, not only the optimal, it generalises over generic rewards, not only goal states, and, unlike abstract similarity of chapter 3, α-realizability is not computed by comparing probabilities for all time steps, but it results from the accumulation of all probabilities.

The idea that state partitions can be seen as inducing a number of sub-MDPs with independent dynamics has been used already in the HRL literature. See, for example, Wen, Precup, et al. (2020) and Infante, Jonsson, et al. (2022). However, in this work, through the introduction of the new sink state, it is possible to relate the values and occupancies of the block MDP with abstract transitions and rewards. These works have also been influential for our use of exit states and their value. However, unlike in the algorithm “Planning with Exit Profiles” from Wen, Precup, et al. (2020), our realization procedure does not assume that occupancy or the value at exits is known in advance.

Similarly to our work, MDP homomorphisms are precise relations that link two decision processes (Ravindran and Andrew G. Barto 2002, 2004). The most significant difference is that, unlike our work, MDP homomorphisms cannot model temporal abstraction and are mostly restricted to symmetries. Temporal abstraction is a primary objective for this work. As such, we were able to avoid the collapsing state abstractions and nonstationarity effects that often appear in HRL (Jothimurugan, Bastani, et al. 2021).

Future Work This chapter provides a complete answer to the first general question on page 27, namely, what should be considered “good” MDP abstractions. However, the second question, asking how MDP abstractions can be used to improve compositionality and sample efficiency, remains partially answered. Two directions remain open to be explored in future work. The first is the development of a complete and sample-efficient RL algorithm, which learns on the ground domain in a compositional way by exploiting the properties of realizable abstractions. In fact, in the second half of section 4.5, we have discussed the realization of each abstract tuple independently, but we did not provide an end-to-end algorithm for HRL. The second direction involves how to learn realizable abstractions from experience. The first half of section 4.5, had provided essential insights, limited to each option. However, we did not discuss how to estimate the abstract discount factor or the global transition and rewards functions from experience. On the other hand, estimating good state partitions may be too challenging to pursue at the moment.

4.7 Proofs

This section contains all the proofs for this chapter. The reader may skip this section and refer to it as needed.

Proposition 4.3. *Any MDP* $\mathbf{M}$ *admits* $\langle \mathbf{M}, \mathrm{I} \rangle$ *as an admissible and perfectly realizable abstraction.*

Proof. The ground domain is $\mathbf{M} = \langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$ and the abstraction is $\langle \mathbf{M}, \mathrm{I} \rangle$. Since admissibility is trivially satisfied, we just need to show that this is a perfectly realizable abstraction. The identity function induces the naive partitioning, in which each state is in a separate block: $\lfloor s \rfloor_{\mathrm{I}} = \{s\}$. Also, if we just consider deterministic I-relative options, we see that these are simple repetitions of the same action for the same state. We can now compute the un-normalized block occupancy measure at any state $s \in \mathcal{S}$ and deterministic $o \in \Omega_s$. Then, for $s' \neq s$,

$$\frac{h^o_{\mathrm{I}(s)}(s' \mid s)}{1-\gamma} = \sum_{s' \in \lfloor \mathrm{I}(s') \rfloor} \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(s_t = s' \mid s_0 = s, o, \mathbf{M}_{\mathrm{I}(s)}) \tag{4.30}$$

$$= \sum_{t=1}^{\infty} \gamma^t \, \mathbb{P}(s_{0:t-1} \in \lfloor s \rfloor^t, s_t = s' \mid s_0 = s, o, \mathbf{M}_s) \tag{4.31}$$

$$= \sum_{t=1}^{\infty} \gamma^t \, T(s \mid s\, o(s))^{t-1} \, T(s' \mid s\, o(s)) \tag{4.32}$$

$$= \frac{\gamma \, T(s' \mid s\, o(s))}{1 - \gamma \, T(s \mid s\, o(s))} \tag{4.33}$$

Now we compute un-normalized eq. (4.5) for $\mathbf{M}$. Importantly, since $T(s_p s a) = T(s s a)$, we can just write $T(sa)$:

$$\frac{\tilde{h}_{s_p s a}(s')}{1-\gamma} = \gamma \, T(s' \mid s\, a) + \frac{\gamma^2 \, T(s \mid s\, a) \, T(s' \mid s\, a)}{1 - \gamma \, T(s \mid s\, a)} = \frac{\gamma \, T(s' \mid s\, a)}{1 - \gamma \, T(s \mid s\, a)} \tag{4.34}$$

This proves that $\pi_o(s) = a$ is a perfect realization of a with respect to eq. (4.7). We now consider rewards. The term $V^o_s(s)$, appearing in eq. (4.8), is the cumulative return obtained by repeating action a (since it is the only reward in $\mathbf{M}_s$).

$$V^o_s(s) = \sum_{t=0}^{\infty} \gamma^t \, \mathbb{P}(s_t = s \mid s_0 = s, a, \mathbf{M}_{\bar{s}}) \, R(s\, a) \tag{4.35}$$

$$= \sum_{t=0}^{\infty} \gamma^t \, T(s \mid s\, a)^{t-1} \, R(s\, a) \tag{4.36}$$

$$= \frac{\gamma \, R(s\, a)}{1 - \gamma \, T(s \mid s\, a)} \tag{4.37}$$

Following a similar procedure of eq. (4.34), we also verify eq. (4.8). □

Proposition 4.4. *If $\langle f, \{g_s\}_{s\in\mathcal{S}}\rangle$ is an MDP homomorphism from* $\mathbf{M}$ *and* $\bar{\mathbf{M}}$, *then* $\langle\bar{\mathbf{M}}, f\rangle$ *is an admissible and perfectly realizable abstraction of* $\mathbf{M}$.

Proof. If the ground domain is $\mathbf{M} = \langle\mathcal{S}, \mathcal{A}, T, R, \gamma\rangle$, we choose as abstraction $\langle\bar{\mathbf{M}}, f\rangle$. We compute the un-normalized block occupancy measure at any state $s \in \mathcal{S}$ and deterministic option $o \in \Omega_{f(s)}$. We also assume that o selects the same action for every $\lfloor f(s)\rfloor$. Then, for $\bar{s}' \neq f(s)$,

$$\frac{h^o_{f(s)}(\bar{s}' \mid s)}{1-\gamma} = \sum_{s'\in\lfloor\bar{s}'\rfloor} \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(s_t = s' \mid s_0 = s, o, \mathbf{M}_{f(s)}) \tag{4.38}$$

$$= \sum_{s'\in\lfloor\bar{s}'\rfloor} \sum_{t=1}^{\infty} \gamma^t\, \mathbb{P}(s_{0:t-1} \in \lfloor f(s)\rfloor^t, s_t = s' \mid s_0 = s, o, \mathbf{M}_{f(s)}) \tag{4.39}$$

$$= \sum_{t=1}^{\infty} \gamma^t \sum_{s_{0:t-1}\in\lfloor f(s)\rfloor^t} \sum_{s'\in\lfloor\bar{s}'\rfloor} \mathbb{P}(s_{0:t-1}, s_t = s' \mid s_0 = s, o, \mathbf{M}_{f(s)}) \tag{4.40}$$

Now, summing from s' to s_{t-1} back to s_0 and substituting eq. (4.2),

$$= \sum_{t=1}^{\infty} \gamma^t\, \bar{T}(f(s) \mid f(s)\, g_s(o(s))^{t-1}\, \bar{T}(\bar{s}' \mid f(s)\, g_s(o(s))) \tag{4.41}$$

$$= \frac{\gamma\, \bar{T}(\bar{s}' \mid f(s)\, g_s(o(s)))}{1 - \gamma\, \bar{T}(f(s) \mid f(s)\, g_s(o(s)))} \tag{4.42}$$

Now we compute un-normalized eq. (4.5) for $\mathbf{M}$. Just like in eq. (4.34), since $\bar{T}(\bar{s}_p\bar{s}\bar{a}) = T(\bar{s}\bar{s}\bar{a})$, we can just write $T(\bar{s}\bar{a})$ and:

$$\frac{\tilde{h}_{\bar{s}_p\bar{s}\bar{a}}(\bar{s}')}{1-\gamma} = \frac{\gamma\, \bar{T}(\bar{s}' \mid \bar{s}\, \bar{a})}{1 - \gamma\, \bar{T}(\bar{s} \mid \bar{s}\, \bar{a})} \tag{4.43}$$

This proves that $\pi_o(s) \in g_s^{-1}(\bar{a})$ is a perfect realization of $\bar{a}$ with respect to eq. (4.7). We now consider rewards. The term $V^o_{f(s)}(s)$, appearing in eq. (4.8), is

$$V^o_{f(s)}(s) = \sum_{t=0}^{\infty} \gamma^t \sum_{s_{0:t}\in\lfloor f(s)\rfloor^{t+1}} \mathbb{P}(s_{0:t} \mid s_0 = s, o, \mathbf{M}_{f(s)})\, R(s\, a) \tag{4.44}$$

$$= \sum_{t=0}^{\infty} \gamma^t\, \bar{T}(f(s) \mid f(s)\, o(a))^t\, \bar{R}(f(s)\, o(a)) \tag{4.45}$$

$$= \frac{\gamma\, \bar{R}(f(s)\, o(a))}{1 - \gamma\, \bar{T}(f(s) \mid f(s)\, o(a))} \tag{4.46}$$

By comparison with $\tilde{V}_{\bar{s}_p\bar{s}\bar{a}}$ in eq. (4.8), the same choice $\pi_o(s) \in g_s^{-1}(\bar{a})$ also satisfies the second constraint. □

Proposition 4.5. *Consider any MDP* $\mathbf{M}$ *and surjective function* $\phi : \mathcal{S} \to \bar{\mathcal{S}}$*. Then, from any state* $s \in \mathcal{S}$*, the value of any deterministic* ϕ*-relative option* $o \in \Omega_{\phi(s)}$ *and policy* π *is*

$$Q^\pi(s, o) = \sum_{s' \in \mathcal{S}} \frac{d^o_{\phi(s)}(s' \mid s)}{1 - \gamma} \left(\mathbb{I}(s' \in \lfloor \phi(s) \rfloor)\, R(s'\, o(s')) + \mathbb{I}(s' \in \mathcal{X}_{\phi(s)})\, V^\pi(s')\right) \quad (4.11)$$

where $d^o_{\phi(s)}$ *is the state occupancy measure of* π_o *in the block-restricted MDP* $\mathbf{M}_{\phi(s)}$*.*

Proof. Let $\lfloor \bar{s} \rfloor^{(t)} \coloneqq \lfloor \bar{s} \rfloor^{t-1} \times (\mathcal{S} \setminus \lfloor \bar{s} \rfloor)$ be the set including all trajectories that leave the block in exactly t transitions. We also abbreviate $\bar{s} \coloneqq \phi(s)$.

$$Q^\pi(s, o) = R(s\, o(s)) + \gamma\, \mathbb{E}_{s'}[\mathbb{I}(s' \in \lfloor \bar{s} \rfloor)\, Q^\pi(s', o) + \mathbb{I}(s' \notin \lfloor \bar{s} \rfloor)\, V^\pi(s'))] \quad (4.47)$$

$$= R(s\, o(s)) + \gamma \sum_{s' \in \lfloor \bar{s} \rfloor} T(s' \mid s\, o(s))\, Q^\pi(s', o) + \gamma \sum_{s' \notin \lfloor \bar{s} \rfloor} T(s' \mid s\, o(s))\, V^\pi(s') \quad (4.48)$$

$$\begin{aligned} &= \sum_{t=0}^{\infty} \gamma^t \sum_{s_{1:t} \in \lfloor \bar{s} \rfloor^t} \mathbb{P}(s_{1:t} \mid s_0 = s, o, \mathbf{M})\, R(s_t\, o(s_t)) \\ &\qquad + \sum_{t=1}^{\infty} \gamma^t \sum_{s_{1:t} \in \lfloor \bar{s} \rfloor^{(t)}} \mathbb{P}(s_{1:t} \mid s_0 = s, o, \mathbf{M})\, V^\pi(s_t) \end{aligned} \quad (4.49)$$

$$\begin{aligned} &= \sum_{t=0}^{\infty} \gamma^t \sum_{s_{1:t} \in \lfloor \bar{s} \rfloor^t} \mathbb{P}(s_{1:t} \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, R_{\bar{s}}(s_t\, o(s_t)) \\ &\qquad + \sum_{t=1}^{\infty} \gamma^t \sum_{s_{1:t} \in \lfloor \bar{s} \rfloor^{(t)}} \mathbb{P}(s_{1:t} \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, V^\pi(s_t) \end{aligned} \quad (4.50)$$

In the last equation, all probabilities are computed on the block-restricted MDP $\mathbf{M}_{\bar{s}}$. This is equivalent, since all probabilities of transitions from $\lfloor \bar{s} \rfloor$ are preserved. Since every trajectory that leaves the block may only reach $s_\perp$, without further rewards in $\mathbf{M}_{\bar{s}}$, we can simplify as follows.

$$\begin{aligned} Q^\pi(s, o) &= \sum_{t=0}^{\infty} \gamma^t \sum_{s_{1:t} \in \mathcal{S}^t_{\bar{s}}} \mathbb{P}(s_{1:t} \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, R_{\bar{s}}(s_t\, o(s_t)) \\ &\qquad + \sum_{t=1}^{\infty} \gamma^t \sum_{s' \in \mathcal{X}_{\bar{s}}} \mathbb{P}(s_t = s' \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, V^\pi(s') \end{aligned} \quad (4.51)$$

$$\begin{aligned} &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r_t \mid s, o, \mathbf{M}_{\bar{s}}\right] \\ &\qquad + \sum_{s' \in \mathcal{S}} \sum_{t=1}^{\infty} \gamma^t\, \mathbb{P}(s_t = s' \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, \mathbb{I}(s' \in \mathcal{X}_{\bar{s}})\, V^\pi(s') \end{aligned} \quad (4.52)$$

$$= V^o_{\bar{s}}(s) + \sum_{s' \in \mathcal{S}} \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(s_t = s' \mid s_0 = s, o, \mathbf{M}_{\bar{s}})\, \mathbb{I}(s' \in \mathcal{X}_{\bar{s}})\, V^\pi(s') \quad (4.53)$$

$$
\begin{aligned}
&= (1-\gamma)^{-1} \sum_{s' \in \lfloor \bar{s} \rceil} d^{o}_{\bar{s}}(s' \mid s)\, R(s'\, o(s')) \\
&\qquad + (1-\gamma)^{-1} \sum_{s' \in \mathcal{S}} d^{o}_{\bar{s}}(s' \mid s)\, \mathbb{I}(s' \in \mathcal{X}_{\bar{s}})\, V^{\pi}(s') && (4.54) \\
&= \sum_{s' \in \mathcal{S}} (1-\gamma)^{-1}\, d^{o}_{\bar{s}}(s' \mid s)\, (\mathbb{I}(s' \in \lfloor \bar{s} \rceil)\, R(s'\, o(s')) + \mathbb{I}(s' \in \mathcal{X}_{\bar{s}})\, V^{\pi}(s')) && (4.55)
\end{aligned}
$$

□

Proposition 4.6. *Given a state partition, let Ω_1, Ω_2 be two deterministic policies of options in $\mathbf{M}$ over that partition. For each $\bar{s} \in \bar{\mathcal{S}}$ and $o_1 \in \Omega_{\bar{s}} \cap \Omega_1, o_2 \in \Omega_{\bar{s}} \cap \Omega_2$, assume that $\|d^{o_1}_{\bar{s}} - d^{o_2}_{\bar{s}}\|_1 \leq \alpha$ and $\max_{s \in \lfloor \bar{s} \rceil} |R(s\, o_1(s)) - R(s\, o_2(s))| \leq \beta$. Then, for any state $s \in \mathcal{S}$,*

$$
|V^{\Omega_1}(s) - V^{\Omega_2}(s)| \leq \frac{\alpha + \beta}{(1-\gamma)^2} + \frac{\alpha}{(1-\gamma)^3} \tag{4.12}
$$

Proof. Let us denote with $V^{\Omega_{1,2}}_k(s)$ the value of the policy that executes k consecutive options from Ω_1, then the options from Ω_2 thereafter. Specifically in this regard, this is a similar proof structure than the one used in Abel, Umbanhowar, et al. (2020). Now, with an inductive proof, we show that

$$
|V^{\Omega_2}(s) - V^{\Omega_{1,2}}_k(s)| \leq \left(\frac{\alpha + \beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} \right) \sum_{t=0}^{k-1} \gamma^t \tag{4.56}
$$

For the base case, $V^{\Omega_{1,2}}_0(s) = V^{\Omega_2}(s)$. So the hypothesis is satisfied. For the inductive case, with $k \geq 1$, we have, for any $o_1 \in \Omega_{\bar{s}} \cap \Omega_1$ and $o_2 \in \Omega_{\bar{s}} \cap \Omega_2$, using proposition 4.5,

$$
\begin{aligned}
&|V^{\Omega_2}(s) - V^{\Omega_{1,2}}_k(s)| = && (4.57) \\
&= \frac{1}{1-\gamma} \Big| \sum_{s' \in \mathcal{S}} d^{o_1}_{\phi(s)}(s' \mid s)\, (\mathbb{I}(s' \in \lfloor \phi(s) \rceil)\, R(s'\, o_1(s')) + \mathbb{I}(s' \in \mathcal{X}_{\phi(s)})\, V^{\Omega_2}(s')) \\
&\qquad - \sum_{s' \in \mathcal{S}} d^{o_2}_{\phi(s)}(s' \mid s)\, (\mathbb{I}(s' \in \lfloor \phi(s) \rceil)\, R(s'\, o_2(s')) + \mathbb{I}(s' \in \mathcal{X}_{\phi(s)})\, V^{\Omega_{1,2}}_{k-1}(s')) \Big| && (4.58) \\
&\leq \frac{1}{1-\gamma} \Big| \sum_{s' \in \lfloor \phi(s) \rceil} (d^{o_1}_{\phi(s)}(s' \mid s)\, R(s'\, o_1(s')) - d^{o_2}_{\phi(s)}(s' \mid s)\, R(s'\, o_2(s'))) \Big| \\
&\qquad + \frac{1}{1-\gamma} \Big| \sum_{s' \in \mathcal{X}_{\phi(s)}} (d^{o_1}_{\phi(s)}(s' \mid s)\, V^{\Omega_2}(s') - d^{o_2}_{\phi(s)}(s' \mid s)\, V^{\Omega_{1,2}}_{k-1}(s')) \Big| && (4.59) \\
&\leq \frac{1}{1-\gamma} \sum_{s' \in \lfloor \phi(s) \rceil} |d^{o_1}_{\phi(s)}(s' \mid s)\, R(s'\, o_1(s')) - d^{o_1}_{\phi(s)}(s' \mid s)\, R(s'\, o_2(s'))| \\
&\qquad + \frac{1}{1-\gamma} \sum_{s' \in \lfloor \phi(s) \rceil} |d^{o_1}_{\phi(s)}(s' \mid s)\, R(s'\, o_2(s')) - d^{o_2}_{\phi(s)}(s' \mid s)\, R(s'\, o_2(s'))| && (4.60)
\end{aligned}
$$

$$+ \frac{1}{1-\gamma} \sum_{s' \in \mathcal{X}_{\phi(s)}} |d^{o_1}_{\phi(s)}(s' \mid s)\, V^{\Omega_2}(s') - d^{o_2}_{\phi(s)}(s' \mid s)\, V^{\Omega_2}(s')|$$

$$+ \frac{1}{1-\gamma} \sum_{s' \in \mathcal{X}_{\phi(s)}} |d^{o_2}_{\phi(s)}(s' \mid s)\, V^{\Omega_2}(s') - d^{o_2}_{\phi(s)}(s' \mid s)\, V^{\Omega_{1,2}}_{k-1}(s')| \tag{4.61}$$

$$\leq \frac{1}{1-\gamma} \sum_{s' \in \lfloor \phi(s) \rfloor} d^{o_1}_{\phi(s)}(s' \mid s)\, |R(s'\, o_1(s')) - R(s'\, o_2(s'))|$$

$$+ \frac{1}{1-\gamma} \sum_{s' \in \lfloor \phi(s) \rfloor} R(s'\, o_2(s'))\, |d^{o_1}_{\phi(s)}(s' \mid s) - d^{o_2}_{\phi(s)}(s' \mid s)|$$

$$+ \frac{1}{1-\gamma} \sum_{s' \in \mathcal{X}_{\phi(s)}} V^{\Omega_2}(s')\, |d^{o_1}_{\phi(s)}(s' \mid s) - d^{o_2}_{\phi(s)}(s' \mid s)|$$

$$+ \frac{1}{1-\gamma} \sum_{s' \in \mathcal{X}_{\phi(s)}} d^{o_2}_{\phi(s)}(s' \mid s)\, |V^{\Omega_2}(s') - V^{\Omega_{1,2}}_{k-1}(s')| \tag{4.62}$$

$$\leq \frac{\beta}{1-\gamma} + \frac{\alpha}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2}$$

$$+ \sum_{s' \in \mathcal{X}_{\phi(s)}} \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(s_t = s' \mid s_0 = s, o_1, \mathbf{M}_{\bar{s}}) \max_{s'' \in \mathcal{X}_{\phi(s)}} |V^{\Omega_2}(s'') - V^{\Omega_{1,2}}_{k-1}(s'')| \tag{4.63}$$

$$\leq \frac{\alpha+\beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} +$$

$$\max_{s' \in \mathcal{X}_{\phi(s)}} |V^{\Omega_2}(s') - V^{\Omega_{1,2}}_{k-1}(s')| \sum_{t=1}^{\infty} \gamma^t\, \mathbb{P}(s_t \in \mathcal{X}_{\phi(s)} \mid s_0 = s, o_1, \mathbf{M}_{\bar{s}}) \tag{4.64}$$

since it is impossible to visit states in $\mathcal{X}_{\phi(s)}$ twice in $\mathbf{M}_{\bar{s}}$,

$$\leq \frac{\alpha+\beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} + \gamma \max_{s' \in \mathcal{X}_{\phi(s)}} |V^{\Omega_2}(s') - V^{\Omega_{1,2}}_{k-1}(s')| \tag{4.65}$$

from induction hypothesis,

$$\leq \frac{\alpha+\beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} + \gamma \left(\frac{\alpha+\beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} \right) \sum_{t=0}^{k-2} \gamma^t \tag{4.66}$$

$$= \left(\frac{\alpha+\beta}{1-\gamma} + \frac{\alpha}{(1-\gamma)^2} \right) \sum_{t=0}^{k-1} \gamma^t \tag{4.67}$$

Now, to conclude,

$$|V^{\Omega_1}(s) - V^{\Omega_2}(s)| = \lim_{k \to \infty} |V^{\Omega_2}(s) - V^{\Omega_{1,2}}_{k}(s)| = \frac{\alpha+\beta}{(1-\gamma)^2} + \frac{\alpha}{(1-\gamma)^3} \tag{4.68}$$

□

Lemma 4.8. *Let $\mathbf{M}_{\bar{s}}$ be any block MDP, computed from some MDP $\mathbf{M}$, mapping function ϕ and abstract state $\bar{s}$. Then, for any option $o \in \Omega_{\bar{s}}$, it holds:*

$$d^o_{\bar{s}}(s_\perp \mid s) = (1 - h^o_{\bar{s}}(\bar{s} \mid s))\,\gamma \tag{4.69}$$

$$\sum_{s' \in \mathcal{X}_{\bar{s}}} d^o_{\bar{s}}(s' \mid s) = (1 - h^o_{\bar{s}}(\bar{s} \mid s))\,(1-\gamma) \tag{4.70}$$

Proof. In a block MDP, we remind that the occupancy measure is spread between the block $\lfloor \bar{s} \rfloor$, the exits and the sink state $s_\perp$. In other words,

$$\sum_{s' \in \mathcal{X}_{\bar{s}}} d^o_{\bar{s}}(s' \mid s) = 1 - \sum_{s' \in \lfloor \bar{s} \rfloor} d^o_{\bar{s}}(s' \mid s) - d^o_{\bar{s}}(s_\perp \mid s) = 1 - h^o_{\bar{s}}(\bar{s} \mid s) - d^o_{\bar{s}}(s_\perp \mid s) \tag{4.71}$$

From the definition of occupancy, we also know that

$$\begin{aligned}
d^o_{\bar{s}}(s_\perp \mid s) &= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(s_t = s_\perp \mid s_0 = s, o, \mathbf{M}_{\bar{s}}) && (4.72)\\
&= (1-\gamma) \sum_{t=1}^{\infty} \gamma^t\, \mathbb{P}(s_t = s_\perp \mid s_0 = s, o, \mathbf{M}_{\bar{s}}) && (4.73)\\
&= (1-\gamma) \sum_{t=1}^{\infty} \gamma^t\, \mathbb{P}(s_{t-1} \in \mathcal{X}_{\bar{s}} \cup \{s_\perp\} \mid s_0 = s, o, \mathbf{M}_{\bar{s}}) && (4.74)\\
&= \gamma\,(1-\gamma) \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(s_t \in \mathcal{X}_{\bar{s}} \cup \{s_\perp\} \mid s_0 = s, o, \mathbf{M}_{\bar{s}}) && (4.75)\\
&= \gamma \left(\sum_{s' \in \mathcal{X}_{\bar{s}}} d^o_{\bar{s}}(s' \mid s) + d^o_{\bar{s}}(s_\perp \mid s) \right) && (4.76)
\end{aligned}$$

Substituting eq. (4.71) into eq. (4.76) gives the result. □

Lemma 4.9. *In any 2-MDP $\mathbf{M}$ and deterministic policy π, for any two distinct states $s_p \in \mathcal{S} \cup \{s_\circ\}$ and $s \in \mathcal{S}$,*

$$V^\pi(s_p s) = R_{s_p s} + \frac{\gamma\, T_{s|s_p s}}{1 - \gamma\, T_{s|ss}} R_{ss} + \sum_{s' \in \mathcal{S}\setminus\{s\}} \left(\gamma\, T_{s'|s_p s} + \frac{\gamma^2\, T_{s|s_p s}\, T_{s'|ss}}{1 - \gamma\, T_{s|ss}} \right) V^\pi(ss') \tag{4.77}$$

where $T_{s_3|s_1 s_2} \coloneqq T(s_3 \mid s_1 s_2\, \pi(s_1 s_2))$ and $R_{s_1 s_2} \coloneqq R(s_1 s_2\, \pi(s_1 s_2))$.

Proof. We use the abbreviations $T_{s_1|s_1 s_2}$ and $R_{s_1 s_2}$ to avoid excessive verbosity. Then,

$$\begin{aligned}
V^\pi(s_p s) &= \sum_{s' \in \mathcal{S}} T_{s'|s_p s}\, (R_{s_p s} + \gamma\, V^\pi(ss')) && (4.78)\\
&= R_{s_p s} + \sum_{s' \in \mathcal{S}\setminus\{s\}} T_{s'|s_p s}\, \gamma\, V^\pi(ss') + T_{s|s_p s}\, \gamma\, V^\pi(ss) && (4.79)
\end{aligned}$$

$$= R_{s_p s} + \sum_{s' \in \mathcal{S} \setminus \{s\}} T_{s'|s_p s} \, \gamma \, V^{\pi}(ss') + T_{s|s_p s} \, \gamma \, R_{ss} \tag{4.80}$$

$$+ T_{s|s_p s} \, \gamma \, T_{s|ss} \, \gamma \, V^{\pi}(ss) + T_{s|s_p s} \, \gamma \sum_{s' \in \mathcal{S} \setminus \{s\}} T_{s'|s_p s} \, \gamma \, V^{\pi}(ss') \tag{4.81}$$

$$= R_{s_p s} + \gamma \, T_{s|s_p s} \, R_{ss} \sum_{t=0}^{\infty} \gamma^t \, T_{s|ss}^t$$

$$+ \sum_{s' \in \mathcal{S} \setminus \{s\}} \gamma \, T_{s'|s_p s} \, V^{\pi}(ss') + \gamma \, T_{s|s_p s} \sum_{s' \in \mathcal{S} \setminus \{s\}} \gamma \, T_{s'|ss} \, V^{\pi}(ss') \sum_{t=0}^{\infty} \gamma^t \, T_{s|ss}^t \tag{4.82}$$

$$= R_{s_p s} + \frac{\gamma \, T_{s|s_p s}}{1 - \gamma \, T_{s|ss}} \, R_{ss} + \sum_{s' \in \mathcal{S} \setminus \{s\}} \left(\gamma \, T_{s'|s_p s} + \frac{\gamma^2 \, T_{s|s_p s} \, T_{s'|ss}}{1 - \gamma \, T_{s|ss}} \right) V^{\pi}(ss') \tag{4.83}$$

□

Theorem 4.1. *Let* $\langle \bar{\mathbf{M}}, \phi \rangle$ *be an* (α, β)*-realizable abstraction of an MDP* $\mathbf{M}$*, whose initial distributions satisfy* $\max_{\bar{s}} |\bar{\mu}(\bar{s}) - \sum_{s \in \lfloor \bar{s} \rfloor} \mu(s)| \leq \xi$*. Then, if* Ω' *is the realization of some deterministic abstract policy* $\bar{\pi}$*,*

$$\bar{V}_{\bar{\mu}}^{\bar{\pi}} - V_{\mu}^{\Omega'} \leq \frac{\beta}{(1-\gamma)^2} + \frac{\alpha \, |\bar{\mathcal{S}}|}{(1-\gamma)^2 (1-\bar{\gamma})} + \frac{\xi \, |\bar{\mathcal{S}}|}{1-\bar{\gamma}} \tag{4.9}$$

Proof. First, let us define an abbreviation for the set of previous states, $\bar{\mathcal{S}}_\circ \coloneqq \bar{\mathcal{S}} \cup \{\bar{s}_\circ\}$. To relate the two values, we start by inductively defining a set of functions $V_0, V_1, \dots$ as $V_0(s_p s) \coloneqq \bar{V}^{\bar{\pi}}(\phi(s_p)\phi(s))$, and

$$V_k(s_p s) \coloneqq \mathbb{E}\left[g^o + \gamma^t \, V_{k-1}(s_{t-1} s_t) \mid s_p s, o \in \Omega' \cap \Omega_{\phi(s_p)\phi(s)} \right] \tag{4.84}$$

where g^o is the cumulative discounted return of the option o.

With an inductive proof, we show that, for every $k \in \mathbb{N}$, $\bar{s}_p \in \bar{\mathcal{S}}_\circ$, $s_p \in \lfloor \bar{s}_p \rfloor$, $s \in \mathcal{X}_{\bar{s}_p}$,

$$\bar{V}^{\bar{\pi}}(\bar{s}_p \bar{s}) - V_k(s_p s) \leq \sum_{i=0}^{k} \gamma^i \, \frac{\beta \, (1-\bar{\gamma}) + \alpha \, \bar{S}}{(1-\gamma)(1-\bar{\gamma})} \tag{4.85}$$

where, for this derivation, we are using the syntactic abbreviation $\bar{s} \coloneqq \phi(s)$ and $\bar{S} \coloneqq |\bar{\mathcal{S}}|$. For the base case, $k = 0$ and $V_0(s_p s) = \bar{V}^{\bar{\pi}}(\bar{s}_p \bar{s})$ everywhere. Now, for the inductive step, we apply lemma 4.9 and proposition 4.5 to the two value functions, respectively. We also use the same abbreviations of lemma 4.9, $\bar{T}_{\bar{s}_3|\bar{s}_1\bar{s}_2}$ and $\bar{R}_{\bar{s}_1\bar{s}_2}$. Then,

$$\bar{V}^{\bar{\pi}}(\bar{s}_p \bar{s}) - V_k(s_p s) = \tag{4.86}$$

$$= \bar{R}_{\bar{s}_p \bar{s}} + \frac{\bar{\gamma} \, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}}{1 - \bar{\gamma} \, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \, \bar{R}_{\bar{s}\bar{s}} + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\bar{\gamma} \, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2 \, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}} \, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma} \, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) \bar{V}^{\bar{\pi}}(\bar{s}\bar{s}')$$

$$- \sum_{s' \in \mathcal{S}_{\bar{s}}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma} \left(\mathbb{I}(s' \in \lfloor \bar{s} \rfloor)\, R(s'\, o(s')) + \mathbb{I}(s' \in \mathcal{X}_{\bar{s}})\, V_{k-1}(s')\right) \tag{4.87}$$

$$\begin{aligned} &= \bar{R}_{\bar{s}_p \bar{s}} + \frac{\bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}}\, \bar{R}_{\bar{s}\bar{s}} - \sum_{s' \in \lfloor \bar{s} \rfloor} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, R(s'\, o(s')) \\ &\quad + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\bar{\gamma}\, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}\, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) \bar{V}^{\pi}(\bar{s}\bar{s}') - \sum_{s' \in \mathcal{X}_{\bar{s}}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, V_{k-1}(s') \end{aligned} \tag{4.88}$$

If $V^o_{\bar{s}}$ is the value function of o in the block-restricted MDP $\mathbf{M}_{\bar{s}}$,

$$\begin{aligned} &= \bar{R}_{\bar{s}_p \bar{s}} + \frac{\bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}}\, \bar{R}_{\bar{s}\bar{s}} - V^o_{\bar{s}}(s) \\ &+ \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\left(\bar{\gamma}\, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}\, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) \bar{V}^{\pi}(\bar{s}\bar{s}') - \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, V_{k-1}(s') \right) \end{aligned} \tag{4.89}$$

using the fact that $s' \in \mathcal{E}_{\bar{s}\bar{s}'}$, and $\langle \bar{\mathbf{M}}, \phi \rangle$ is an (α, β)-realizable abstraction,

$$\leq \frac{\beta}{1-\gamma} + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\left(\bar{\gamma}\, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}\, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) \bar{V}^{\pi}(\bar{s}\bar{s}') - \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, V_{k-1}(s') \right) \tag{4.90}$$

Now, we add and subtract $\sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s'|s)}{1-\gamma}\, \bar{V}^{\pi}(\bar{s}\bar{s}')$,

$$\begin{aligned} &= \frac{\beta}{1-\gamma} + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, \bar{V}^{\pi}(\bar{s}\bar{s}') - \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, V_{k-1}(s') \right) \\ &\quad + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \left(\left(\bar{\gamma}\, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}\, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) \bar{V}^{\pi}(\bar{s}\bar{s}') - \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma}\, \bar{V}^{\pi}(\bar{s}\bar{s}') \right) \end{aligned} \tag{4.91}$$

$$\begin{aligned} &= \frac{\beta}{1-\gamma} + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \sum_{s' \in \mathcal{E}_{\bar{s}\bar{s}'}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma} \left(\bar{V}^{\pi}(\bar{s}\bar{s}') - V_{k-1}(s')\right) \\ &\quad + \sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} \bar{V}^{\pi}(\bar{s}\bar{s}') \left(\left(\bar{\gamma}\, \bar{T}_{\bar{s}'|\bar{s}_p \bar{s}} + \frac{\bar{\gamma}^2\, \bar{T}_{\bar{s}|\bar{s}_p \bar{s}}\, \bar{T}_{\bar{s}'|\bar{s}\bar{s}}}{1 - \bar{\gamma}\, \bar{T}_{\bar{s}|\bar{s}\bar{s}}} \right) - \frac{h^o_{\bar{s}}(\bar{s}' \mid s)}{1-\gamma} \right) \end{aligned} \tag{4.92}$$

applying the inductive hypothesis to the first line and the definition of an (α, β)-realizable abstraction to the second line,

$$\leq \frac{\beta}{1-\gamma} + \sum_{s' \in \mathcal{X}_{\bar{s}}} \frac{d^o_{\bar{s}}(s' \mid s)}{1-\gamma} \sum_{i=0}^{k-1} \gamma^i\, \frac{\beta\,(1-\bar{\gamma}) + \alpha\, \bar{S}}{(1-\gamma)(1-\bar{\gamma})} + \frac{\alpha\, \bar{S}}{(1-\gamma)(1-\bar{\gamma})} \tag{4.93}$$

It only remains to quantify $\sum_{s' \in \mathcal{X}_{\bar{s}}} d^o_{\bar{s}}(s' \mid s)$. To do this, we apply lemma 4.8 which gives,

$$\sum_{s' \in \mathcal{X}_{\bar{s}}} d^o_{\bar{s}}(s' \mid s) = (1 - h^o_{\bar{s}}(\bar{s} \mid s))\,(1-\gamma) \tag{4.94}$$

However, since the option starts in $s \in \lfloor \bar{s} \rfloor$, the occupancy $h^o_{\bar{s}}(\bar{s} \mid s)$ cannot be less than $(1-\gamma)$. This allows us to complete the inequality and obtain

$$\bar{V}^{\bar{\pi}}(\bar{s}_p \bar{s}) - V_k(s_p s) \le \frac{\beta\,(1-\bar{\gamma}) + \alpha\,\bar{S}}{(1-\gamma)(1-\bar{\gamma})} + \gamma \sum_{i=0}^{k-1} \gamma^i\, \frac{\beta\,(1-\bar{\gamma}) + \alpha\,\bar{S}}{(1-\gamma)(1-\bar{\gamma})} \tag{4.95}$$

$$= \sum_{i=0}^{k} \gamma^i\, \frac{\beta\,(1-\bar{\gamma}) + \alpha\,\bar{S}}{(1-\gamma)(1-\bar{\gamma})} \tag{4.96}$$

This verifies the inductive step.

To conclude the proof, we express the value difference from initial distributions:

$$\bar{V}^{\bar{\pi}}_{\bar{\mu}} - V^{\Omega'}_{\mu} = \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{\mu}(\bar{s})\, \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s}) - \sum_{s \in \mathcal{S}} \mu(s)\, V^{\Omega'}(s) \tag{4.97}$$

$$\begin{aligned} &= \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{\mu}(\bar{s})\, \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s}) - \sum_{\bar{s} \in \bar{\mathcal{S}}} \sum_{s \in \lfloor \bar{s} \rfloor} \mu(s)\, \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s}) \\ &\quad + \sum_{\bar{s} \in \bar{\mathcal{S}}} \sum_{s \in \lfloor \bar{s} \rfloor} \mu(s)\, \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s}) - \sum_{s \in \mathcal{S}} \mu(s)\, V^{\Omega'}(s) \end{aligned} \tag{4.98}$$

$$= \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s})\, (\bar{\mu}(\bar{s}) \;- \sum_{s \in \lfloor \bar{s} \rfloor} \mu(s)) + \sum_{s \in \mathcal{S}} \mu(s)\, (\bar{V}^{\bar{\pi}}(\bar{s}_\circ \phi(s)) - V^{\Omega'}(s)) \tag{4.99}$$

using the assumption on initial distributions, and the derivation above,

$$\le \sum_{\bar{s} \in \bar{\mathcal{S}}} \bar{V}^{\bar{\pi}}(\bar{s}_\circ \bar{s})\, \xi + \sum_{s \in \mathcal{S}} \mu(s) \lim_{k \to \infty} (\bar{V}^{\bar{\pi}}(\bar{s}_\circ \phi(s)) - V_k(s_\circ s)) \tag{4.100}$$

$$\le \frac{\bar{S}\,\xi}{1-\bar{\gamma}} + \sum_{s \in \mathcal{S}} \mu(s) \lim_{k \to \infty} (\bar{V}^{\bar{\pi}}(\bar{s}_\circ \phi(s)) - V_k(s_\circ s)) \tag{4.101}$$

$$\le \frac{\bar{S}\,\xi}{1-\bar{\gamma}} + \lim_{k \to \infty} \sum_{i=0}^{k} \gamma^i\, \frac{\beta\,(1-\bar{\gamma}) + \alpha\,\bar{S}}{(1-\gamma)(1-\bar{\gamma})} \tag{4.102}$$

$$\le \frac{\bar{S}\,\xi}{1-\bar{\gamma}} + \frac{\beta\,(1-\bar{\gamma}) + \alpha\,\bar{S}}{(1-\gamma)^2(1-\bar{\gamma})} \tag{4.103}$$

□

Proposition 4.7. *Consider an MDP* $\mathbf{M}$*, an abstraction* $\langle \bar{\mathbf{M}}, \phi \rangle$*, any tuple* $(\bar{s}_p \bar{s} \bar{a})$ *with* $\bar{s}_p \neq \bar{s}$*, distribution* $\nu \in \Delta(\mathcal{E}_{\bar{s}_p \bar{s}})$*, option* $o \in \Omega_{\bar{s}_p \bar{s}}$*, with associated targets* h^o_ν, V^o_ν*. Then, under assumption 4.1, the output of* $\textsc{AbstractOne}(\bar{\mathbf{M}}, (\bar{s}_p \bar{s} \bar{a}), h^o_\nu, V^o_\nu)$ *is a valid 2-MDP and the option* o *is a perfect realization of* $(\bar{s}_p \bar{s} \bar{a})$ *from* ν*.*

Proof. The first property to verify is that the output of $\textsc{AbstractOne}(\bar{\mathbf{M}}, (\bar{s}_p \bar{s} \bar{a}), h^o_\nu, V^o_\nu)$ is a valid MDP. In other words, the modified transition and reward functions must be in the valid ranges. This is true for each $\bar{s}'_p \in \bar{\mathcal{S}} \setminus \{\bar{s}_p, \bar{s}\}$. In fact, after the assignment, $\bar{R}(\bar{s}'_p \bar{s}\, \bar{a}) \in [0, 1]$. Also, $\tilde{h}_{\bar{s}'_p \bar{s}\bar{a}}(\bar{s}') \in [0, (1-\gamma)\bar{\gamma}]$, thanks to eq. (4.5), which is sufficient to guarantee that $\bar{T}(\bar{s}'_p \bar{s}, \bar{a})$ is indeed a probability distribution.

It only remains to verify $\bar{T}(\bar{s}_p \bar{s}\bar{a})$ and $\bar{R}(\bar{s}\bar{s}\bar{a})$. Since the input occupancy corresponds to some real option in $\mathbf{M}$, using the fact that h^o_ν is an expectation, and according to eq. (4.15),

$$h^o_\nu(\bar{s}) \geq \min_{s \in \mathcal{E}_{\bar{s}_p \bar{s}}} h^o_{\bar{s}}(\bar{s} \mid s) \geq 1 - \bar{\gamma} \tag{4.104}$$

Then,

$$(1-\gamma)(1 - h^o_\nu(\bar{s})) \leq (1-\gamma)\bar{\gamma} \tag{4.105}$$

We know that, similarly to eq. (4.70), it holds

$$\sum_{\bar{s}' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} h^o_\nu(\bar{s}') = (1 - h^o_\nu(\bar{s}))\,(1-\gamma) \tag{4.106}$$

Then, for each $\bar{s}' \neq \bar{s}$,

$$h^o_\nu(\bar{s}') \leq \sum_{\bar{s}'' \in \bar{\mathcal{S}} \setminus \{\bar{s}\}} h^o_\nu(\bar{s}'') \leq (1-\gamma)\bar{\gamma} \tag{4.107}$$

This confirms that $\bar{T}(\bar{s}_p \bar{s}\bar{a})$ is indeed a distribution after the assignment. It only remains to verify $\bar{R}(\bar{s}\bar{s}\bar{a})$. However, thanks to eq. (4.16), we also know that $\bar{R}(\bar{s}\bar{s}\bar{a}) \in [0, 1]$. In the special case of $h^o_\nu(\bar{s}) = 1 - \bar{\gamma}$, both the numerator and the denominator become 0 in line 4 of the AbstractOne function. This is resolved with the convention that $0/0 = 0$.

The second half of the statement is that the option o generating V^o_ν and h^o_ν is a $(0, 0)$-realization of $(\bar{s}_p \bar{s}, \bar{a})$ from the entry distribution ν. To verify this, we compute the desired block occupancy and value from eqs. (4.5) and (4.6) using the abstract MDP returned by AbstractOne. Since $\bar{T}(\bar{s}' \mid \bar{s}\bar{s}\bar{a}) \leftarrow 0$ and $\bar{T}(\bar{s}' \mid \bar{s}_p \bar{s}\bar{a}) \leftarrow h^o_\nu(\bar{s}')/((1-\gamma)\bar{\gamma})$, we have $\tilde{h}_{\bar{s}_p \bar{s}\bar{a}} = h^o_\nu$, which satisfies eq. (4.13). For rewards, we first quantify the following:

$$\bar{T}(\bar{s} \mid \bar{s}_p \bar{s}\bar{a}) \leftarrow 1 - \sum_{\bar{s}'' \neq \bar{s}} \bar{T}(\bar{s}'' \mid \bar{s}_p \bar{s}\bar{a}) = 1 - \sum_{\bar{s}' \neq \bar{s}} \frac{h^o_\nu(\bar{s}')}{(1-\gamma)\bar{\gamma}} \tag{4.108}$$

using the expectation of eq. (4.70), now,

$$\bar{T}(\bar{s} \mid \bar{s}_p \bar{s}\bar{a}) = 1 - \frac{1 - h^o_\nu(\bar{s})}{\bar{\gamma}} = \frac{h^o_\nu(\bar{s}) - (1 - \bar{\gamma})}{\bar{\gamma}} \tag{4.109}$$

Substituting this and the other assignments of AbstractOne into eq. (4.6), we obtain

$$\begin{aligned}
\tilde{V}_{\bar{s}_p\bar{s}\bar{a}} &= \bar{R}(\bar{s}_p\bar{s}\bar{a}) + \bar{\gamma}\,\bar{T}_{\bar{s}_p\bar{s}\bar{a}}\,\bar{R}(\bar{s}\bar{s}\bar{a}) && (4.110)\\
&= \min\{1, V^o_\nu\} + \left(\frac{h^o_\nu(\bar{s})}{1-\bar{\gamma}} - 1\right)\,\bar{R}(\bar{s}\bar{s}\bar{a}) && (4.111)\\
&= \min\{1, V^o_\nu\} + \max\{0, V^o_\nu - 1\} && (4.112)\\
&= V^o_\nu && (4.113)
\end{aligned}$$

□

Part III

Learning in Non-Markov Decision Processes

Introduction to part III

MDPs have been extensively studied in the RL literature. They can be effectively regarded as the target decision process of choice for introductory courses in RL (Sutton and Andrew G. Barto 2018). The joint effort of the RL community is mainly due to their very favourable properties. However, real-world applications often cannot be regarded as Markovian. In particular, when considering AI agents as individual actors with their own perceptions, the environment state inevitably becomes partially observable, since no sensor can provide the agent with complete information of the environment state. Apart from missing perceptions, this is also true in environments with multiple agents. Despite the huge relevance for AI, relatively limited progress has been made in POMDP with respect to MDPs. This is clearly well motivated. A classic result from (Papadimitriou and Tsitsiklis 1987) states that planning in POMDP is PSPACE-hard with respect to the horizon length. For comparison, we recall that planning in MDPs is polynomial. This inevitably makes many of these problems inherently complex to tackle.

In this second part of the thesis we consider RL in non-Markovian environments. Specifically, we focus on a relatively recent decision process, which, like POMDPs, does not rely on Markov assumptions. This is called the Regular Decision Process (RDP) (Brafman and De Giacomo 2019). RDPs are very interesting models to study because they are strictly more expressive than MDPs and k-MDPs, while they are also less expressive than the full class of POMDPs, as we will show. In fact, this model has the potential to act as an important middle ground between the very distant classes of k-MDPs and POMDPs. They can capture many interesting environment dynamics, which can be solved with RL algorithms specific to RDPs, and would have to be solved with generic POMDP algorithms otherwise. As we shall see, studying RL for RDPs also gives applicable insights for RL in POMDPs.

This part is composed of two chapters. After an introduction to RDPs, chapter 5, discusses the properties of this model, and shows original results on their expressive power and their relation to POMDPs. Chapter 6, instead, illustrates an original RDP learning algorithm with formal efficiency guarantees.

Chapter 5

The Expressive Power of RDPs

The content of this chapter is based on original work.

5.1 Introduction

Regular Decision Processes have been introduced in Brafman and De Giacomo (2019), as models for capturing a favourable class of history-dependent dynamics. This allowed them to achieve the desired expressiveness, which, as stated in the original paper, can be written as

$$\text{MDP} \subset \text{k-MDP} \subset \text{RDP} \subseteq \text{POMDP} \tag{5.1}$$

Despite this very interesting positioning, planning in RDPs remains polynomial in all relevant variables. This is somehow a surprising fact, since planning in POMDP is PSPACE-hard. Given these very favourable computational properties, we aim to answer two currently open questions: how can we characterise the expressive power of RDPs? What is the complexity of learning in RDPs? This chapter mainly addresses the first question. Although we will also provide a first answer to the second question, this will be specifically addressed in chapter 6, which proposes an original RL algorithm for RDPs.

Before proceeding, it is important to summarise the working principle of RDPs. They have been formally defined in section 2.2, and we refer the reader to this introduction first. Here, we aim to give a more explicit interpretation of that definition. In this thesis, an RDP is represented as a finite-state transducer, in which each state uniquely determines the output distributions on observations and rewards. The automaton state of an RDP is not observable, since it does not appear in the trace. In this regard, it acts as a hidden state of a POMDP. The major difference, however, is that this hidden state can be deterministically computed from the features that

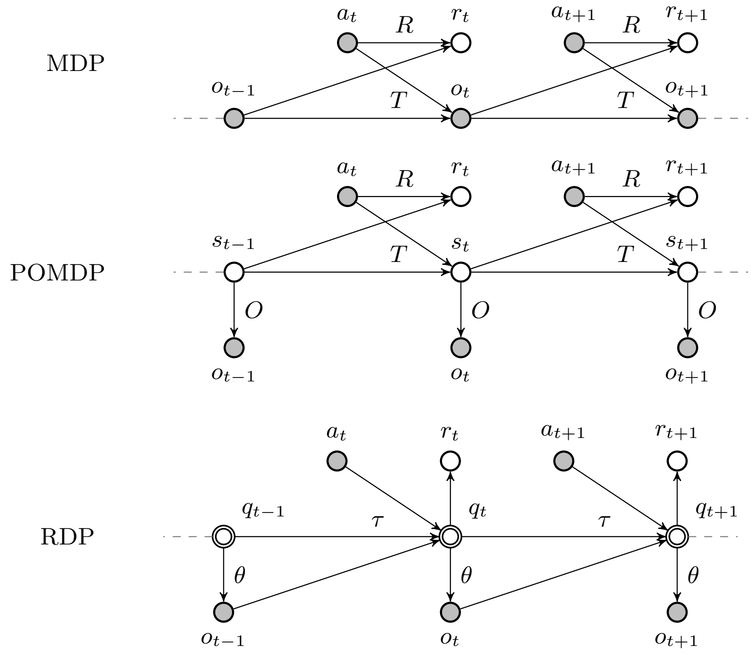
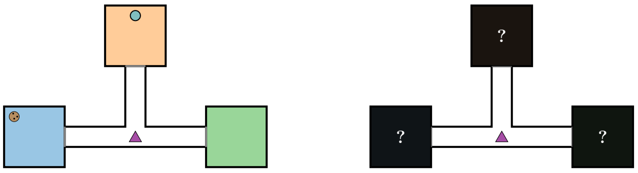


Figure 5.1. Bayesian networks for different decision processes. The variables ○ are hidden, ● are observable, and ◎ are deterministic given the incoming arcs.

are present as visible quantities in the history. Although the RDP state is uniquely determined by each history, the same is not true for POMDPs. We should not underestimate the expressive power of RDPs: the hidden state can be deterministically computed from h_t, but not from the last observation, nor any proper subset of h_t.

In fig. 5.1, we can see the directed graphical models for MDPs, POMDPs, and RDPs. These are Bayesian networks, and they are associated with a specific semantics. In particular, each node is a random variable, and two variables are *not* connected by an arc if they are conditionally independent, given the other nodes. More details are provided in the caption. For a more complete reference on Bayesian networks, the reader may refer to chapter 10 of Murphy (2012).

In this chapter, RDPs are represented as Moore machines as defined in section 2.2. However, we recall that this representation is not the only possibility. RDPs have been previously defined in the literature from temporal logics (Brafman and De Giacomo 2019), mealy machines (Abadi and Brafman 2020), and Moore machines with conditional outputs (Cipollone, Jonsson, et al. 2024). Due to some classic results in automata theory, these definitions are largely equivalent. The only meaningful change is the extension from deterministic outputs to fully stochastic observations and rewards. In fact, rather than being tied to a single automaton structure, we recall that RDPs are more properly characterised by the regularity of the non-Markovian transition and reward functions, $\bar{T}$ and $\bar{R}$.



(a) Cookie domain. **(b)** Agent's view.

Figure 5.2. The *cookie* domain: the agent can only see what is in the room it occupies (figure from Icarte, Waldie, et al. (2019)).

To further clarify how RDPs work, we define a concrete example, based on a partially observable environment. We consider an environment called the "cookie domain" from Icarte, Waldie, et al. (2019). The cookie domain (fig. 5.2a) has three rooms connected by a hallway. The agent (purple triangle) can move in the four cardinal directions. When pressing a button in the orange room, a cookie randomly appears in either the green or the blue room. After finding the cookie, the agent can eat it and the button can be pressed again. This domain is partially observable because the agent can only see what is in the room it currently occupies (fig. 5.2b). More precisely, it is a POMDP with actions $\mathcal{A} = \{\mathit{eat}, \mathit{push}, \rightarrow, \leftarrow, \uparrow, \downarrow\}$ and observations $\mathcal{O} = \mathit{Rooms} \times \{$🍪, 🚫$\}$, where *Rooms* is the colour of the current room.

This domain can also be modelled as an RDP having states $\mathcal{Q} = \mathit{Rooms} \times \mathcal{U}$, which keep track of the agent's position and the current belief about the position of the cookie. Because of these two independent components, the complete automaton is also shown as a composition of two automata. In fig. 5.3, the one on the left generates the observation in *Rooms*, while the automaton on the right generates the observation in $\{$🍪, 🚫$\}$. The full RDP is the synchronous composition of the two.

What is so peculiar of the RDP formulation of this domain is that, upon pushing the button, the RDP goes into an uncertain state ❓, where the cookie may be generated in either the blue or the green room with uniform probability. In a POMDP formulation, the button would cause a cookie to be generated somewhere, possibly hidden from the agent's perspective. In RDPs, instead, the cookie is not generated at all, not until the agent enters one of the two rooms, at which point it will remain where it appeared until it gets eaten. This interesting behaviour is consistent with the conclusions that we draw in this chapter.

5.1.1 Contributions

Being RDPs relatively recent, studying the expressive power of this class is an important topic on its own. However, the main motivation for this work comes from the need of exploring new learning paradigms for partially observable and

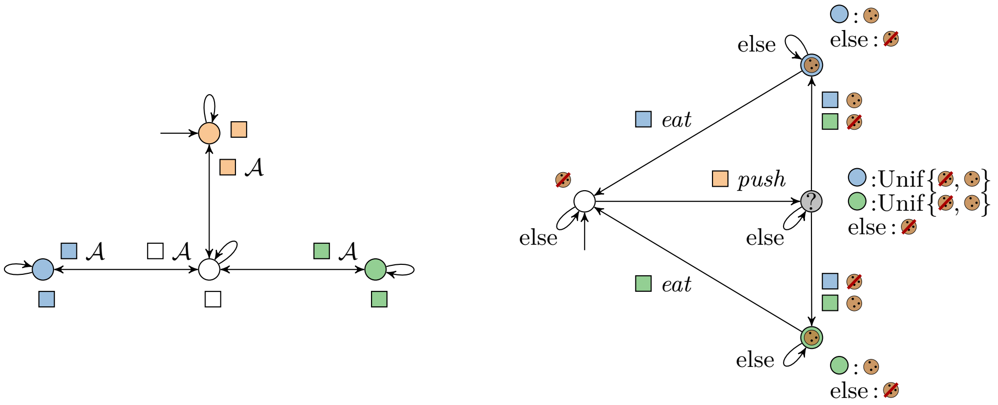


Figure 5.3. In these graphs, the arc label between q and q' is oa if $\tau(q, oa) = q'$. $\mathcal{A}$ represents any action. The label of a state q is ν if $\theta_{\mathsf{o}}(q) = \nu$ (note that some outputs are conditional on the current state of the left component). Finally, if the output is deterministic, we just write o instead of δ_o.

non-Markovian environments. By characterising the expressive power of RDPs, it may be possible to understand that the same RDP algorithms that are currently available are also applicable in other decision processes, which might appear not really related at first sight. This would allow for connecting branches of the RL literature that currently evolve as separate and adopting innovative approaches in the respective decision processes, without additional effort required. In fact, although learning in RDPs remains a complex problem, planning over them is very effective.

The contributions of this work are multiple.

- After section 5.2, containing an extensive comparison of how RDPs and POMDPs work, section 5.3.1 defines what does it mean for two generic decision processes to be equivalent based on their external outputs. Using this notion, in the first original contribution, theorem 5.4, we demonstrate that RDPs are strictly less expressive than POMDPs. To confirm this fact, in proposition 5.5, we verify that the class of optimal policies for POMDPs is not a regular language.

- In section 5.3.3, we demonstrate a number of positive results that confirm that RDPs can approximate, or exactly capture, many POMDPs. In particular, the POMDPs for which an equivalent RDP exists are characterised in corollary 5.8. Then, after defining what it means to approximate a decision process, we show that ξ-observable and ρ-mixing POMDPs can both be approximated by RDPs. Importantly, these results are applicable to infinite horizons and the arguments do not involve any discount factor.

- In the conclusive section, we generalise the previous results and observe that

RDPs can approximate a large group of POMDPs. Consistently with generic POMDPs, in theorem 5.14, we show a generic sample efficiency lower bound for RL in RDPs, with an exponential dependence on the horizon.

5.1.2 Related Work

Despite its complexity, the optimal control problem of partially observable systems in discrete time dates back to very early works such as Åström (1965), which identified closed-form solutions for the finite-horizon planning problem for POMDPs. Later, Smallwood and Sondik (1973) explicitly expressed the recursive formulation of belief states and showed that value functions in finite-horizon POMDPs are piecewise linear convex. In the infinite horizon, instead, the value functions are only convex (Sondik 1978). These two observations allowed the development of new exact and approximate POMDP planning algorithms, such as Hansen (1998), N. L. Zhang, S. S. Lee, et al. (1999), and Pineau, Gordon, et al. (2003).

Unfortunately, an important negative result from Papadimitriou and Tsitsiklis (1987), demonstrated that optimal planning in POMDPs for finite horizons is PSPACE-hard. Even computing optimal policies within important policy classes is intractable (Mundhenk 2000). Moreover, in infinite horizons, deciding whether there exists a policy achieving a value higher than some threshold is undecidable (Madani, Hanks, et al. 1999). So, only near-optimal planning seems to be feasible. Unfortunately, this thread of complexity results was complemented by Lusena, Goldsmith, et al. (2001), which showed that near-optimal POMDP policies cannot be computed in polynomial time. For comparison, planning optimal policies in MDPs is P-complete under both horizon settings.

Due to all these negative results, many authors focused on favourable subclasses of POMDPs or instance-dependent descriptions. Some characterisations are mainly associated with transition functions and mixing times (Boyen and Koller 1998), while others mainly target the observation function (Even-Dar, Sham M. Kakade, et al. 2007; Golowich, Moitra, et al. 2022a). One very successful characterisation, which is not associated with any strict assumption, is that of covering numbers (Hsu, W. S. Lee, et al. 2007). These will be formally defined in section 5.2. Intuitively, the covering number for a POMDP quantifies how many belief points are needed to have a sufficient finite cover of the reachable belief space. As the authors show, the time required to compute near-optimal policies can be expressed as a polynomial function of this covering number. The same parameter has also been shown to well characterise the efficiency of *learning* algorithms in POMDPs (Z. Zhang, Littman, et al. 2012). In fact, perhaps unsurprisingly, without some assumptions or instance-dependent parameters, RL in POMDPs is also intractable in the general case. As

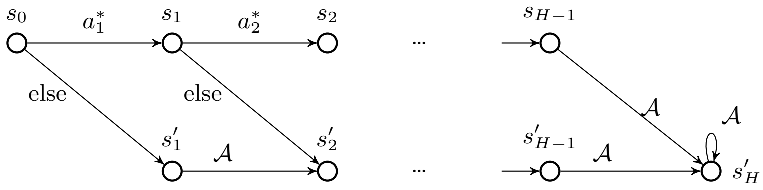


Figure 5.4. Hard POMDP instance from Krishnamurthy, Agarwal, et al. (2016). Transitions are deterministic, as shown by the arcs. The observation function is $O(s_i) = O(s'_i) = \delta_{s_i}$, for each i. The reward function is zero everywhere, except for the states s_{H-1}, s'_{H-1}, where it is $R(s\,a) = \text{Ber}(0.5 + \epsilon)$, if $s = s_{H-1}$ and $a = a^*_H$, and $R(s\,a) = \text{Ber}(0.5)$, otherwise. $\mathcal{A}$ denotes any action.

shown in Krishnamurthy, Agarwal, et al. (2016), learning near-optimal policies can require an exponential number of episodes, in the worst case. The worst-case instance used for this result is shown in fig. 5.4. This POMDP is a simple combinatorial lock, which generates null rewards up to the last states. This decision process is very complex to solve because a single observation is associated to both states at each time step, and the agent receives no information about its progress up to the last action. After the complete sequence, the environment produces a positively biased reward if the sequence of actions is $a^*_1, a^*_2, \dots, a^*_H$, while the rewards are uniform in $\{0, 1\}$, for any other history. Because of this unobservable behaviour, the only way to discover the winning sequence of actions is to try all sequences.

This thesis tries to follow a generic treatment of decision processes, by focusing on the induced probability over traces. This view has been strongly influenced by Hutter (2009), which perfectly emphasises how decision processes are formalizations of, sometimes complex, environment dynamics. In this view, the Markov state is only a sufficient statistics of the historical information, that allows us to accurately predict future observations and rewards. This idea was then expanded in later works (Lattimore, Hutter, et al. 2013; Hutter 2014, 2016). In Majeed and Hutter (2018), the authors showed that any sufficient statistics for rewards, but not for observations, is also a feasible state for learning algorithms. Although the agent may not predict the observations accurately, rewards can be predicted. As the authors proved, a simple algorithm such as Q-learning converges towards the optimal policy using these states. The idea of approximate information states has been recently studied in Subramanian, Sinha, et al. (2022).

The environments that we consider here exhibit complex non-Markovian dependencies both in observations and in rewards. Restricting our attention to rewards, there is a long line of research studying non-Markovian reward specifications (Bacchus, Boutilier, et al. 1996; Brafman, De Giacomo, and Patrizi 2018; Icarte, T. Klassen, et al. 2018; Camacho, Icarte, et al. 2019; De Giacomo, Iocchi, et al. 2019;

De Giacomo, Favorito, et al. 2020; Icarte, T. Q. Klassen, et al. 2022). With the due differences, in the literature these approaches are called Rewards Machines (RMs) or Restraining Bolts. Similarly to RDPs, these are also automata-based formalisms. However, apart from targeting rewards specifically, the fundamental difference between the topic discussed in this thesis and these works is that rewards specifications are defined by a human, and therefore, they are known to the agent. The dynamics considered here, instead, is unknown. This difference is crucial for learning and planning complexity.

In Icarte, Waldie, et al. (2019), the authors draw connections between automata-based reward specifications (RMs) and POMDPs. They observe that some automata provide sufficient memory for planning in some POMDPs. These findings are also relevant for this thesis. Unlike this work, however, this thesis targets Regular Decision Processes, for multiple reasons. Firstly, RDPs are natively designed to handle both non-Markovian rewards and observations; secondly, RDPs describe a specific property of the environment dynamics, and are not tied to a single formalisation; lastly, RDPs are decision processes, meaning, environment models, akin to MDPs, POMDPs, etc; in fact, in their standard form, they do not include human-specific parts such as labelling functions.

In this paragraphs, we reviewed some of the related literature for the theoretical results about partially-observable environments. For other references, and those more related to learning algorithms, the reader can refer to the related work section of chapter 6.

5.2 Preliminaries

This chapter relies on the global preliminaries in chapter 2. In particular, we will be using histories, traces, and values for the infinite-horizon setting, as well as all the decision processes classes defined in section 2.2. Since this chapter discusses more general models, we do not adopt the specific conventions that have been used in part II for MDPs. In particular, MDP states will be denoted as observations $o \in \mathcal{O}$.

Planning in POMDPs

As defined in section 2.2, a POMDP is a tuple $\mathbf{P} \coloneqq \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}, T, R, O \rangle$, with states $\mathcal{S}$, actions $\mathcal{A}$, rewards $\mathcal{R}$, observations $\mathcal{O}$, transition function $T : \mathcal{S}\mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S}\mathcal{A} \to \Delta(\mathcal{R})$ and observation function $O : \mathcal{S} \to \Delta(\mathcal{O})$. The initial state distribution is $\mu \coloneqq T(s_\circ\, a_\circ)$. For simplicity, we assume that all the sets above are finite, and the POMDP is tabular.

A fundamental object associated with POMDPs is the concept of *belief*. With this term, we may refer to any probability distribution over the state space. The belief space is $\mathcal{B} \coloneqq \Delta(\mathcal{S})$. This is an important concept, because only the history of actions and observations is observable and the agent can only rely on stochastic estimates of the current state. For constructing such a state estimate, the agent may use the whole sequence of actions and observations produced so far. Since rewards also depend on the hidden state, similarly to observations, they are also informative of the unknown state. So, it may be sensible to compute beliefs over the full trace of observations actions and rewards. However, in order to be consistent with the POMDP literature, we do not explore this possibility here, and we only regard actions and observations as observable, in the strict sense. This motivates our initial distinction between histories and traces in section 2.1. However, we observe that they may always be incorporated within the observation space, whenever needed.

A *belief state* is the belief associated with the posterior distribution computed up to the current time step. Specifically, the belief state of some POMDP $\mathbf{P}$ at time $t \in \mathbb{N}$, history $h_t \in \mathcal{H}_t$, and observation $o_t \in \mathcal{O}$ is $\mathbb{P}(s_t \mid h_t o_t, \mathbf{P})$. We recall that the history $h_t = o_0 a_1 \dots o_{t-1} a_t$ ends with the last action. Using common terminology for POMDPs, we refer to $\mathbb{P}(s_t \mid h_t)$ as the "prior" belief and $\mathbb{P}(s_t \mid h_t o_t)$ as the "posterior", because the latter is obtained after conditioning on o_t.

The posterior belief state at time t will be written as $b_t \in \mathcal{B}$. The belief states can be updated from one time step to the next, after each action a and observation o, via the belief update function $U : \mathcal{B} \times \mathcal{A} \times \mathcal{O} \to \mathcal{B}$. In turn, this can be decomposed in two operators, $U_T : \mathcal{B} \times \mathcal{A} \to \mathcal{B}$ and $U_O : \mathcal{B} \times \mathcal{O} \to \mathcal{B}$, and written as $U(b, a, o) = U_O(U_T(b, a), o)$, for:

$$U_T(s' \mid b, a) \coloneqq \sum_{s \in \mathcal{S}} T(s' \mid sa)\, b(s) \tag{5.2}$$

$$U_O(s \mid b, o) \coloneqq \frac{O(o \mid s)\, b(s)}{\sum_{s' \in \mathcal{S}} O(o \mid s')\, b(s')} \tag{5.3}$$

Sometimes we also write $U_O(s \mid b, o) \propto O(o \mid s)\, b(s)$, for hiding the unique normalization factors that are constant in the main argument. In summary, we have $b_t = U(b_{t-1}, a_t, o_t)$. This update is the transformation needed to update the posteriors from $\mathbb{P}(s_{t-1} \mid h_{t-1} o_{t-1})$ to $\mathbb{P}(s_t \mid h_t o_t)$. This process is illustrated in fig. 5.5. The posterior belief b_t associated to $\mathbf{P}$ and sequence $h_t o_t$ is computed recursively as $U^*(h_t o_t) \coloneqq U(U^*(h_{t-1} o_{t-1}), a_t, o_t)$, and $U^*(o_0) \coloneqq U_O(\mu, o_0)$.

Definition 5.1. The *belief construction* of a POMDP $\mathbf{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}, T, R, O \rangle$ is an MDP $\mathbf{M}_b = \langle \mathcal{B}, \mathcal{A}, \mathcal{R}, T_b, R_b \rangle$, with:

$$\mu_b(b) = T(b \mid s_\circ a_\circ) \coloneqq \mathbb{P}(b_0 \mid \mathbf{P}) \tag{5.4}$$

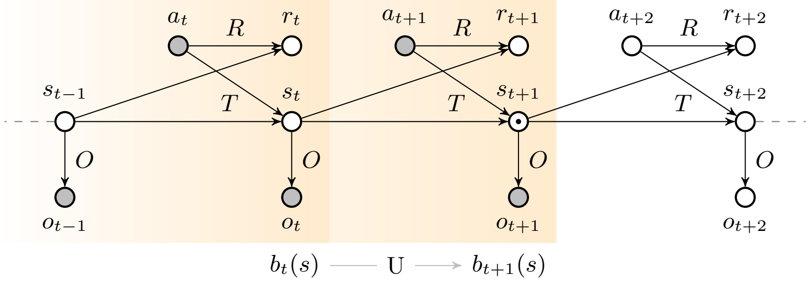


Figure 5.5. Illustration of the posterior belief update. White nodes are not observed.

$$= \sum_{o \in \mathcal{O}} \sum_{s \in \mathcal{S}} \delta_{U_O(\mu,o)}(b)\, O(o \mid s)\, \mu(s) \tag{5.5}$$

$$T_b(b' \mid b\, a') \coloneqq \mathbb{P}(b' \mid b, a', \mathbf{P}) \tag{5.6}$$

$$= \sum_{o' \in \mathcal{O}} \sum_{s,s' \in \mathcal{S}} \delta_{U(b,a',o')}(b')\, O(o' \mid s')\, T(s' \mid sa')\, b(s) \tag{5.7}$$

$$R_b(r' \mid ba') \coloneqq \mathbb{P}(r' \mid b, a', \mathbf{P}) \tag{5.8}$$

$$= \sum_{s \in \mathcal{S}} R(r' \mid sa')\, b(s) \tag{5.9}$$

So, the belief construction is a fully observable decision process over the posterior beliefs. A fundamental property of this MDP is that it encodes all the information for decision-making. In particular, any action is optimal for a $\mathbf{P}$ in $h_t o_t$ if and only if it is optimal for $\mathbf{M}_b$ at state $U^*(h_t o_t)$. Note that the belief construction is an MDP with an infinite state space.

Covering number

Definition 5.2. Consider any metric space $\langle \mathcal{X}, l \rangle$, for some metric l. The *covering number*, $c_\mu(\mathcal{X}, l) \in \mathbb{N}$, is the smallest number such that there exists a set of points $\tilde{\mathcal{X}} \subset \mathcal{X}$, with $|\tilde{\mathcal{X}}| = c_\mu(\mathcal{X}, l)$, that satisfies:

$$\forall x \in \mathcal{X},\, \exists \tilde{x} \in \tilde{\mathcal{X}} :\, l(x, \tilde{x}) < \mu \tag{5.10}$$

The set $\tilde{\mathcal{X}}$ is called a μ-cover of $\langle \mathcal{X}, l \rangle$.

The covering number has been found to be an important complexity measure for POMDPs, where it is usually computed over the reachable space of the associated belief construction (Q. Liu, Chung, et al. 2022; Hsu, W. S. Lee, et al. 2007). In these cases, this is computed for $\mathcal{X} \subseteq \mathcal{B}$.

Planning in RDPs A very appealing property of RDPs is that it is very efficient to plan and compute optimal policies for them. In fact, if the RDP is known, it is possible to simulate its deterministic automaton and compute the hidden state without uncertainty. When the RDP is known, the automaton states can be regarded observable and the whole system evolves as an MDP (Brafman and De Giacomo 2019). More precisely, the observations and rewards are Markovian with respect to the automaton states $\mathcal{Q}$. As a result, it is possible to find the optimal policy of any RDP by planning over an associated MDP that has $\mathcal{Q}$ as observations. Therefore, planning in RDPs is P-complete (reduction from Brafman and De Giacomo (2019) and complexity from Papadimitriou and Tsitsiklis (1987)).

An important class of policies for RDPs is the set of regular policies. Given an RDP $\mathbf{R}$, a policy $\pi : \mathcal{HO} \to \Delta(\mathcal{A})$ is called *regular* if $\pi(ho) = \pi(h'o)$ whenever $\bar{\tau}(h) = \bar{\tau}(h')$, for all $h, h' \in \mathcal{H}$. Let $\Pi_{\mathbf{R}}$ denote the set of regular policies for $\mathbf{R}$. Regular policies exhibit powerful properties. First, under any regular policy, suffixes have the same probability of being generated for histories that map to the same RDP state. Second, for any RDP there exists at least one optimal policy that is regular and deterministic. The reader can find more properties about regular policies in Brafman and De Giacomo (2019) and section 6.2.

5.3 The Expressive Power of RDPs

In this section, we study the expressive power of Regular Decision Processes. In order to follow a generic approach, we define what it means to be equivalent, or to approximate, a decision process, not only RDPs. This definition simply encodes the fact that two processes can be regarded as equivalent if they induce the same conditional probability over observations and rewards. In this case, the two models are perfectly indistinguishable, based on their external behaviours. To express this simple idea, we use NMDPs as the common formalism for comparing diverse decision processes. In fact, Non-Markov Decision Processes, as defined in section 2.2, are the most general formalism possible, and allow for representing any history-dependent probability distribution.

Definition 5.3. Given a decision process $\mathbf{D}$ over observations $\mathcal{O}$, rewards $\mathcal{R}$ and actions $\mathcal{A}$, we define the *induced NMDP* of $\mathbf{D}$ as the NMDP $\mathbf{N} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$, with $\mu = \bar{T}(h_0) \coloneqq \mathbb{P}(o_0 \mid \mathbf{D})$, and, for all $h_t \in \mathcal{H}$, $\bar{T}(o_t \mid h_t) \coloneqq \mathbb{P}(o_t \mid h_t, \mathbf{D})$ and $\bar{R}(r_t \mid h_t) \coloneqq \mathbb{P}(r_t \mid h_t, \mathbf{D})$.

Except for minor differences in the function arguments, which can be added appropriately, we can immediately observe that each MDP and k-MDP is also the induced NMDP of itself. The same is also true for RDPs, since they are already defined as NMDPs. However, when starting from the automaton representation, it is convenient to explicitly write the associated probabilities over traces. So, the

NMDP induced by an RDP $\mathbf{R} = \langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$ is $\mathbf{N} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$, where, for each $h_t \in \mathcal{H}$, $\bar{T}(o \mid h_t) = \theta_{\mathsf{o}}(o \mid q_t)$ and $\bar{R}(r \mid h_t) = \theta_{\mathsf{r}}(r \mid q_t)$, where $q_t \coloneqq \bar{\tau}(q_0, h_t)$,

Lastly, we compute the NMDP associated with any POMDP. This is stated without proof, as it directly follows from the meaning of posterior beliefs, as computed in the preliminaries of this chapter.

Proposition 5.1. *The NMDP induced by a POMDP* $\mathbf{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}, T, R, O \rangle$ *is* $\mathbf{N} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$, *where, for each* $h_t \in \mathcal{H}$,

$$\bar{T}(o \mid h_0) = \sum_{s \in \mathcal{S}} O(o \mid s)\, \mu(s) \tag{5.11}$$

$$\bar{T}(o' \mid h_t a_{t+1} o_{t+1}) = \sum_{s,s' \in \mathcal{S}} O(o' \mid s')\, T(s' \mid s\, a_{t+1})\, b_t(s) \tag{5.12}$$

$$\bar{R}(r' \mid h_t a_{t+1} o_{t+1}) = \sum_{s \in \mathcal{S}} R(r' \mid s\, a_{t+1})\, b_t(s) \tag{5.13}$$

where the belief state is $b_t = U^*(h_t o_t)$.

These definitions allow us to compare decision processes in terms of their visible traces.

Definition 5.4. Two decision processes are *equivalent* iff their induced NMDPs coincide.

Equivalence also allows us to establish relations between classes of decision processes. For this purpose, we will use the names of the decision processes, such as MDP and POMDP, as classes. In these statements, k-MDP refers to the class of k-MDPs, for any positive k. In general, we write $\mathcal{C}_1 \subseteq \mathcal{C}_2$ if the class $\mathcal{C}_2$ is at least as expressive as $\mathcal{C}_1$. Precisely,

Definition 5.5. For two classes of decision processes, we write $\mathcal{C}_1 \subseteq \mathcal{C}_2$ iff, for all $\mathbf{D}_1 \in \mathcal{C}_1$, there exists some $\mathbf{D}_2 \in \mathcal{C}_2$ such that $\mathbf{D}_1$ and $\mathbf{D}_2$ are equivalent.

5.3.1 Strict Relations

The first known result on the topic of RDPs expressiveness is that these models are strictly more expressive than MDPs and k-MDPs. This has been first stated in Brafman and De Giacomo (2019). For completeness, we restate the result here using our notion of equivalence and inclusion, and we provide a complete proof.

Proposition 5.2. *k-MDP* $\subset$ *RDP*

Proof. See page 117. □

As a second important relation, Brafman and De Giacomo (2019) stated that POMDPs are at least as expressive as RDPs. In light of definition 5.5, we restate the result here, and we also provide a proof.

Proposition 5.3. *RDP ⊆ POMDP*

Proof. See page 118. □

We are now ready to show an original statement that proves that RDPs are strictly less expressive than POMDPs. In other words, there are decision processes that can be modelled as POMDPs, but not as RDPs.

Theorem 5.4. *RDP ⊂ POMDP.*

Proof. See page 118. □

The proof is based on a POMDP whose reachable belief space is infinite. This allows us to create marginal distributions that are impossible to generate in RDPs, which can only produce finite sets of output distributions. This leads to the conclusive strict hierarchy of models as:

$$\text{MDP} \subset \text{k-MDP} \subset \text{RDP} \subset \text{POMDP} \tag{5.14}$$

A similar result for this strict containment can also be obtained in policy space. Since optimal RDP policies are known to be regular, if we show that optimal POMDP policies are not regular, the two models are clearly distinct.

Proposition 5.5. *Consider the set $\mathcal{X} = \{hoa \in \mathcal{HOA} \mid \exists \pi^* : \pi^*(ho) = a\}$, composed of all histories that end in some optimal action. Then, there exists a POMDP in which $\mathcal{X}$ is not a regular language.*

Proof. See page 119. □

Both theorem 5.4 and proposition 5.5 might appear as strong negative results for RDPs, at first sight. However, because of the inherent complexity of partial observations, POMDPs should be solved only at near-optimality, in the general case. Therefore, one sensible question to ask is: can POMDPs be approximated by RDPs? This is not possible between k-MDPs and RDPs, for example, since they simply lack the expressive power to capture long-term dependencies. However, between RDPs and POMDPs, we cannot easily identify such a clear separation.

5.3.2 Belief Covers

With the purpose of constructing POMDP approximations, we immediately notice a significant difference. In RDPs, if the actions and observations are given, the transitions are deterministic. Generic POMDPs, on the other hand, do not have deterministic transitions. However, there is one general deterministic transformation in POMDPs: the belief update. Following this intuition, we aim to construct RDPs whose automaton states are to be interpreted as specific beliefs of the POMDP to approximate.

More precisely, due to the way RDPs are defined here, we will find a relationship between the RDP states and *prior beliefs*. We recall that the posterior beliefs are $b_t \coloneqq \mathbb{P}(s_t \mid h_t o_t) = U(h_t o_t)$. On the other hand, prior beliefs are written $p_t \coloneqq \mathbb{P}(s_t \mid h_t)$. Priors can be defined in a very similar way to posteriors, with a recursive update as $U_p^*(h_0) \coloneqq T(s_\circ a_\circ)$ and $U_p^*(h_t) \coloneqq U_p(U_p^*(h_{t-1}), o_{t-1}, a_t)$, with $U_p(h, o, a') \coloneqq U_T(U_O(h, o), a')$. Therefore, $p_t = U_p^*(h_t)$. Based on prior beliefs, we define a new POMDP construction as follows.

Definition 5.6. The *prior belief construction* (or simply, *prior construction*), of a POMDP $\mathbf{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, T, R, O \rangle$ is a POMDP $\mathbf{P}_p \coloneqq \langle \mathcal{S}_p, \mathcal{A}, \mathcal{O}, \mathcal{R}, T_p, R_p, O_p \rangle$. Each state $po \in \mathcal{S}_p \coloneqq \mathcal{B}\mathcal{O}$ represents the current prior belief p and the current observation o. Transition, reward and observation functions are defined as:

$$T_p(po \mid s_\circ \, a_\circ) \coloneqq \mathbb{P}(p_0 = p, o_0 = o \mid \mathbf{P}) \tag{5.15}$$

$$= \delta_\mu(p) \sum_{s \in \mathcal{S}} O(o \mid s) \, \mu(s) \tag{5.16}$$

$$T_p(p'o' \mid (po) \, a') \coloneqq \mathbb{P}(p', o' \mid p, o, a', \mathbf{P}) \tag{5.17}$$

$$= \delta_{U_p(p,o,a')}(p') \sum_{s' \in \mathcal{S}} O(o' \mid s') \, p'(s') \tag{5.18}$$

$$R_p(r' \mid (po) \, a') \coloneqq \sum_{s \in \mathcal{S}} R(r' \mid s \, a') \, U_O(s \mid p, o) \tag{5.19}$$

$$O_p(o' \mid (po)) \coloneqq \delta_o(o') \tag{5.20}$$

There are two important differences with respect to the classic belief construction of definition 5.1. First, although each state is composed of variables that may be observed, we adopt the POMDP formulation, instead of an MDP, so to distinguish between original observations and other quantities. This allows the induced NMDP of a prior construction to be defined over the original observations only. Second, prior beliefs, which are used in this definition, are predictions over the next state when an action has been performed, but the following observation has not been received yet: hence the name "prior". With reference to fig. 5.5, the prior p_{t+1} would be represented by changing o_{t+1} to be white.

As for the classic belief construction, the prior construction also preserves optimal actions. In fact, a stronger result can be stated:

Proposition 5.6. *Any POMDP is equivalent to its prior construction.*

Proof. See page 120. □

Prior beliefs and the prior construction are particularly useful for defining RDPs that approximate the POMDPs dynamics. Essentially, we will choose the RDP states to be appropriate points within the prior belief space. In case the reachable portion of this space is finite, we immediately obtain an equivalence result. The reachable prior belief space (prior space, for short) is defined as follows. Let $\mathbf{P}$ be a POMDP and let $\mathcal{H}' \subseteq \mathcal{H}$ be the set of reachable histores in $\mathbf{P}$ under any policy. The reachable prior space is $\mathcal{P} \coloneqq \{U_p^*(h)\}_{h \in \mathcal{H}'} \subseteq \mathcal{B}$.

Theorem 5.7. *Any POMDP, whose reachable prior space is finite, admits an RDP that is equivalent to it.*

Proof. See page 121. □

The equivalent RDP, constructed for the proof of theorem 5.7 has an automaton state space of $\mathcal{POA}$. However, in alternative representations, this may change slightly. In RDPs defined as Mealy machines, for example, the output function would also receive the last input symbol as $\theta_\mathsf{o} : \mathcal{Q} \times \mathcal{OA} \to \Delta(\mathcal{O})$. Thus, allowing the RDP state space to be defined simply as $\mathcal{P}$. The core intuition is, in fact, to define automaton states as carefully placed points within the reachable prior space. The other two symbols are only needed to generate properly conditioned rewards. Since observation space is also finite, however, if the posterior belief space is finite, then also the prior space is finite (since it is obtained after conditioning). This means that we can state the theorem above in a slightly different form.

Corollary 5.8. *Any POMDP with finite observation and action spaces, whose reachable belief space is finite, admits an RDP equivalent to it.*

When stated in this form, using posteriors, we can also find some relations with Icarte, Waldie, et al. (2019, Theorem 4.1). However, unlike Reward Machines, RDPs are decision processes, and we have the explicit expression of the equivalent environment model.

Ultimately, finite belief spaces can be caused by various assumptions, including deterministic transition functions and deterministic initial belief, block MDP assumptions, and all the behaviours that are more specific to RDPs, such as the one of fig. 5.2.

5.3.3 POMDP Approximations

The previous section has characterised which POMDPs can be captured by RDPs exactly. However, in light of the negative result of theorem 5.4, a more sensible objective to pursue is understanding whether RDPs can at least *approximate* the POMDPs which they do not exactly capture. Fortunately, as we will see, the answer is affirmative in many cases. We first formalise what it means for a decision process to approximate any other.

Definition 5.7. A decision process $\mathbf{D}_1$ is said to be an ϵ-*approximation* for another decision process $\mathbf{D}_2$ iff, $\mathbf{N}_1 = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}_1, \bar{R}_1 \rangle$ and $\mathbf{N}_2 = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}_2, \bar{R}_2 \rangle$, the respective NMDPs induced by $\mathbf{D}_1$ and $\mathbf{D}_2$, satisfy:

$$\mathbb{E}_{h_t \sim \mathbf{D}_2 | a_{1:t}} |\bar{T}_1(o \mid h_t) - \bar{T}_2(o \mid h_t)| \leq \epsilon \tag{5.21}$$

$$\mathbb{E}_{h_t \sim \mathbf{D}_2 | a_{1:t}} |\bar{R}_1(r \mid h_t) - \bar{R}_2(r \mid h_t)| \leq \epsilon \tag{5.22}$$

for all $t \in \mathbb{N}, a_{1:t} \in \mathcal{A}^t, o \in \mathcal{O}, r \in \mathcal{R}$.

Similarly to equivalence, approximation should also be defined solely on the visible quantities. This is why the comparison is made on the induced probabilities represented by the NMDPs. There are many other, equally valid, definitions of approximations, alternatives to the one used here. However, we emphasise a couple of interesting features of our notion. First, approximation implies that the two processes are related by similar discounted values in the following sense.

Proposition 5.9. *For any policy π, let V_1^π and V_2^π be the discounted values that π obtains in two NMDPs, $\mathbf{N}_1$ and $\mathbf{N}_2$, from the respective initial distributions. If $\mathbf{N}_1$ is an ϵ-approximation of $\mathbf{N}_2$, then π_1^*, the optimal policy in $\mathbf{N}_1$, satisfies*

$$V_2^* - V_2^{\pi_1^*} \leq \frac{2\epsilon|\mathcal{R}|}{1-\gamma} + \frac{2\gamma\epsilon|\mathcal{O}|}{(1-\gamma)^2} \tag{5.23}$$

Proof. See page 122. □

This result says that if $\mathbf{N}_1$ approximates $\mathbf{N}_2$, then the optimal policies computed from $\mathbf{N}_1$ are near-optimal when executed in $\mathbf{N}_1$. Note that the cardinalities of $\mathcal{R}$ and $\mathcal{O}$ only appear because definition 5.7 amounts to some L_∞ distance and can be omitted by adopting L_1 instead. However, the most interesting characteristic of definition 5.7, which does not rely on discount factors and optimal values, is that it quantifies over histories of unbounded length. This is a particularly strict requirement that can be easily interpreted in case of deterministic decision processes. In fact, if $\mathbf{D}_2$ is deterministic, then $\mathbf{D}_1$ should approximate its outputs with constant accuracy, even after arbitrarily long histories. For stochastic processes, however, we do allow some major discrepancy for histories that are very unlikely under the decision process to approximate. This motivates the asymmetric expectation $h_t \sim \mathbf{D}_2$.

What drives us to pursue such a strong approximation criterion is that many POMDPs, which cannot be exactly captured by RDPs, can, in fact, be approximated. This is even true for the POMDP of fig. 5.6, which has been used as a counterexample to prove theorem 5.4.

Proposition 5.10. *For any $0 \leq \epsilon_1 \leq 1$ and $\epsilon > 0$, there exists an RDP which is an ϵ-approximation of $\mathbf{P}_1$, the POMDP of fig. 5.6.*

Proof. See page 124. □

Motivated by this positive result, we aim to find sufficient conditions that guarantee the existence of approximating RDPs. By now, we know that the structure of the approximating RDP should reflect the structure of the prior construction. However, if the reachable prior space is infinite, the RDP states cannot cover these beliefs exhaustively. Inspired by finite covers of compact spaces, we observe that $\mathcal{B}$ is a probability simplex, which can be "covered" with a finite set of appropriately spaced points. The notion of "cover" may be derived from the classic one, the one used to define covering numbers from definition 5.2. Let $\tilde{\mathcal{P}} \subseteq \mathcal{P}$ represent some finite set of prior beliefs. Then, for any POMDP, we define its "covering RDP" as we did in the proof of theorem 5.7, with the difference that after each update the new belief is projected into $\tilde{\mathcal{P}}$. The projection operator $\tilde{f} : \mathcal{P} \to \tilde{\mathcal{P}}$, returns $\tilde{f}(p) \coloneqq \arg\min_{\tilde{p} \in \tilde{\mathcal{P}}} d(p, \tilde{p})$, that is, the point minimising the distance function $d : \mathcal{B} \times \mathcal{B} \to \mathbb{R}_+$. We will give more details about this function later. Then, the structure of a covering RDP is as follows.

Definition 5.8. Given a POMDP $\mathbf{P}$ and a finite set $\tilde{\mathcal{P}} \subseteq \mathcal{P}$, with associated projection $\tilde{f}$, the *covering RDP* of $\mathbf{P}$ into $\tilde{\mathcal{P}}$ is $\mathbf{R_P} = \langle \tilde{\mathcal{P}}\mathcal{OA}, \mathcal{OA}, \Omega, \tau, \theta, \tilde{p}_0 \rangle$, with $\tilde{p}_0 \coloneqq \tilde{f}(p_0)$, and transition and output functions chosen as:

$$\tau(\tilde{p}oa, o'a') \coloneqq (\tilde{f}(U_p(\tilde{p}, o, a')), o', a') \tag{5.24}$$

$$\theta_{\mathsf{o}}(o' \mid \tilde{p}oa) \coloneqq \sum_{s' \in \mathcal{S}} O(o' \mid s)\, \tilde{p}(s) \tag{5.25}$$

$$\theta_{\mathsf{r}}(r' \mid \tilde{p}oa) \coloneqq \sum_{s' \in \mathcal{S}} R(r' \mid sa)\, U_O(\tilde{p}(s), o) \tag{5.26}$$

Similarly to the RDP constructed in the proof of theorem 5.7, covering RDPs defined as Mealy machines, not Moore, would only require a state space of $\tilde{\mathcal{P}}$. Apart from these minor details, the main question that covering RDPs leave open regards the error accumulated after each projection step. In fact, each belief update might lead to portions of the belief space that are not accurately covered. In general, nothing prevents successive approximation errors from building up and diverging from real beliefs. This is especially possible after conditioning with very rare observations. Therefore, some form of stability is needed in the belief dynamics and there should be a stabilising effect preventing this divergence. In this thesis, we identify two sufficient conditions.

The first criterion we analyse is a notion called ξ-observability.

Definition 5.9. A POMDP is ξ-*observable* if its observation function satisfies

$$\|O(o \mid p_1) - O(o \mid p_2)\|_1 \geq \xi \, \|p_1 - p_2\|_1 \tag{5.27}$$

for any two beliefs $p_1, p_2 \in \mathcal{P}$, and some $\xi > 0$. The parameter ξ is called *value of observation*. With a slight abuse of notation, we wrote the marginal distribution over observations as $O(o \mid p) \coloneqq \sum_{s \in \mathcal{S}} O(o \mid s) \, p(s)$.

This definition has been introduced in Even-Dar, Sham M. Kakade, et al. (2007) and recently used in Golowich, Moitra, et al. (2022b,a). It is also connected with other assumptions from the literature, since any ξ-observable POMDP is at least $(\xi/\sqrt{|\mathcal{S}|})$ weakly-revealing. The weakly-revealing condition has been used in Q. Liu, Chung, et al. (2022).

According to definition 5.9, the differences in belief are reflected as small differences in the observation distributions. This assumption has a stabilising effect on beliefs, because, although each observation may not allow the identification of the hidden state, it contains some, possibly low, information content about it. However, there is still nothing to prevent approximation errors that are caused by misplaced RDP states. One possibility would be to require the points to be accurately spaced according to some Euclidean norm. However, the error introduced after each conditioning step cannot be uniformly bounded in all directions. For this reason, we choose a projection operator that keeps the Kullback–Leibler divergence small.

Definition 5.10. We say that a finite set $\tilde{\mathcal{P}}$ is a η-divergent cover of $\mathcal{P}$ if the maximum relative information between any belief and its approximation is bounded by η. In other words, for

$$\iota_{p\|p'} \coloneqq \sup_{s \in \mathcal{S}} \log \frac{p(s)}{p'(s)} \qquad \text{and} \qquad \tilde{f}(p) \coloneqq \arg\min_{\tilde{p} \in \tilde{\mathcal{P}}} \iota_{p\|\tilde{p}} \tag{5.28}$$

it satisfies $\iota_{p\|\tilde{f}(p)} \leq \eta$, $\forall p \in \mathcal{P}$.

Also, we will say that $\tilde{\mathcal{P}}$ is a (η, ν)-divergent cover if it is η-divergent and there is some minimum probability associated to each state, namely $\forall p \in \tilde{\mathcal{P}}, s \in \mathcal{S} : p(s) \geq$

ν. This is useful whenever we need to require divergent covers that are not too deterministic. We are finally ready to state the following result.

Theorem 5.11. *Given any ξ-observable POMDP $\mathbf{P}$, if $\tilde{\mathcal{P}}$ is a (η, ν)-divergent cover of the reachable prior space, the covering RDP of $\mathbf{P}$ into $\tilde{\mathcal{P}}$ is an ϵ-approximation of $\mathbf{P}$, for $\epsilon = \sqrt{\frac{4\eta}{\xi^2 \nu}}$, provided that $\nu < 2/\xi^2$.*

Proof. See page 126. □

This theorem is a positive result regarding a first general subclass of POMDPs that can be approximated by RDPs. We remind the reader that our notion of approximation is more demanding than others that can be found in the literature. This is important, because in the finite-horizon setting, POMDPs may be approximated by k-MDPs with arbitrary precision. This is the approach followed in Brafman and De Giacomo (2019, Theorem 3). Also, approximating POMDPs in belief space, rather than in history space, has a strong advantage when the reachable belief space is “small”, since we do not need to encode historical data explicitly, which is exponential on the horizon.

However, we should note that theorem 5.11 does not imply that it is possible to construct approximations for POMDPs with arbitrary accuracy $\epsilon > 0$, because, in general, η cannot be arbitrarily small for a given $\nu > 0$. This can be seen when a deterministic belief is passed to the projection function: The minimum probability of any state ν has an impact on the minimum divergence achievable η. However, we hypothesise that this may be an artefact of the proof structure, which uses an argument from Even-Dar, Sham M. Kakade, et al. (2007). Using more advanced concentration techniques, such as the ones of Golowich, Moitra, et al. (2022b), it should be possible to achieve a similar result for a generic ϵ. If possible, this would be achieved by η-divergent covers, with no lower bound to the minimum probability of a state.

Next, we consider a second sufficient condition for the existence of approximating RDPs that involves the transition function instead of the observation function. This definition appears in Boyen and Koller (1998), here adapted for POMDPs.

Definition 5.11. The *mixing rate* ρ of a POMDP $\mathbf{P}$ is

$$\rho \coloneqq \min_{s_1, s_2 \in \mathcal{S}} \min_{a \in \mathcal{A}} \sum_{s'} \min\{T(s' \mid s_1 a), T(s' \mid s_2 a)\} \tag{5.29}$$

In other words, the mixing rate is the minimum probability mass that any two states have in common, according to the transition function. This is a measure of the total probability that any two states assign for the same event. Unlike ξ-observability, positive mixing guarantees contraction of beliefs after marginalisation, instead of conditioning.

Lemma 5.12. *(Boyen and Koller 1998) Given a POMDP* $\mathbf{P}$ *with mixing rate* $\rho > 0$*, for any two beliefs* $p_1, p_2 \in \mathcal{P}$ *and action* $a \in \mathcal{A}$*, the relative entropy after a prediction step satisfies*

$$D_{KL}(U_T(p_1, a) \parallel U_T(p_2, a)) \leq (1 - \rho)\, D_{KL}(p_1 \parallel p_1) \tag{5.30}$$

This allows us to obtain an even stronger result than the one we have for observable POMDPs, that is, approximating RDPs with arbitrary accuracy.

Theorem 5.13. *For any* $\epsilon > 0$ *and POMDP* $\mathbf{P}$ *whose mixing rate is* $\rho > 0$*, there exists a covering RDP that is an* ϵ*-approximation of* $\mathbf{P}$*.*

Proof. See page 127. □

The mixing assumption is relatively restrictive because, for positive ρ, it severely constraints the transition function. However, together with the observability assumption, it allows us to identify an interesting pattern: both are sufficient conditions for contraction of beliefs from different initializations. Belief contraction has been described in Golowich, Moitra, et al. (2022b). We adapt it here for generic actions and prior beliefs. Let $U_p^*(p, h)$ be the belief update function, where the initial prior is set to p. We can say that beliefs contract in a POMDP $\mathbf{P}$ with respect to some divergence $D : \mathcal{B} \times \mathcal{B} \to \mathbb{R}_+$ if, for any initial history $h_t \in \mathcal{H}_t$, and a successive sequence of actions $a_{t+1} \dots a_n$,

$$\mathbb{E}_{h_n | h_t}\, D(U_p^*(\mathrm{Unif}(\mathcal{S}), h_n), U_p^*(p_t, h_n)) \leq g(n) \tag{5.31}$$

for a strictly decreasing function $g : \mathbb{N} \to \mathbb{R}_+$, and where D can be a divergence or a norm. Intuitively, belief contraction can compensate for erroneous initialisation of beliefs, where the wrong initial belief is represented by the uniform distribution. Essentially, it is acceptable to forget the initial belief after a sufficient number of correct updates. This is also relevant for covering RDPs, because, after each update, the projection operator introduces some error, which can be regarded as a tiny initialisation error for U. This discrepancy should contract in expectation, not diverge. Finally, we observe that expectation is essential since beliefs might diverge for some unlikely observations.

The class of POMDPs with contracting beliefs, which is closely related to "filter stability" (Kara and Yüksel 2022), might seem sufficiently general. However, we can reason that, if the beliefs contract, then the POMDP is not only approximated by an RDP, but also by some k-MDP, for sufficiently large k. Namely, its transition function could emulate the belief update from any stochastic prior distribution, that is $\bar{T}(o_t \mid h_t) \coloneqq O(o_t \mid U_p^*(\mathrm{Unif}(\mathcal{S}), h_{t-k}))$. Clearly, a covering RDP that reproduces the same dynamic could have a much more compact state space with respect to this k-MDP, in which it is $(\mathcal{O}\mathcal{A})^k$. However, the pure existence of approximating k-MDPs shows how limiting the assumption of contracting beliefs might be for POMDPs, when taken alone. Essentially, it prevents past transitions from having lasting impacts that extend arbitrarily far into the future. This behaviour, on the other hand, is exactly what characterises RDPs.

We can identify two classes. Let $\mathrm{POMDP}_C \subset \mathrm{POMDP}$ be the class of POMDPs with contracting beliefs (according to some D and g). Also, let $\mathrm{POMDP}_R \subset \mathrm{POMDP}$ be the class of POMDPs for which there exists an equivalent RDP. Then, we can expect that RDPs can approximate any decision process in the combination of the two classes. Consider any two $\mathbf{P}_C \in \mathrm{POMDP}_C$ and $\mathbf{P}_R \in \mathrm{POMDP}_R$, with their respective state spaces $\mathcal{S}_C$ and $\mathcal{S}_R$. It is possible to construct a new POMDP, obtained by composition of the two sets of features, whose state space is $\mathcal{S}_C\mathcal{S}_R$, in which beliefs do not contract, but can be approximated by some RDP. In fact, only the beliefs in $\mathcal{S}_C$ contract, while those in $\mathcal{S}_R$ do not. However, when initialised correctly, the hidden states $\mathcal{S}_R$ can be tracked exactly through some transducer. More generally, as long as the beliefs in $\mathcal{S}_C$ contract, it is also possible to introduce additional dependencies from the features in $\mathcal{S}_R$, in the form of

$$T(s'_c s'_r \mid s_c s_r a) = T_c(s'_c \mid s_c s_r a)\, T_r(s'_r \mid s_r a) \tag{5.32}$$

In this section, we discussed some general sufficient conditions for the existence of approximating RDPs. Some dynamics can be captured exactly, while others can only be approximated. One remaining open question is the following: is there any POMDP which cannot be approximated by some RDP? If this question could be answered negatively, RDPs would be generic approximators for POMDPs. In light of related positive results in the literature (Yu and Bertsekas 2008), we currently believe that this generic approximation result could be true for RDPs.

5.3.4 Learning With Partial Observations via RDPs

Regular Decision Processes are interesting environment models, and studying their expressive power is relevant in its own right. One second motivation for addressing this problem is the development of alternative learning algorithms for POMDPs. POMDPs are very expressive. With just a few variables, they can capture very complex environment dynamics. This behaviour is not specific to POMDPs, but is shared with many other models with latent variables (Murphy 2012). Thanks to compactness, it is very convenient for a person to specify them, in case some dependencies are known.

These advantages, on the other hand, do not imply that POMDPs are the best suitable models for learning. In a pure learning setting, the agent observes a stream of variables, or multiple traces of them, whose probability depends on past actions and observations in ways that are hard to predict. The POMDP approach suggests that the agent should assume the existence of an unknown number of hidden variables, which might justify the observations. However, since the transition between the new hidden states is also stochastic, the agent should also estimate the current state after learning. In an RDP formulation of the same dynamics, instead, a state would only be generated if it serves to distinguish two conditional distributions. Importantly, thanks to deterministic transitions, no state estimation is necessary since they can be tracked with certainty after learning. In this subsection, we argue that RDPs are suitable models for performing model-based RL over POMDPs.

As we have seen in section 5.3.3, no POMDP is known at the moment, which cannot be approximated by some RDP. In general, let $\mathrm{POMDP}_r \subseteq \mathrm{POMDP}$ be the class of POMDPs for which approximating RDPs exist. The above results, and definition 5.7, seem to suggest that no learning algorithm relying on probabilistic estimates can distinguish if the observations are generated by a POMDP or a carefully constructed RDP with a sufficiently large state space. In other words, RDP learning algorithms may also provide original methods for finding near-optimal solutions in the presence of partial observations. In fact, especially for equivalence, RDP learning algorithms can be applied to POMDP with finite reachable beliefs, without modifications. Ultimately, a similar possibility also opens for all POMDPs that RDPs approximate.

Clearly, it is also important to understand whether learning in RDPs is fundamentally easier than in POMDPs. This does not seem to be the case for generic RDP instances, as we show in the result below.

Theorem 5.14. *Consider the fixed-horizon setting with horizon $H \in \mathbb{N}_+$, and fix $A \geq 2$, $\varepsilon \in (0, \sqrt{1/8})$. For any RL algorithm, there exists an RDP with A actions and at most $3HA$ total states, such that the probability that the algorithm outputs an ε-optimal policy after $T \leq cA^H/\varepsilon^2$ episodes is at most $2/3$, where $c > 0$ is a global constant.*

Proof. The result follows from the sample complexity lower bound for POMDPs from proposition 1 of Krishnamurthy, Agarwal, et al. (2016). The POMDP instance used in their proof, $\mathbf{P}$, is shown in fig. 5.4. We observe that this is an RDP. More precisely, there exists and RDP $\mathbf{R}$ that is equivalent to it. Since $\mathbf{P}$ has a single initial state and its transition function is deterministic, the reachable prior belief space $\mathcal{P}$ is finite. Then, equivalence follows from theorem 5.7 and the number of RDP states of the equivalent RDP comes from the proof of this theorem. In particular, since both the initial belief and the POMDP transitions are deterministic, the cardinality of the reachable prior space $\mathcal{P}$ coincides with $|\mathcal{S}| = 2H$. Then, there exists an equivalent

RDP with $\mathcal{POA} = 3HA$ states. Ultimately, equivalence means that $\mathbf{P}$ and $\mathbf{R}$ cannot be distinguished from traces, and if theorem 5.14 was false, then proposition 1 of Krishnamurthy, Agarwal, et al. (2016) would be false. □

Due to this theorem, we see that learning with RDPs may not be strictly more effective in the general case. However, the approach of learning in POMDPs using RDP representations is particularly relevant for model-based learning. In fact, while planning in POMDPs is PSPACE-hard, planning in RDPs is in P. This allows algorithms to maintain very effective state representations, which are convenient for planning, since they do not require belief estimations, which are computationally demanding. This does not contradict the hypothesis that RDPs can be general POMDP approximators because it might be necessary to have an exponential number of RDP states in the general case.

5.4 Discussion

In this chapter, we have shown multiple results regarding the expressive power of RDPs. Among the negative results, that is, those that identify a clear separation between RDPs and POMDPs, we demonstrated theorem 5.4, showing the strict containment RDP $\subset$ POMDP, and proposition 5.5, for an analogous separation in policy space. Among the positive results, on the other hand, we have characterised in theorem 5.7 the POMDPs that admit equivalent RDPs. Moreover, we have shown in theorems 5.11 and 5.13 that both ξ-observable and ρ-mixing POMDPs can be approximated by RDPs in the infinite horizon. More in general, these two last results suggest that POMDPs with stable belief dynamics, which would include the two classes above, could be approximated by RDPs. Together with the class of POMDPs that RDPs exactly capture, the relationship between the two classes becomes very strong. Currently, whether RDPs are universal POMDP approximators remains an open question. In fact, although this work does not give a conclusive statement to this issue, it also fails to identify a counterexample that would clearly separate the two classes in approximation.

The relationship between these two models is especially tight in light of the lower bound of theorem 5.14. In fact, this exponential lower bound is shared by both RDPs and POMDPs. The main difference between these two models is planning complexity, which is only polynomial for RDPs.

Most of the positive results of this chapter have been obtained by relating the RDP states to a finite set of beliefs of the prior construction. In this regard, prior beliefs have been the appropriate target to approximate because, similarly to RDP states, they represent the environment configurations after the agent has taken an action. Some RDP representations, such as the one that will be used in chapter 6, are more agent-oriented, in the sense that each RDP state corresponds to a decision point, after the environment has selected an observation. In this case, RDP states would be more properly related to a finite set of posterior beliefs, instead.

Future Work Many directions remain unexplored and are excellent candidates for future work. Although we gave multiple results and insights on this topic, the question of whether RDPs are general approximators for POMDPs still remains partially open. This could be resolved both positively and negatively, with interesting directions in either case. This result would also largely depend on the specific definition of the approximation adopted. The one proposed here in definition 5.7 is only one of various possibilities.

This topic has an impact on RL algorithms for both RDPs and POMDPs. However, we did not study the practical applicability of RDP algorithms to POMDPs, nor vice versa. An RDP learning algorithm will be proposed in chapter 6.

5.5 Proofs

This section contains all the proofs for this chapter. The reader may skip this section and refer to it as needed.

Proposition 5.2. *k-MDP ⊂ RDP*

Proof. We first show k-MDP ⊆ RDP. Let $\mathbf{M} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, T, R \rangle$ be any $\mathbf{M} \in k$-MDP for a specific $k \in \mathbb{N}_+$. We define the RDP that stores the last k state-action tuples in its automaton states as $\mathbf{R} \coloneqq \langle \mathcal{H}_k, \mathcal{OA}, \Omega, \tau, \theta, q_0 \rangle$, for $q_0 \coloneqq s_\circ a_\circ \dots s_\circ a_\circ$. For any $h_k = o_{t-k} a_{t-k+1} \dots a_t \in \mathcal{H}_k$, the transition and output functions are:

$$\tau(h_k, oa') \coloneqq o_{t-k+1} a_{t-k+2} \dots a_t o a' \tag{5.33}$$

$$\theta_{\mathsf{o}}(h_k) \coloneqq R(h_k) \tag{5.34}$$

$$\theta_{\mathsf{r}}(h_k) \coloneqq T(h_k) \tag{5.35}$$

$\mathbf{M}$ and $\mathbf{R}$ are clearly equivalent, because both models compute the output distributions from T and R and the last k transitions.

To show k-MDP ≠ RDP, we can just construct an RDP whose dynamics have history dependency that extends arbitrarily far into the future. We choose, $\mathbf{R} \coloneqq \langle \mathcal{Q}, \mathcal{OA}, \Omega, \tau, \theta, q_0 \rangle$, with $\mathcal{Q} \coloneqq \{q_0, q_1\}$, $\mathcal{O} \coloneqq \{\top, \bot\}$, and a single action $\mathcal{A} \coloneqq \{a\}$. The RDP memorizes whether $\bot$ was observed at least once along the history: $\tau(q_0, \top a) = q_0$, $\tau(q_0, \bot a) = q_1$, and $\tau(q_1, oa) = q_1$. The output probabilities depend on this condition: $\theta_{\mathsf{o}}(\top \mid q_0, oa) = 0.5$ and $\theta_{\mathsf{o}}(\top \mid q_1, oa)_\top = 0.4$. Now, for any $k \in \mathbb{N}_+$ and k-MDP $\mathbf{M}$, consider two histories $h_{k+1} \coloneqq \bot a (\top a)^k$ and $h'_{k+1} \coloneqq \top a (\top a)^k$. Under $\mathbf{R}$, the probability of the following $\top$ differs in the two histories, $\mathbb{P}(\top \mid h_{k+1}, \mathbf{R}) \neq \mathbb{P}(\top \mid h'_{k+1}, \mathbf{R})$. Since this cannot be captured with the most k recent observations in h_{k+1} and k'_{k+1}, $\mathbf{M}$ cannot represent the probability induced by $\mathbf{R}$. This proof can be repeated for any k. □

Proposition 5.3. *RDP ⊆ POMDP*

Proof. Consider any RDP $\mathbf{R} = \langle \mathcal{Q}, \mathcal{OA}, \Omega, \tau, \theta, q_0 \rangle$. We define a POMDP $\mathbf{P} \coloneqq \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}, T, R, O \rangle$, in which $\mathcal{S} \coloneqq \mathcal{QO}$, and:

$$T(qo \mid s_\circ a_\circ) \coloneqq \delta_{q_0}(q)\, \theta_{\mathsf{o}}(o \mid q_0) \tag{5.36}$$

$$T(q'o' \mid (qo)\, a') \coloneqq \delta_{\tau(q, oa')}(q')\, \theta_{\mathsf{o}}(o' \mid q') \tag{5.37}$$

$$O(o' \mid qo) \coloneqq \delta_o(o') \tag{5.38}$$

$$R(r' \mid (qo)\, a') \coloneqq \theta_{\mathsf{r}}(r' \mid q) \tag{5.39}$$

We now show that $\mathbf{R}$ and $\mathbf{P}$ are equivalent. To do so we need to compute the respective induced NMDPs, $\mathbf{N_R}$ and $\mathbf{N_P}$. First, from eqs. (5.36) and (5.38), the distribution of o_0 is $\theta_{\mathsf{o}}(q_0)$ under both models. Next, we compute the beliefs of $\mathbf{P}$, which are required in proposition 5.1 for computing $\mathbf{N_P}$. Since the transition function T is deterministic in $\mathcal{Q}$ and stochastic in $\mathcal{O}$, but the stochastic component is observable, we recognize that beliefs of $\mathbf{P}$ are fully deterministic. In particular, we can show by induction that $b_t(qo) = \delta_{o_t}(o)\, \delta_{q_t}(q)$, by starting from the initial (deterministic) belief, and recursively applying the update function. Now that the expression for the beliefs of $\mathbf{N_P}$ is known, we can compute its transition and reward probabilities and verify equivalence:

$$\mu_{\mathbf{N_P}} = \theta_{\mathsf{o}}(q_0) = \mu_{\mathbf{N_R}} \tag{5.40}$$

$$\bar{T}_{\mathbf{N_P}}(o \mid h_t) = \theta_{\mathsf{o}}(o \mid q_t) = \bar{T}_{\mathbf{N_R}}(o \mid h_t) \tag{5.41}$$

$$\bar{R}_{\mathbf{N_P}}(r \mid h_t) = \theta_{\mathsf{r}}(r \mid q_t) = \bar{R}_{\mathbf{N_R}}(r \mid h_t) \tag{5.42}$$

□

Theorem 5.4. *RDP ⊂ POMDP.*

Proof. As a consequence of proposition 5.3, we prove we only need to show that POMDP $\neq$ RDP. To do so, we construct a POMDP for which no equivalent RDP exists. Consider the POMDP $\mathbf{P}_1 = \langle \{s_0, s_1, s_2\}, \{a_0, a_1\}, \mathcal{R}, \{\top, \bot\}, T, R, O \rangle$, illustrated in fig. 5.6. Each arc is labelled with an action and the associated

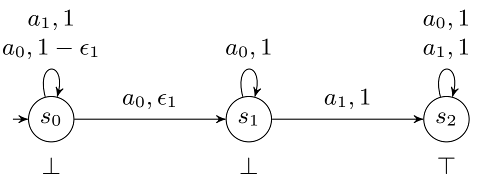


Figure 5.6. POMDP used in proof of theorem 5.4.

probability for that transition. The observations are deterministic and shown below each node.

For arbitrarily long repetitions of a_0, $\mathbf{P}_1$ only assigns positive probability to equally long sequences of observations $\bot$. Now consider what is the probability assigned to $\top$ if a_1 is executed at the end of this sequence. Let $\langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$ be the NMDP induced by $\mathbf{P}_1$. We are asking what is the value of $\bar{T}(\top \mid h_{t+1})$ for $h_{t+1} = (\bot\, a_0)^t \bot\, a_1$. According to equation (5.12), it is necessary to evaluate the belief at time t, as $b_t = U^*((\bot\, a_0)^t)$. Since observations provide no information, each application of U_O leaves the belief unchanged and we can only consider the impact of U_T. This leads to the probability vector $b_t = [(1-\epsilon_1)^t, 1-(1-\epsilon_1)^t, 0]$. From here, we know that after executing the next action, a_1, $\bar{T}(\top \mid h_{t+1}) = b_t(s_1) = 1-(1-\epsilon_1)^t$.

This shows that, for $0 < \epsilon_1 < 1$, for histories $(\bot\, a_0)^* \bot\, a_1$ of increasing length, the probability of $\top$ increases without ever repeating. On the other hand, RDPs can only generate one in a finite number of distributions, one for each state in the finite set $\mathcal{Q}$. Thus, there is no RDP which that is equivalent to $\mathbf{P}_1$. □

Proposition 5.5. *Consider the set $\mathcal{X} = \{hoa \in \mathcal{HOA} \mid \exists \pi^* : \pi^*(ho) = a\}$, composed of all histories that end in some optimal action. Then, there exists a POMDP in which $\mathcal{X}$ is not a regular language.*

Proof. We define a POMDP $\mathbf{P}_3 = \langle \mathcal{Q}, \mathcal{A}, \mathcal{O}, T, R, O \rangle$, with states $\mathcal{Q} = \{q_s, q_+, q_-, q_w, q_l\}$, actions $\mathcal{A} = \{\bar{a}_0, \bar{a}_1, \bar{a}_2\}$, observations $\mathcal{O} = \{S, +, -, W, L\}$. Intuitively, the observation are associated to the start state q_s, to states q_+, q_-, and the winning and losing states q_w, q_l. Transitions and reward functions are defined as in fig. 5.7. The stochastic transitions are shown in the arc labels and $\mathcal{A}$ denotes any action. The symbol δ_x is the deterministic distribution at x, and v_ξ is the distribution defined as $v_\xi(+) = (1+\xi)/2$ and $v_\xi(-) = (1-\xi)/2$. We assume $0 < \xi < 0.5$. The reward function is zero everywhere except for any action in state q_w.

As we can see, the first transition uniformly leads to q_+ or q_-. The optimal action from each of these states is $\bar{a}_1$ and $\bar{a}_2$, respectively. Intuitively, a good policy should first execute $\bar{a}_0$ for a number of time steps to receive observations. The exact

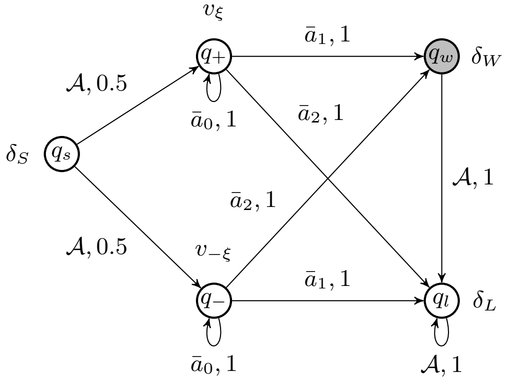


Figure 5.7. The POMDP used in the proof of proposition 5.5.

number depends on the discount factor. This allows to collect sufficient evidence from the observations about whether the current state is q_+ or q_-.

Now, we restrict our attention to the sequences in $\mathcal{HO}$ that only contains action $\bar{a}_0$. Under this sequence of actions, the only set of possible sequences is $\mathcal{X}_\pm \coloneqq \{S\}\,(\{\bar{a}_0\}\,\{+,-\})^*$, all of which have a positive probability of occurring. Let $\mathcal{X}_{\bar{a}_1} \subseteq \mathcal{X}_\pm$ be the set sequences for which $\bar{a}_1$ is optimal. We show that $\mathcal{X}_{\bar{a}_1}$ is not a regular language. First, we construct $b(q_+ \mid h_t o_t)$, the belief associated to any $h_t o_t \in \mathcal{X}_\pm$, computed at q_+.

$$b(q_+ \mid h_{t-1} o_{t-1} \bar{a}_0 o_t) \propto O(o_t \mid q_+) \sum_{q \in \mathcal{Q}} T(q_+ \mid q \bar{a}_0)\, b(q \mid h_{t-1} o_{t-1}) \tag{5.43}$$

$$= O(o_t \mid q_+)\, b(q_+ \mid h_{t-1}) \tag{5.44}$$

We recall that $O(+ \mid q_+) = (1+\xi)/2$ and $O(- \mid q_+) = (1-\xi)/2$. Then, expanding over time,

$$b(q_+ \mid h_t o_t) \propto \frac{1}{2}\left(\frac{1+\xi}{2}\right)^{n_+} + \frac{1}{2}\left(\frac{1-\xi}{2}\right)^{n_-} \tag{5.45}$$

where n_+, n_- is the number of occurrences in h_t of $+$, $-$, respectively, and we have used the fact that, at the initial transition, $b(q_+ \mid S) = 1/2$. Similarly,

$$b(q_- \mid h_t o_t) \propto \frac{1}{2}\left(\frac{1+\xi}{2}\right)^{n_-} + \frac{1}{2}\left(\frac{1-\xi}{2}\right)^{n_+} \tag{5.46}$$

for the same normalizing factor. Then, the action $\bar{a}_1$ is optimal at $h_t o_t$ iff $b(q_+ \mid h_t o_t) \geq b(q_- \mid h_t o_t)$. This is true iff $n_+ \geq n_-$. This means $\mathcal{X}_{\bar{a}_1}$ is the subsets of sequences $ho \in \mathcal{X}_\pm$ for which the number of occurrences of $+$ is higher of equal than the number of $-$. We have that both $\mathcal{X}_{\bar{a}_1}$ and $\mathcal{X}_{\bar{a}_1}\{\bar{a}_1\}$ are strictly context-free. □

Proposition 5.6. *Any POMDP is equivalent to its prior construction.*

Proof. The statement might be verified by simply observing that the prior construction preserves the original conditional probabilities over observations and rewards. For a more formal proof, we should follow the procedure outlined below.

Consider a POMDP $\mathbf{P} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, O \rangle$ and its prior construction $\mathbf{P}_p = \langle \mathcal{S}_p, \mathcal{A}, \mathcal{O}, T_p, R_p, O_p \rangle$. Let $\mathbf{N} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R} \rangle$ and $\mathbf{N}_p = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}_p, \bar{R}_p \rangle$ be the respective induced NMDPs. To prove equivalence, we need to show that $\bar{T} \equiv \bar{T}_p$ and $\bar{R} \equiv \bar{R}_p$, for all histories. For comparing each pair of functions we should compute the transition and reward functions of the two NMDPs. As dictated by eqs. (5.12) and (5.13), this requires computing the expressions for the posterior beliefs of $\mathbf{P}$ and of $\mathbf{P}_p$. In particular, this last belief space would be $\Delta(\mathcal{S}_p)$, where $\mathcal{S}_p$ includes itself a distribution. However, we observe that the computed posterior would collapse to being deterministic in the first component, since prior beliefs are deterministic given the actions and the observations. The remaining algebraic substitutions are omitted. □

Theorem 5.7. *Any POMDP, whose reachable prior space is finite, admits an RDP that is equivalent to it.*

Proof. We know that the set $\mathcal{P} \coloneqq \{U_p^*(h)\}_{h \in \mathcal{H}'}$ is finite. Hence, it is possible to define an RDP having a state space of $\mathcal{POA}$ that mimics the behaviour of the prior construction. Let us define an RDP $\mathbf{R} \coloneqq \langle \mathcal{POA}, \mathcal{OA}, \Omega, \tau, \theta, p_0 oa \rangle$, with initial state composed of the prior belief $p_0 = \mu = T(s_\circ\, a_\circ)$, and any observation and action oa. The transition function updates the prior belief and stores the new observation action pair as $\tau(poa, o'a') \coloneqq (U_p(p, o', a'), o', a')$. The transitions and rewards are defined as:

$$\theta_{\mathsf{o}}(o' \mid poa) \coloneqq \sum_{s \in \mathcal{S}} O(o' \mid s)\, p(s) \tag{5.47}$$

$$\theta_{\mathsf{r}}(r' \mid poa) \coloneqq \sum_{s \in \mathcal{S}} R(r' \mid s\, a)\, U_O(s \mid p, o) \tag{5.48}$$

This RDP perfectly mimics the dynamics of the prior construction. In fact, by comparing the respective induced NMDPs of the RDP $\mathbf{R}$ and the original POMDP $\mathbf{P}$, we can see that the probabilities from all histories are the same. To verify this, we remind that $b_t = U_O(p_t, o_t)$ and $p_t = U_T(b_{t-1}, a_t)$. □

Proposition 5.9. *For any policy π, let V_1^π and V_2^π be the discounted values that π obtains in two NMDPs, $\mathbf{N}_1$ and $\mathbf{N}_2$, from the respective initial distributions. If $\mathbf{N}_1$ is an ϵ-approximation of $\mathbf{N}_2$, then π_1^*, the optimal policy in $\mathbf{N}_1$, satisfies*

$$V_2^* - V_2^{\pi_1^*} \leq \frac{2\epsilon|\mathcal{R}|}{1-\gamma} + \frac{2\gamma\epsilon|\mathcal{O}|}{(1-\gamma)^2} \tag{5.23}$$

Proof. Let π_1^* and π_2^* be optimal policies in the two NMDPs.

$$V_2^* - V_2^{\pi_1^*} = |V_2^{\pi_2^*} - V_2^{\pi_1^*}| \tag{5.49}$$

$$\le |V_2^{\pi_2^*} - V_1^{\pi_1^*}| + |V_1^{\pi_1^*} - V_2^{\pi_1^*}| \tag{5.50}$$

$$= |\sup_\pi V_2^\pi - \sup_\pi V_1^\pi| + |V_1^{\pi_1^*} - V_2^{\pi_1^*}| \tag{5.51}$$

$$\le \sup_\pi |V_2^\pi - V_1^\pi| + |V_1^{\pi_1^*} - V_2^{\pi_1^*}| \tag{5.52}$$

$$\le 2\sup_\pi |V_2^\pi - V_1^\pi| \tag{5.53}$$

now we apply lemma 5.15,

$$\begin{aligned}\le \frac{2}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^\pi}|\mathbb{E}_{r_t\sim\bar{R}(h_t)}\,r_t - \mathbb{E}_{r_t\sim\bar{R}'(h_t)}\,r_t| \\ + \frac{2\gamma}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^\pi}|\mathbb{E}_{o_t\sim\bar{T}(h_t)}\,V^\pi(h_to_t) - \mathbb{E}_{o_t\sim\bar{T}'(h_t)}\,V^\pi(h_to_t)|\end{aligned} \tag{5.54}$$

$$\begin{aligned}\le \frac{2}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^\pi}\sum_{r\in\mathcal{R}}|\bar{R}(r\mid h_t) - \bar{R}'(r\mid h_t)| \\ + \frac{2\gamma}{(1-\gamma)^2}\,\mathbb{E}_{h_t\sim d'^\pi}\sum_{o\in\mathcal{O}}|\bar{T}(o\mid h_t) - \bar{T}'(o\mid h_t)|\end{aligned} \tag{5.55}$$

using the fact that $\mathbf{N}_1$ approximates $\mathbf{N}_2$,

$$\le \frac{2\epsilon|\mathcal{R}|}{1-\gamma} + \frac{2\gamma\epsilon|\mathcal{O}|}{(1-\gamma)^2} \tag{5.56}$$

□

Lemma 5.15. *For any policy π and any two NMDPs $\mathbf{N}$ and $\mathbf{N}'$, let V_μ^π and $V_{\mu'}'^\pi$ be their respective values. Then,*

$$\begin{aligned}|V_\mu^\pi - V_{\mu'}'^\pi| \le \frac{1}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^\pi}|\mathbb{E}_{r_t\sim\bar{R}(h_t)}\,r_t - \mathbb{E}_{r_t\sim\bar{R}'(h_t)}\,r_t| \\ + \frac{\gamma}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^\pi}|\mathbb{E}_{o_t\sim\bar{T}(h_t)}\,V^\pi(h_to_t) - \mathbb{E}_{o_t\sim\bar{T}'(h_t)}\,V^\pi(h_to_t)|\end{aligned} \tag{5.57}$$

Proof. This result is the analogue of what the Simulation Lemma (Kearns and S. Singh 2002) is for MDPs. Let $\mathbf{N}$ and $\mathbf{N}'$ be two NMDPs, and V^π, V'^π their values functions for some policy π. Then, for any initial observation $o_0 \in \mathcal{O}$,

$$V^\pi(o_0) - V'^\pi(o_0) \tag{5.58}$$

$$\begin{aligned}= \mathbb{E}_{a_1\sim\pi(o_0)}\,\mathbb{E}_{r_1,o_1\sim\bar{R}(h_1),\bar{T}(h_1)}[r_1 + \gamma\,V^\pi(h_1o_1)] \\ + \mathbb{E}_{a_1\sim\pi(o_0)}\,\mathbb{E}_{r_1,o_1\sim\bar{R}'(h_1),\bar{T}'(h_1)}[r_1 + \gamma\,V'^\pi(h_1o_1)]\end{aligned} \tag{5.59}$$

$$\begin{aligned}= \mathbb{E}_{a_1\sim\pi(o_0)}[\mathbb{E}_{r_1\sim\bar{R}(h_1)}\,r_1 - \mathbb{E}_{r_1\sim\bar{R}'(h_1)}\,r_1] \\ + \gamma\,\mathbb{E}_{a_1\sim\pi(o_0)}[\mathbb{E}_{o_1\sim\bar{T}(h_1)}\,V^\pi(h_1o_1) - \mathbb{E}_{o_1\sim\bar{T}'(h_1)}\,V'^\pi(h_1o_1)]\end{aligned} \tag{5.60}$$

$$\begin{aligned}
&= \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{r_1 \sim \bar{R}(h_1)}\, r_1 - \mathbb{E}_{r_1 \sim \bar{R}'(h_1)}\, r_1] \\
&\quad + \gamma\, \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{o_1 \sim \bar{T}(h_1)}\, V^\pi(h_1 o_1) - \mathbb{E}_{o_1 \sim \bar{T}'(h_1)}\, V^\pi(h_1 o_1)] \\
&\quad + \gamma\, \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{o_1 \sim \bar{T}'(h_1)}\, V^\pi(h_1 o_1) - \mathbb{E}_{o_1 \sim \bar{T}'(h_1)}\, V'^\pi(h_1 o_1)]
\end{aligned} \tag{5.61}$$

Let $\pi\bar{T} : \mathcal{HO} \to \Delta(\mathcal{AO})$ represent the joint probability function over the next action and observation. We continue,

$$\begin{aligned}
&= \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{r_1 \sim \bar{R}(h_1)}\, r_1 - \mathbb{E}_{r_1 \sim \bar{R}'(h_1)}\, r_1] \\
&\quad + \gamma\, \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{o_1 \sim \bar{T}(h_1)}\, V^\pi(h_1 o_1) - \mathbb{E}_{o_1 \sim \bar{T}'(h_1)}\, V^\pi(h_1 o_1)] \\
&\quad + \gamma\, \mathbb{E}_{a_1 o_1 \sim \pi\bar{T}'(o_0)}[V^\pi(h_1 o_1) - V'^\pi(h_1 o_1)]
\end{aligned} \tag{5.62}$$

and expand the third term recursively,

$$\begin{aligned}
&= \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{r_1 \sim \bar{R}(h_1)}\, r_1 - \mathbb{E}_{r_1 \sim \bar{R}'(h_1)}\, r_1] \\
&\quad + \gamma\, \mathbb{E}_{a_1 o_1 a_2 \sim \pi\bar{T}'\pi(o_0)}[\mathbb{E}_{r_2 \sim \bar{R}(h_2)}\, r_2 - \mathbb{E}_{r_2 \sim \bar{R}'(h_2)}\, r_2] \\
&\quad + \gamma\, \mathbb{E}_{a_1 \sim \pi(o_0)}[\mathbb{E}_{o_1 \sim \bar{T}(h_1)}\, V^\pi(h_1 o_1) - \mathbb{E}_{o_1 \sim \bar{T}'(h_1)}\, V^\pi(h_1 o_1)] \\
&\quad + \gamma^2\, \mathbb{E}_{a_1 o_1 a_2 \sim \pi\bar{T}'\pi(o_0)}[\mathbb{E}_{o_2 \sim \bar{T}(h_2)}\, V^\pi(h_2 o_2) - \mathbb{E}_{o_2 \sim \bar{T}'(h_2)}\, V^\pi(h_2 o_2)] \\
&\quad + \gamma^2\, \mathbb{E}_{a_1 o_1 a_2 o_2 \sim \pi\bar{T}'\pi\bar{T}'(o_0)}[V^\pi(h_2 o_2) - V'^\pi(h_2 o_2)]
\end{aligned} \tag{5.63}$$

$$\begin{aligned}
&= \sum_{t=1}^{\infty} \gamma^{t-1}\, \mathbb{E}_{h_t \sim \mathbf{N}', \pi, o_0}[\mathbb{E}_{r_t \sim \bar{R}(h_t)}\, r_t - \mathbb{E}_{r_t \sim \bar{R}'(h_t)}\, r_t] \\
&\quad \sum_{t=1}^{\infty} \gamma^{t}\, \mathbb{E}_{h_t \sim \mathbf{N}', \pi, o_0}[\mathbb{E}_{o_t \sim \bar{T}(h_t)}\, V^\pi(h_t o_t) - \mathbb{E}_{o_t \sim \bar{T}'(h_t)}\, V^\pi(h_t o_t)]
\end{aligned} \tag{5.64}$$

Now, let us extend the usual notion of state occupancy measure to NMDPs. This would be a probability distribution $d^\pi \in \Delta(\mathcal{H})$, defined as

$$d^\pi(h) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t\, \mathbb{P}(h_t = h \mid \mathbf{N}, \pi) \tag{5.65}$$

Note that this object already takes the expectation with respect to the initial distribution. Then, we can resume the computation above, and take its absolute value as:

$$\begin{aligned}
|V^\pi_\mu - V'^\pi_{\mu'}| &\leq \Big|\sum_{t=1}^{\infty} \gamma^{t-1}\, \mathbb{E}_{h_t \sim \mathbf{N}', \pi}[\mathbb{E}_{r_t \sim \bar{R}(h_t)}\, r_t - \mathbb{E}_{r_t \sim \bar{R}'(h_t)}\, r_t]\Big| \\
&\quad + \Big|\sum_{t=1}^{\infty} \gamma^{t}\, \mathbb{E}_{h_t \sim \mathbf{N}', \pi}[\mathbb{E}_{o_t \sim \bar{T}(h_t)}\, V^\pi(h_t o_t) - \mathbb{E}_{o_t \sim \bar{T}'(h_t)}\, V^\pi(h_t o_t)]\Big|
\end{aligned} \tag{5.66}$$

$$\begin{aligned}
&= \frac{1}{1-\gamma} |\mathbb{E}_{h_t \sim d'^\pi}[\mathbb{E}_{r_t \sim \bar{R}(h_t)}\, r_t - \mathbb{E}_{r_t \sim \bar{R}'(h_t)}\, r_t]| \\
&\quad + \frac{\gamma}{1-\gamma} |\mathbb{E}_{h_t \sim d'^\pi}[\mathbb{E}_{o_t \sim \bar{T}(h_t)}\, V^\pi(h_t o_t) - \mathbb{E}_{o_t \sim \bar{T}'(h_t)}\, V^\pi(h_t o_t)]|
\end{aligned} \tag{5.67}$$

$$\begin{aligned}&\leq \frac{1}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^{\pi}}|\mathbb{E}_{r_t\sim\bar{R}(h_t)}\, r_t - \mathbb{E}_{r_t\sim\bar{R}'(h_t)}\, r_t| \\ &\quad + \frac{\gamma}{1-\gamma}\,\mathbb{E}_{h_t\sim d'^{\pi}}|\mathbb{E}_{o_t\sim\bar{T}(h_t)}\, V^{\pi}(h_t o_t) - \mathbb{E}_{o_t\sim\bar{T}'(h_t)}\, V^{\pi}(h_t o_t)|\end{aligned} \tag{5.68}$$

□

Proposition 5.10. *For any* $0 \leq \epsilon_1 \leq 1$ *and* $\epsilon > 0$*, there exists an RDP which is an* ϵ*-approximation of* $\mathbf{P}_1$*, the POMDP of fig. 5.6.*

Proof. Let us define an RDP $\mathbf{R}_{\mathbf{P}_1} = \langle \tilde{\mathcal{P}}\mathcal{O}, \mathcal{O}\mathcal{A}, \Omega, \tau, \theta, v_0\bot\rangle$. For a number $n \in \mathbb{N}_+$, to be specified, we define $\tilde{\mathcal{P}} \coloneqq \{p_2, v_0, \dots, v_n, w_0, \dots, w_n\}$, where $p_2 = \delta_{s_2}$, $v_i \coloneqq [(1-\epsilon_1)^i, 1-(1-\epsilon_1)^i, 0]$ and $w_i \coloneqq [(1-\epsilon_1)^i, 0, 1-(1-\epsilon_1)^i]$. The output function θ is a composition between null rewards and the distribution over observations, $\theta_{\mathsf{o}}(po) = \mathrm{Ber}(\top \mid p(s_2))$. As a side note, the term $p(s_2)$ can be interpreted as the approximate prior probability associated to s_2, the priors v_i are associated to sequences of consecutive a_0, and the priors w_i are associated to sequences of a_0 followed by one a_1. Finally, the transition function is defined:

$$\tau(v_i o, o'a') = \begin{cases} (v_{i+1}, o') & \text{if } i < n \text{ and } a' = a_0 \\ (v_n, o') & \text{if } i = n \text{ and } a' = a_0 \\ (w_i, o') & \text{if } a' = a_1 \end{cases} \tag{5.69}$$

$$\tau(w_i o, o'a') = \begin{cases} (p_2, o') & \text{if } o' = \top \\ (v_1, o') & \text{if } o' = \bot \text{ and } a' = a_0 \\ (v_0, o') & \text{if } o' = \bot \text{ and } a' = a_1 \end{cases} \tag{5.70}$$

$$\tau(p_2 o, o'a') = p_2 o' \tag{5.71}$$

It now remains to show that $\mathbf{R}_{\mathbf{P}_1}$ ϵ-approximates $\mathbf{P}_1$. By construction, over any history h_t of arbitrarily length that does not end in more than n consecutive repetitions of a_0, the current state $\bar{\tau}(v_0\bot, h_t)$ is equal to the prior belief $\mathbb{P}(s_t \mid h_t)$. On the other hand, the belief δ_{s_1}, which is only reachable in the limit of $h_i = (\bot\, a_0)^i$ for i that tends to infinity, is approximated by v_n. It can be shown that this approximation error is the maximum reachable distance with true beliefs, that is $\max_{h\in\mathcal{H}} \|\bar{\tau}(v_0\bot, h) - \mathbb{P}(s \mid h)\|_1 \leq \|\delta_{s_1} - v_n\|_1$. Computing this distance explicitly, we have $\|\delta_{s_1} - v_n\|_1 = 2(1-\epsilon_1)^n$. Also, since observations of the induced NMDPs satisfy

$$\|\theta_{\mathsf{o}}(\bar{\tau}(v_0\bot, h_t)) - \mathbb{P}(o_t \mid h_t, \mathbf{P}_1)\|_1 \leq \|\bar{\tau}(v_0\bot, h_t) - \mathbb{P}(s_t \mid h_t, \mathbf{P}_1)\|_1 \tag{5.72}$$

it suffices to ensure $2(1-\epsilon_1)^n \leq \epsilon$. This is true for $n \geq \log_{1-\epsilon_1}(\epsilon/2)$. □

Lemma 5.16. *(Even-Dar, Sham M. Kakade, et al. 2007) Given a POMDP* $\mathbf{P}$*, for any two beliefs* $b, b' \in \mathcal{B}$ *and action* $a \in \mathcal{A}$*, the expected relative entropy after a belief update satisfies,*

$$\mathbb{E}_{o\sim O(b)}[D_{KL}(U_p(b,o,a') \parallel U_p(b',o,a'))] \leq D_{KL}(b \parallel b') - D_{KL}(O(b) \parallel O(b')) \tag{5.73}$$

for all $a' \in \mathcal{A}$*.*

Proof. This is an application of the data processing inequality. It can be also seen as an instance of proposition 3.4 of Even-Dar, Sham M. Kakade, et al. (2007), for $\epsilon_U = \epsilon_O = \epsilon_T = 0$. □

Lemma 5.17. *Given a* ξ*-observable POMDP* $\mathbf{P}$ *and a* (η, ν)*-divergent cover of* $\mathcal{P}$*, for any* $p \in \mathcal{P}$ *and* $\tilde{p} := \tilde{f}(p)$*,*

$$D_{KL}(O(p) \parallel O(\tilde{p})) \geq \frac{\xi^2 \nu}{2} D_{KL}(p \parallel \tilde{p}) \tag{5.74}$$

Proof. From the Pinsker's inequality and the definition of ξ-observable POMDP,

$$D_{KL}(O(p) \parallel O(\tilde{p})) \geq \|O(p) - O(\tilde{p})\|_1^2 /2 \tag{5.75}$$

$$\geq \frac{\xi^2}{2} \|p - \tilde{p}\|_1^2 \tag{5.76}$$

From theorem 2 of Verdú (2014) (derived from Csiszár and Talata (2006)), we can lower bound the last term to obtain

$$D_{KL}(O(p) \parallel O(\tilde{p})) \geq \frac{\xi^2 \nu}{2} D_{KL}(p \parallel \tilde{p}) \tag{5.77}$$

□

Lemma 5.18. *Given a POMDP* $\mathbf{P}$ *and a* η*-divergent cover* $\tilde{\mathcal{P}}$ *of* $\mathcal{P}$*, for any* $p, \dot{p} \in \mathcal{P}$*,*

$$D_{KL}(p \parallel \tilde{f}(\dot{p})) - D_{KL}(p \parallel \dot{p}) \leq \eta \tag{5.78}$$

Proof. This statement first appeared in Boyen and Koller (1998). We provide a proof here. For any $p, \dot{p} \in \mathcal{B}$,

$$D_{KL}(p \parallel \tilde{f}(\dot{p})) - D_{KL}(p \parallel \dot{p}) = \tag{5.79}$$

$$= \sum_{s\in\mathcal{S}} p(s) \left(\log\left(\frac{p(s)}{\tilde{f}(s \mid p)} \right) - \log\left(\frac{p(s)}{\dot{p}(s)} \right) \right) \tag{5.80}$$

$$= \mathbb{E}_{s\sim p}\left[\log\left(\frac{\dot{p}(s)}{\tilde{f}(s \mid \dot{p})} \right) \right] \tag{5.81}$$

$$\leq \iota_{\dot{p}\|\tilde{f}(s|\dot{p})} \tag{5.82}$$

The result now follows from the definition of η-divergent cover. □

Theorem 5.11. *Given any ξ-observable POMDP $\mathbf{P}$, if $\tilde{\mathcal{P}}$ is a (η, ν)-divergent cover of the reachable prior space, the covering RDP of $\mathbf{P}$ into $\tilde{\mathcal{P}}$ is an ϵ-approximation of $\mathbf{P}$, for $\epsilon = \sqrt{\frac{4\eta}{\xi^2 \nu}}$, provided that $\nu < 2/\xi^2$.*

Proof. We use $p_t \in \mathcal{P}$ to represent the exact prior belief $p_t \coloneqq U_p^*(h_t)$. Let $\mathbf{R_P}$ be the approximating RDP and $\tilde{\mathcal{P}}$ the (η, ν)-divergent cover of $\mathcal{P}$. We write $\tilde{p}_t$ to denote the approximate prior belief, computed from $\mathbf{R_P}$, as $\tilde{p}_t \coloneqq \bar{\tau}(\tilde{p}_0, h_t)$. Also, let $\dot{p}_t$ be the approximate belief before the projection into $\tilde{\mathcal{P}}$, that is $\dot{p}_t \coloneqq U_p(\tilde{p}_{t-1}, o_{t-1}, a_t)$.

Now, regarding observations, to verify the theorem we need to show that,

$$\mathbb{E}_{h_t \sim \mathbf{P}|a_{1:t}} |\bar{T}_{\mathbf{R_P}}(o \mid h_t) - \bar{T}_{\mathbf{P}}(s \mid h_t)| \leq \epsilon \tag{5.83}$$

which, by definition of $\mathbf{P}$ and $\mathbf{R_P}$ is equivalent to ensure

$$\mathbb{E}_{o:t-1} \|O(p_t) - O(\tilde{p}_t)\|_\infty \leq \mathbb{E}_{o:t-1} \|O(p_t) - O(\tilde{p}_t)\|_1 \leq \epsilon \tag{5.84}$$

where here and in the following, expectations are implicitly computed with respect to $\mathbf{P}|a_{1:t}$. We start by observing that the error induced over the generated observations is limited by the error in prior belief:

$$\|O(p_t) - O(\tilde{p}_t)\|_1 \leq \|p_t - \tilde{p}_t\|_1 \tag{5.85}$$

which, under Pinsker's inequality satisfies

$$\|p_t - \tilde{p}_t\|_1 \leq \sqrt{2\, D_{KL}(p_t \parallel \tilde{p}_t)} \tag{5.86}$$

Thus, we now proceed to show that the relative entropy between real and approximated beliefs remains bounded at all $t \in \mathbb{N}$. This is proved via induction. Specifically, we show

$$\mathbb{E}_{o:t-1}[D_{KL}(p_t \parallel \tilde{p}_t)] \leq \sum_{i=0}^{t} \eta(1 - \xi^2\, \nu/2)^i \tag{5.87}$$

for any $a_{1:t} \in \mathcal{A}^t$, at all $t \in \mathbb{N}$.

For the base case, at $t = 0$, we have by definition $\tilde{p}_0 \coloneqq \tilde{f}(p_0)$. Therefore,

$$D_{KL}(p_0 \parallel \tilde{f}(p_0)) \leq \iota_{p_0\|\tilde{f}(p_0)} \leq \eta \tag{5.88}$$

where the second inequality follows by construction of η-divergent cover of $\mathcal{P}$. For the inductive step, assume (5.87) is true at some t. For any history h_t, from lemma 5.16, we know that, in expectation, divergence of beliefs reduces after a belief update:

$$\mathbb{E}_{o_t \sim O(b_t)}[D_{KL}(p_{t+1} \parallel \dot{p}_{t+1})] \leq D_{KL}(p_t \parallel \tilde{p}_t) - D_{KL}(O(p_t) \parallel O(\tilde{p}_t)) \tag{5.89}$$

for all $a_{t+1} \in \mathcal{A}$. Furthermore, from an application of lemma 5.17, we get

$$\mathbb{E}_{o_t \sim O(b_t)}[D_{KL}(p_{t+1} \parallel \dot{p}_{t+1})] \leq D_{KL}(p_t \parallel \tilde{p}_t) - \frac{\xi^2 \nu}{2} D_{KL}(p_t \parallel \tilde{p}_t) \tag{5.90}$$

Taking expectations over $o_{:t-1} \sim \mathbf{P} \mid a_{:t}$, in both sides, yields

$$\mathbb{E}_{o_{:t}}[D_{KL}(p_{t+1} \parallel \dot{p}_{t+1})] \leq (1 - \xi^2 \nu/2)\, \mathbb{E}_{o_{:t-1}}[D_{KL}(p_t \parallel \tilde{p}_t)] \tag{5.91}$$

for all $a_{t+1} \in \mathcal{A}$. Now, with an application of lemma 5.18 over p_{t+1} and $\dot{p}_{t+1}$,

$$\mathbb{E}_{o_{:t}}[D_{KL}(p_{t+1} \parallel \tilde{f}(\dot{p}_{t+1}))] - \eta \leq (1 - \xi^2 \nu/2)\, \mathbb{E}_{o_{:t-1}}[D_{KL}(p_t \parallel \tilde{p}_t)] \tag{5.92}$$

$$\mathbb{E}_{o_{:t}}[D_{KL}(p_{t+1} \parallel \tilde{p}_{t+1})] \leq (1 - \xi^2 \nu/2)\, \mathbb{E}_{o_{:t-1}}[D_{KL}(p_t \parallel \tilde{p}_t)] + \eta \tag{5.93}$$

Finally, from an application of the inductive hypothesis,

$$\mathbb{E}_{o_{:t}}[D_{KL}(p_{t+1} \parallel \tilde{p}_{t+1})] \leq \sum_{i=0}^{t+1} \eta (1 - \xi^2 \nu/2)^i \tag{5.94}$$

This proves eq. (5.87). For $\nu < 2/\xi^2$, each term is positive and we also know

$$\mathbb{E}_{o_{:t}}[D_{KL}(p_t \parallel \tilde{p}_t)] \leq \sum_{i=0}^{\infty} \eta (1 - \xi^2 \nu/2)^i = \frac{2\eta}{\xi^2 \nu} \tag{5.95}$$

To conclude, because of eq. (5.86), to verify eq. (5.21), it suffices that

$$\sqrt{\frac{4\eta}{\xi^2 \nu}} \leq \epsilon \tag{5.96}$$

The same derivation is also sufficient for rewards. □

Theorem 5.13. *For any $\epsilon > 0$ and POMDP $\mathbf{P}$ whose mixing rate is $\rho > 0$, there exists a covering RDP that is an ϵ-approximation of $\mathbf{P}$.*

Proof. We follow the same definitions and reasoning as the proof for theorem 5.11 up to eq. (5.86). In the inductive proof, instead, we show

$$\mathbb{E}_{o_{:t-1} \sim \mathbf{P} \mid a_{1:t}}[D_{KL}(p_t \parallel \tilde{p}_t)] \leq \sum_{i=0}^{t} \eta\, (1 - \rho)^t \tag{5.97}$$

for any $a_{1:t} \in \mathcal{A}^t$, at all $t \in \mathbb{N}$. The base case also applies without modifications. In the inductive step, assume at some t, eq. (5.97) holds. For any history h_t, divergence in beliefs does not increases in expectation after a conditioning step:

$$\mathbb{E}_{o_t \sim O(b_t)}\left[D_{KL}(U_O p_t, o_t \parallel U_O \tilde{p}_t, o_t)\right] \leq D_{KL}(p_t \parallel \tilde{p}_t) \tag{5.98}$$

This can be seen as an instance of lemma 3.9 in (Even-Dar, Sham M. Kakade, et al. 2007) with $\epsilon_O = 0$. Furthermore, from lemma 5.12, divergence in beliefs decreases after a prediction step. Combining the two results:

$$D_{KL}(p_{t+1} \parallel \dot{p}_{t+1}) \leq (1-\rho)\, D_{KL}(p_t \parallel \tilde{p}_t) \tag{5.99}$$

Taking expectations and with an application of lemma 5.18 over p_{t+1} and $\dot{p}_{t+1}$,

$$\mathbb{E}_{o:t}\left[D_{KL}(p_{t+1} \parallel \tilde{p}_{t+1})\right] \leq (1-\rho)\, \mathbb{E}_{o:t-1}\left[D_{KL}(p_t \parallel \tilde{p}_t)\right] + \eta \tag{5.100}$$

which proves the inductive step. As a consequence,

$$\mathbb{E}_{o:t}\left[D_{KL}(p_t \parallel \tilde{p}_t)\right] \leq \sum_{i=0}^{\infty} \eta(1-\rho)^i = \frac{\eta}{\rho} \tag{5.101}$$

Because of eqs. (5.85) and (5.86), to ensure

$$\|\mathbb{P}(o_t \mid p_t, \mathbf{P}) - \mathbb{P}(o_t \mid \tilde{p}_t, \mathbf{R_P}))\|_1 \leq \epsilon \tag{5.102}$$

it suffices that

$$\sqrt{\frac{2\,\eta}{\rho}} \leq \varepsilon \tag{5.103}$$

The same is also sufficient for rewards. □

Chapter 6

Offline Reinforcement Learning in RDPs

> The content of this chapter is based on the work: Roberto Cipollone, Anders Jonsson, Alessandro Ronca, and Mohammad Sadegh Talebi (2024). "Provably Efficient Offline Reinforcement Learning in Regular Decision Processes". In: *Thirty-Seventh Conference on Neural Information Processing Systems, NeurIPS 2024*.

The previous chapter had been an in-depth study of how Regular Decision Processes work, what are their properties, and their relevance for environments with complex history dependencies or partial observations. In this chapter, we provide a complete RL algorithm for RDPs with sample efficiency guarantees.

6.1 Introduction

When learning in RDPs, the RL algorithm has no access to the hidden states of the RDP, but it may only observe the executed actions and the generated rewards and observations. The objective of an RL algorithm is to learn a near-optimal policy for the unknown RDP, by only receiving the traces produced as a result of the interaction. In this work, we specifically consider the Offline Reinforcement Learning setting. In offline RL, the agent may not interact with the environment directly. Rather, it is only given access to a dataset of environment interactions that have been previously collected using some behaviour policy. In MDPs, the dataset is usually composed of tuples of individual transitions. In RDPs, on the other hand, the entire interaction sequence is relevant. For this reason, the dataset is composed of some number of complete episode traces. In this work, we consider the finite-horizon setting. In summary, the purpose of the learning algorithm is to compute a near-optimal policy for an unknown RDP **R** in the finite horizon setting, by receiving a finite number of episodes that have been sampled with an unknown behaviour policy in **R**. Despite the extensive and rich literature on MDPs, comparatively little work exists on offline RL in non-Markovian decision processes. The scarcity of results may likely be attributed to the difficult nature of the problem, rather than the lack of interest.

6.1.1 Contributions

In this work, we establish a first, to the best of our knowledge, sample complexity lower bound for offline RL of RDPs (section 6.6). We introduce an offline RL algorithm, called `RegORL`, that learns ε-optimal policies for any RDP, in the episodic setting. At the core of `RegORL`, there is a component called AdaCT–H, which is a variant of AdaCT (Balle, J. Castro, et al. 2013), carefully tailored to episodic RDPs. AdaCT–H learns a minimal automaton that underlies the unknown RDP, without any prior knowledge. The output automaton is then used to construct a Markovian transformation of the input data. Thus, for solving the original RDP, the resulting dataset can be passed to any off-the-shelf algorithm for offline RL in episodic MDPs.

We present a sample-complexity bound for AdaCT–H to return a minimal automaton consistent with the input data with high probability. This bound substantially improves the existing bound for the original AdaCT, and can be of independent interest. In view of the modular design of `RegORL`, the total sample complexity is controlled by twice that of AdaCT–H (theorem 6.6) and that for the incorporated off-the-shelf algorithm.

We also present another variant of AdaCT–H, called AdaCT–H–A. In contrast to AdaCT–H, which learns a complete RDP, AdaCT–H–A only reconstructs the subset of states that are likely under the behaviour policy, in relation to an input accuracy parameter. As such, AdaCT–H–A is more aligned with the common practice of RL than AdaCT–H.

Furthermore, we provide a first lower bound for offline RL in RDPs that involves the relevant parameters for the problem, such as the RDP single-policy concentrability, which extends an analogous notion for MDPs from the literature. Finally, if contrasted to both online learning in RDPs and automata learning, our results suggest possible improvements in sample complexity results for both areas.

6.1.2 Related Work

Offline RL in MDPs There is a rich and growing literature on offline RL, and provably sample efficient algorithms have been proposed for various settings of MDPs (Uehara and Sun 2022; Yin and Y.-X. Wang 2021; J. Chen and N. Jiang 2019; Xie,

N. Jiang, et al. 2021; Rashidinejad, Zhu, et al. 2021; G. Li, L. Shi, et al. 2022; Zhan, Huang, et al. 2022; Y. Jin, Z. Yang, et al. 2021; Ren, J. Li, et al. 2021; Uehara, X. Zhang, et al. 2022). In the case of episodic MDPs, it is established that the optimal sample size in offline RL depends on the size of state-space, the episode length, as well as some notion of concentrability, reflecting the distribution mismatch between the behaviour and optimal policies. A closely related problem is off-policy learning; see, for example, P. S. Thomas and Brunskill (2016), Maei, Szepesvári, et al. (2010), and Kallus and Uehara (2020) and the recent survey Uehara, C. Shi, et al. (2022). For offline RL in MDPs, the papers cited above report learning algorithms with theoretical guarantees on their sample efficiency. The majority of these algorithms are designed based on the *pessimism principle.* While most literature focuses on tabular MDPs, the case of linear function approximation is discussed in some papers, such as, Uehara, X. Zhang, et al. (2022).

Online RL in RDPs RDPs have been introduced in Brafman and De Giacomo (2019) as a formalism based on temporal logic. They admit an equivalent formulation in terms of automata, which is favoured in the context of RL. Several algorithms for *online* RL in RDPs exist (Abadi and Brafman 2020; Ronca and De Giacomo 2021; Ronca, Licks, et al. 2022), but complexity bounds are only given in Ronca and De Giacomo (2021) for the infinite-horizon discounted setting. This last work shows the correspondence between RDPs and Probabilistic Deterministic Finite Automata (PDFAs), and it introduces the idea of using PDFA-learning techniques to learn RDPs. Their sample complexity bounds are not immediately comparable to ours, due to the different setting. Importantly, this algorithm uses the uniform policy for learning. So, the algorithm might be adapted to our setting only under the assumption that the behaviour policy is uniform. Even in this case, our bounds show an improved dependency on several key quantities. Furthermore, we provide a sample complexity lower bound, whereas their results are limited to showing that a dependency on the quantities occurring in their upper bounds is necessary.

The first RL algorithm for RDPs appears in (Abadi and Brafman 2020) for the online discounted setting. It is automaton-based, and in particular, it learns the RDP in the form of a Mealy machine. The algorithm is shown in (Ronca and De Giacomo 2021) to incur in an exponential dependency on the length of the relevant histories. An algorithm that integrates a more effective exploration strategy is given in (Ronca, Licks, et al. 2022). This work also introduces the idea of seeing the transition function of a PDFA as a Markov abstraction of the histories to be passed to an RL algorithm for MDPs, so as to employ it in a modular manner.

The algorithms in Icarte, Waldie, et al. (2019), Hutter (2009), Veness, K. S. Ng, et al. (2011), and Mahmud (2010) apply to RDPs even though they have not been developed specifically for RDPs. In Icarte, Waldie, et al. (2019) the authors present an RL algorithm for the subclass of POMDPs that have a finite set of reachable belief states. As we saw in the previous chapter, these means that they can be modelled as RDPs. Their algorithm is based on automata learning, but it does not come with an analysis of its performance guarantees.

The RL techniques presented in (Hutter 2009; Veness, K. S. Ng, et al. 2011) for *feature MDPs* are in fact applicable to episodic RDPs. The techniques are based on suffix trees, rather than automata. However, there are cases when the size of the smallest suffix tree is exponential in the horizon, while an automaton of linear size exists. Thus, their techniques cannot yield optimal bounds for RDPs. Mahmud (2010) introduces an RL algorithm for *Deterministic Markov Decision Models* (MDDs). Such MDDs are also automaton-based, and their RL algorithm applies to RDPs as well. However, the algorithm is provided without guarantees.

Non-Markov Rewards and Reward Machines MDPs with non-Markov rewards are a special case of NMDPs, where only rewards are non-Markovian. Namely, observations satisfy the Markov property, while rewards may depend on the entire history. The specific kind of non-Markovian rewards considered in the literature amount to the subclass of RDPs where the automaton state is only needed to predict the next reward—while the next observation can be predicted from the last observation. The relation between RDPs and automata-based formalisms for reward specifications has been discussed in the related work section of chapter 5.

More relevant to this work are Gaon and Brafman (2019) and Xu, Gavran, et al. (2020). These are RL algorithms with unknown reward machines, meaning, unknown temporal specifications, however they are also presented with no performance guarantees.

State Representations State representations are maps from histories to a finite state space. The map defined by the transition function of an RDP is a state representation. Works, such as Maillard, Munos, et al. (2011), Nguyen, Maillard, et al. (2013), Maillard, Nguyen, et al. (2013), and Ortner, Pirotta, et al. (2019), are studies on state representations, that focus on regret bounds for RL given a candidate set of state representations. While in our case the state representations are concretely defined by the class of finite-state automata, in their case they are arbitrary maps. This is a challenging setting, which does not allow for taking advantage of the properties of specific classes of state representations. The regret bounds in Maillard, Munos, et al. (2011), Maillard, Nguyen, et al. (2013), and Ortner, Pirotta, et al. (2019) are for finite sets of state representations, and they all show a linear dependency on the cardinality of the given set of state representations.

In our case, the candidate state representations corresponds to the set of automata with at most $Q = 2(AO)^H$ states and AO input letters. Such a set contains at least Q^{QAO} automata—the number of distinct transition functions. Thus, if we could instantiate their bounds in our setting, they would have an exponential dependency on the number Q of RDP states, and hence a doubly-exponential dependency on the horizon H. We avoid this dependency, obtaining polynomial bounds in the mentioned quantities.

Nguyen, Maillard, et al. (2013) consider the case of a countably infinite set of state representations, and present an algorithm whose regret bound does not show a dependency such as the one discussed above. Instead, they show a dependency on a quantity K_0, which admits several interpretations, including one based on the descriptional complexity of the candidate state representations. Thus, there may be a way to relate K_0 to the quantities we use in our bounds. However, the formal relationship between the two, if any, renders highly non-trivial, which prevents one to use their ideas in the case of RDPs. We believe establishing a formal relationship between their model and RDPs is an interesting, yet challenging, topic for future work. Furthermore, it should be stressed that even if the relationship was clear and one could borrow ideas from this paper, the resulting sample complexity bound would have to grow as $1/\varepsilon^3$ in view of their regret bound scaling as $T^{2/3}$. In contrast, our bounds achieve an optimal dependency of $1/\varepsilon^2$ on ε.

PSRs Predictive State Representations (PSRs) (Littman, Sutton, et al. 2001; S. P. Singh, Littman, et al. 2003; James and S. Singh 2004; Bowling, McCracken, et al. 2006; Kulesza, N. Jiang, et al. 2015) are general descriptions of dynamical systems that capture POMDPs and hence RDPs. There exist polynomial PAC bounds for online RL in PSRs (Zhan, Uehara, et al. 2023). Nonetheless, these bounds are looser than the one we show here, since they must necessarily consider a wider class of models. Moreover, although a minimum core set for PSRs is similar to a minimal RDP, the bounds feature a number of quantities that are specific to PSRs (such as, the regularity parameter) and do not immediately apply to RDPs. Since POMDPs are more restrictive than PSRs, in the specific subclass of POMDPs, we remind Monte-Carlo algorithms (Silver and Veness 2010).

Feature MDPs and General RL Hutter (2009) introduces *feature MDPs*, where histories are mapped to states by a feature map. This relates to our work since the map provided by the transition function of an RDP is a feature map. The concrete feature maps they consider are based on U-Trees (McCallum 1996). The idea is also revisited in (Veness, K. S. Ng, et al. 2011) with Prediction Suffix Trees (PSTs) (Rissanen 1983; Ron, Singer, et al. 1996). Both U-Trees and PSTs are suffix trees.

There are cases when their size is exponential in the horizon, while an automaton of linear size exists. For instance, in the case of a parity condition over the history. To see this, note that a suffix x of a bit string bx does not suffice to establish parity of bx. In fact, the parity of $0x$ is different from the parity of $1x$. Thus, a suffix tree for parity must encode all suffixes, and hence it will have a number of leaves that is exponential in the maximum length of a relevant string—the horizon H in the case of episodic RL.

Lattimore, Hutter, et al. (2013) consider General RL as the problem of RL when we are given a set of candidate NMDPs, rather than assuming the decision process to belong to a fixed class. Similarly to the works on state representations, it does not commit to specific classes of NMDPs, and their bounds have a linear dependency on the number of candidate models. As remarked above, in our setting, it amounts to an exponential dependency on the number of states of the candidate RDPs, and hence a doubly-exponential dependency on the horizon; we avoid such exponential dependencies.

Learning PDFA Our algorithms for learning an RDP borrow and improve over techniques for learning Probabilistic-Deterministic Finite Automata (PDFA). The first PAC learning algorithm for acyclic PDFA has been presented in Ron, Singer, et al. (1998), then followed by extensions and variants that can handle PDFA with cycles (Clark and Thollard 2004; Palmer and Goldberg 2007; Balle, J. Castro, et al. 2013; Balle Pigem 2013; Balle, J. Castro, et al. 2014). All bounds feature some variant of a *distinguishability parameter*, which we adopt in our bounds, properly adapting it to the offline RL setting. Our algorithm builds upon the state-of-the-art algorithm AdaCT (Balle, J. Castro, et al. 2013), and we derive bounds that are a substantial improvement over the ones that can be obtained from a straightforward application of any existing PDFA-learning algorithm to the offline RL setting.

RL with Neural Networks Among the approaches without formal guarantees, arguably the most common solution technique is to use Deep RL algorithms, coupled with Neural Networks architectures that are able to process sequences, instead of individual observations. In fact, applying Deep RL with feed-forward architectures to partially-observable environments may lead to arbitrarily suboptimal results, even for very promising learning algorithms (see for example the worst performing environments in Mnih, Kavukcuoglu, et al. (2015)). For processing sequences, Recurrent Neural Networks (RNNs) are a very natural choice. The integration of RL with RNN has been advised as a promising solution since Lin and Mitchell (1993) and Hauskrecht (2000). The idea was later expanded to LSTMs (Bakker 2001; Hausknecht and Stone 2015; Heess, Hunt, et al. 2015) and policy gradients (Wierstra, Förster, et al. 2007). Thanks to these successes, since Mnih, Badia, et al. (2016), each new RL algorithm is natively compatible with recurrent architectures. This has led to impressive results in video games with first-person view (Oh, Chockalingam, et al. 2016; Lample and Chaplot 2017; Mirowski, Pascanu, et al. 2017). Model-based learning algorithms have also been developed, with similar techniques (X. Li, L. Li, et al. 2015; Ha and Schmidhuber 2018). We also remind a different approach based on variational RL (Igl, Zintgraf, et al. 2018), and an in-depth study about the impact of the experience replay buffer in non-Markovian RL (Kapturowski, Ostrovski, et al. 2019). Lastly, we recall that any NN architecture for sequences may be suitable for the purpose, even attention mechanisms (Vaswani, Shazeer, et al. 2017). Although very related to the problem studied here, these works mostly provide no correctness guarantees, and they may fail to converge for the hardest temporal dependencies.

6.2 Preliminaries

This chapter adopts most of the common notation that has been set in chapter 2, with some differences that we will highlight next. Since we work under the finite-horizon setting, all the appropriate definitions apply, including the finiteness of histories, traces, episodes, and undiscounted value functions. Each episode will be composed by $H \in \mathbb{N}_+$ transition steps, after which, the interaction stops, and it may only be resumed from the initial distribution.

The only significant difference with the shared notation used in the rest of the these regards how histories and RDPs are represented. In this chapter, an RDP will be defined as a Moore machine, with input symbols in $\mathcal{AO}$. This is different from the definition that we have in section 2.2 and in chapter 5, where the input alphabet was $\mathcal{OA}$. Because of this choice, the most recent action has not been consumed after a transition, yet. Therefore, similarly to what would happen in a multi-armed bandit, each RDP state outputs a conditional distribution that associates each action to stochastic observations and rewards. As we discussed in the previous chapter, all these definitions are largely equivalent, since they still satisfy the main property of RDPs, namely, that the non-Markovian functions over histories $\bar{T}$ and $\bar{R}$ are regular. Therefore, every result be derived here and in other chapters continue to hold.

Consistently with this choice of the RDP inputs, histories will be defined analogously, as a concatenation of symbols in $\mathcal{AO}$. Unlike the rest of this thesis, then, they will not terminate with the last action, but the last observation, instead. This will allow writing h, instead of ho, in many locations, including the input arguments of policies, of value functions, and of RDP transition functions, extended over sequences. This change in how histories and RDPs are represented is motivated by specific needs.

In chapter 5, RDPs have been solely used as representations of the environment dynamics, and compactness has been of key importance. In this chapter, on the other hand, we are developing a learning algorithm. Therefore, RDP states should represent the agent's decision points, where the environment has generated and observation, but the agent has not selected an action yet.

Summarizing, in this chapter, histories and traces are defined as $\mathcal{H}_t \coloneqq (\mathcal{A}\mathcal{O})^t$ and $\mathcal{T}_t \coloneqq (\mathcal{A}\mathcal{R}\mathcal{O})^t$. The set of all histories and traces is $\mathcal{H} \coloneqq \cup_{i=0,\dots,H}\mathcal{H}_t$ and $\mathcal{T} \coloneqq \cup_{i=0,\dots,H+1}\mathcal{T}_t$ An episode $e_{0:H} \in \mathcal{T}_{H+1}$ is a complete trace of H transitions, as

$$e_H = a_0 r_0 o_0 a_1 r_1 o_1 \dots o_H \in \mathcal{T}_{H+1} \tag{6.1}$$

In general $e_{i:j} \in \mathcal{T}_{i-j+1}$ denotes a trace from time i to time j, included. The irrelevant variables $a_0 r_0$ are only included for simplifying the notation. They are always set to $a_0 = a_\circ$ and $r_0 = 0$. Histories are defined analogously, by omitting all rewards.

Generic policies are functions in $\Pi \coloneqq \mathcal{H} \to \Delta(\mathcal{A})$. With the due changes, the definitions of Markovian and stationary policies from page 13 still apply. The value function of a policy π in any decision process is written

$$V_t^\pi(h_t) \coloneqq \mathbb{E}[g_{t+1} \mid \mathbf{D}, \pi, h_t] \tag{6.2}$$

and $V_H^\pi \equiv 0$. Without referring to any history, the value of a policy is $V_\mu^\pi \coloneqq \mathbb{E}_{o_0 \sim \mu}[V_0^\pi(a_\circ o_0)]$, with respect to the initial observation distribution μ. Optimality and near-optimality is defined as usual.

Following the decisions above, we formalize an episodic Regular Decision Process (RDP) as a finite transducer (Moore machine) $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$, where $\mathcal{Q}$ is a finite set of states, $\Sigma \coloneqq \mathcal{A}\,\mathcal{O}$ is a finite input alphabet composed of actions and observations, Ω is a finite output alphabet, $\tau : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is a transition function, $\theta : \mathcal{Q} \to \Omega$ is an output function, and $q_0 \in \mathcal{Q}$ is a fixed initial state. The output space $\Omega \coloneqq \Omega_\mathsf{o} \times \Omega_\mathsf{r}$ consists of a finite set of functions that compute the conditional probabilities of observations and rewards, meaning $\Omega_\mathsf{o} \subset \mathcal{A} \to \Delta(\mathcal{O})$ and $\Omega_\mathsf{r} \subset \mathcal{A} \to \Delta(\mathcal{R})$. For simplicity, we use two output functions, $\theta_\mathsf{o} : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{O})$ and $\theta_\mathsf{r} : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{R})$, to denote the individual conditional probabilities. Also, let τ^{-1} denote the inverse of τ. In other words, $\tau^{-1}(q) \subseteq \mathcal{Q} \times \mathcal{A}\,\mathcal{O}$ is the subset of state-symbol pairs that map to $q \in \mathcal{Q}$. An RDP $\mathbf{R}$ implicitly represents a function $\bar{\tau} : \mathcal{H} \to \mathcal{Q}$ from histories in $\mathcal{H}$ to states in $\mathcal{Q}$, recursively defined as $\bar{\tau}(h_0) \coloneqq \tau(q_0, a_0 o_0)$ and $\bar{\tau}(h_t) \coloneqq \tau(\bar{\tau}(h_{t-1}), a_t o_t)$. The dynamics and of $\mathbf{R}$ are defined as $\bar{T}(o \mid ha) = \theta_\mathsf{o}(o \mid \bar{\tau}(h), a)$ and $\bar{R}(r \mid ha) = \theta_\mathsf{r}(o \mid \bar{\tau}(h), a)$, $\forall h \in \mathcal{H}, \forall aro \in \mathcal{A}\mathcal{R}\mathcal{O}$. In this context, an input symbol is an element of $\mathcal{A}\mathcal{O}$. We use A, R, O, Q to denote the cardinality of $\mathcal{A}, \mathcal{R}, \mathcal{O}, \mathcal{Q}$, respectively.

In the RL literature for episodic settings, the environment is often regarded to be non-stationary. To capture this time dependency in RDPs, we define *episodic* RDPs to be acyclic. This means that the states can be partitioned as $\mathcal{Q} = \mathcal{Q}_0 \cup \cdots \cup \mathcal{Q}_{H+1}$, where each $\mathcal{Q}_t$ is the set of states generated by histories in $\mathcal{H}_t$. An RDP is minimal if its Moore machine is minimal. Since there is nothing to predict at time $H+1$, a minimal RDP contains a single state q_{H+1} in $\mathcal{Q}_{H+1}$. To ensure that an acyclic RDP $\mathbf{R}$ is minimal, we introduce a designated termination observation $o_\perp$ in $\mathcal{O}$ and define $\tau(q_{H+1}, ao) = q_{H+1}$ and $\theta_{\mathsf{o}}(q_{H+1}, a) = \delta_{o_\perp}$ for any $ao \in \mathcal{AO}$. Hence, q_{H+1} is absorbing, and the states in $\mathcal{Q}$ must implicitly count how many steps are left until we observe $o_\perp$. This ensured the partitioned structured. Without $o_\perp$, a Moore machine could potentially represent all episodes using fewer than $H+2$ states.

As usual, since the conditional probabilities of observations and rewards are fully determined by the current state-action pair (q, a), an RDP $\mathbf{R}$ adheres to the Markov property over its states, but not over the observations. Given a state $q_t \in \mathcal{Q}$ and an action $a_t \in \mathcal{A}$, the probability of the next transition is

$$\mathbb{P}(r_t, o_t, q_{t+1} \mid q_t, a_t, \mathbf{R}) = \theta_{\mathsf{r}}(r_t \mid q_t, a_t)\, \theta_{\mathsf{o}}(o_t \mid q_t, a_t)\, \mathbb{I}(q_{t+1} = \tau(q_t, a_t o_t))$$

Evidently, in the special case where an RDP is Markovian in both observations and rewards, it reduces to an episodic MDP. More precisely, any episodic MDP with actions $\mathcal{A}$, states $\mathcal{O}$ and horizon H can be represented by some episodic RDP with states $\mathcal{Q} \subseteq \mathcal{O} \times [H+2]$ and inputs $\mathcal{AO}$.

As already we know, an important class of policies for RDPs are the regular policies. We summarize the main results here using this slightly modified RDP definition. Given an RDP $\mathbf{R}$, a policy $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ is called *regular* if $\pi(h_1) = \pi(h_2)$, whenever $\bar{\tau}(h_1) = \bar{\tau}(h_2)$, for all $h_1, h_2 \in \mathcal{H}$. Let $\Pi_{\mathbf{R}}$ denote the set of regular policies for $\mathbf{R}$. Regular policies exhibit powerful properties. First, under a regular policy, suffixes have the same probability of being generated for histories that map to the same RDP state. Second, there exists at least one optimal policy that is regular, deterministic, and it can be written as $\mathcal{Q} \to \mathcal{A}$. The following statements appear in Brafman and De Giacomo (2019), in analogous forms. We report the statements here and provide the proofs for completeness.

Proposition 6.1. *Consider an RDP $\mathbf{R}$, a regular policy $\pi \in \Pi_{\mathbf{R}}$ and two histories h_1 and h_2 in $\mathcal{H}_t$, $t \in [H]$, such that $\bar{\tau}(h_1) = \bar{\tau}(h_2)$. For each suffix $e_{t+1:H} \in \mathcal{T}_{H-t}$, the probability of generating $e_{t+1:H}$ is the same for h_1 and h_2, i.e. $\mathbb{P}(e_{t+1:H} \mid h_1, \pi, \mathbf{R}) = \mathbb{P}(e_{t+1:H} \mid h_2, \pi, \mathbf{R})$.*

Proof. See page 148. □

Proposition 6.2. *Each RDP $\mathbf{R}$ has at least one optimal policy $\pi^* \in \Pi_{\mathbf{R}}$.*

Proof. See page 149. □

Due to proposition 6.2, when solving an RDP $\mathbf{R}$, we can restrict our search to the set of regular policies $\Pi_{\mathbf{R}}$. A regular policy can be compactly defined as $\pi : \mathcal{Q} \to \Delta(\mathcal{A})$, with value function expressed as $V_t^\pi : \mathcal{Q} \to \mathbb{R}$, for $t \in [H+1]$.

Next, we define occupancy measures for RDPs. Given a regular policy $\pi : \mathcal{Q} \to \Delta(\mathcal{A})$ and $t \in [H+1]$, let $d_t^\pi \in \Delta(\mathcal{Q}_t \times \mathcal{A}\,\mathcal{O})$ be the induced probability distribution over the states in $\mathcal{Q}_t$ and input symbols in $\mathcal{A}\,\mathcal{O}$, recursively defined as $d_0^\pi(q_0, a_0 o_0) \coloneqq \theta_{\mathsf{o}}(o_0 \mid q_0, a_0)$ and

$$d_t^\pi(q_t, a_t o_t) \coloneqq \sum_{(q,ao) \in \tau^{-1}(q_t)} d_{t-1}^\pi(q, ao)\, \pi(a_t \mid q_t)\, \theta_{\mathsf{o}}(o_t \mid q_t, a_t)$$

We also overload the notation by writing $d_t^\pi(q_t, a_t) = \sum_{o \in \mathcal{O}} d^\pi(q_t, a_t o)$. Of particular interest is the occupancy distribution $d_t^* \coloneqq d_t^{\pi^*}$, associated with an optimal policy π^*.

6.3 Offline RL in RDPs

We are now ready to formalize the offline RL problem in episodic RDPs. Assume that we have access to a batch dataset $\mathcal{D}$, collected by interacting with an unknown (but fixed) episodic RDP $\mathbf{R}$, using a regular *behaviour* policy π^{b}. We assume that $\mathcal{D}$ comprises N episodes, where the k-th episode is of the form $e_{0:H}^k = a_0^k r_0^k o_0^k \cdots a_H^k r_H^k o_H^k$, where $q_0^k = q_0$ and where, for each $t \in [H]$,

$$a_t^k \sim \pi^{\mathsf{b}}(q_t^k), \quad r_t^k \sim \theta_{\mathsf{r}}(q_t^k, a_t^k), \quad o_t^k \sim \theta_{\mathsf{o}}(q_t^k, a_t^k), \quad q_{t+1}^k = \tau(q_t^k, a_t^k o_t^k) \qquad (6.3)$$

We remind that $\pi^{\mathsf{b}}, \theta_{\mathsf{o}}, \theta_{\mathsf{r}}, \tau$ are all unknown to the learner. The goal is to compute a near-optimal policy $\widehat{\pi}$ using the dataset $\mathcal{D}$, without further exploration. More precisely, for a pre-specified accuracy $\varepsilon \in (0, H]$, we aim to find an ε-optimal policy $\widehat{\pi}$, using the smallest dataset $\mathcal{D}$ possible.

By virtue of proposition 6.2, one may expect that it is sufficient to search for regular ε-optimal policies, which is indeed the case. In order to learn an ε-optimal policy from $\mathcal{D}$, some assumption is necessary regarding the policy π^{b} that was used to collect the episodes. Let $d_t^{\mathsf{b}} \coloneqq d_t^{\pi^{\mathsf{b}}}$ be the occupancy distribution of π^{b}. The following assumption requires that the behaviour policy assigns a positive probability to all actions, which ensures that π^{b} explores the entire minimal RDP.

Assumption 6.1. $\min_{t \in [H+1], q \in \mathcal{Q}_t, a \in \mathcal{A}} d_t^{\mathsf{b}}(q, a) > 0$

This assumption is only needed by theorem 6.6, which reconstructs the full unknown RDP. Theorem 6.8, instead, relies on a weaker assumption that can be expressed with the coefficient that will be introduced in definition 6.1.

The second assumption we require concerns the richness of π^{b} and its capability to allow us to distinguish the various RDP states. This is perfectly captured by notions of *distiguishability* arising in automata theory, such as in Balle Pigem (2013). We apply these concepts in our context, where such discrete distributions are generated from an RDP and a policy. Consider a minimal RDP $\mathbf{R}$, with states $\mathcal{Q} = \cup_{t \in [H+2]} \mathcal{Q}_t$. Given some policy π, at each time step $t \in [H+1]$, every RDP state $q \in \mathcal{Q}_t$ defines a unique probability distribution over the episode suffixes $\mathcal{T}_{H-t+1} = (\mathcal{ARO})^{H-t+1}$. Then, the states in each $\mathcal{Q}_t$ can be compared through the probability distributions they induce over $\mathcal{T}_{H-t+1}$. Consider any $L = \{L_\ell\}_{\ell=1}^{H+1}$, where each L_ℓ is a metric over $\Delta(\mathcal{T}_\ell)$. We define the L*-distinguishability* of $\mathbf{R}$ and π as the maximum μ_0 such that, for any $t \in [H+1]$ and any two distinct $q, q' \in \mathcal{Q}_t$, the probability distributions over suffix traces $e_{t:H} \in \mathcal{T}_\ell$ from the two states satisfy

$$L_{H-t+1}(\mathbb{P}(e_{t:H} \mid q_t = q, \pi), \mathbb{P}(e_{t:H} \mid q_t = q', \pi)) \geq \mu_0$$

We will often omit the remaining length of the episode $\ell = H - t + 1$ from L_ℓ and simply write L. We consider the L_∞^{p}-distinguishability, constructed by instantiating the definition above with the metric $L_\infty^{\mathsf{p}}(p_1, p_2) = \max_{u \in [\ell+1], e \in \mathcal{T}_u} |p_1(e\,*) - p_2(e\,*)|$, where $p_i(e\,*)$ represents the probability of the trace prefix $e \in \mathcal{T}_u$, followed by any trace $e' \in \mathcal{T}_{\ell-u}$. The L_1^{p}-distinguishability is defined analogously using $L_1^{\mathsf{p}}(p_1, p_2) = \max_{u \in [\ell+1]} \sum_{e \in \mathcal{T}_u} |p_1(e\,*) - p_2(e\,*)|$. Instead of comparing the probability of entire suffixes, both the metrics just defined compare their respective probabilities, together with the probability of any of their prefixes. An extended description of the various distinguishability parameters is provided in section 6.8.5. We can now require a positive distinguishability with our second assumption.

Assumption 6.2. The L_∞^{p}-distinguishability of the input RDP and the behaviour policy π^{b} is at least $\mu_0 > 0$.

Finally, in order to capture the mismatch in occupancy measure between the optimal policy and the behaviour policy, we introduce a key quantity called *single-policy RDP concentrability coefficient*, which extends the single-policy concentrability coefficient in MDPs to RDPs:

Definition 6.1. The *single-policy RDP concentrability coefficient* of an RDP $\mathbf{R}$ with episode horizon H and with respect to a policy π^{b} is defined as:

$$C_{\mathbf{R}}^* = \max_{t \in [H+1], q \in \mathcal{Q}_t, ao \in \mathcal{AO}} \frac{d_t^*(q, ao)}{d_t^{\mathsf{b}}(q, ao)} \tag{6.4}$$

This concentrability coefficient resembles similar notions of concentrability in MDPs, such as Xie, N. Jiang, et al. (2021) and Rashidinejad, Zhu, et al. (2021). It should be stressed, however, that those for MDPs are defined in terms of observation-action pairs (o, a), whereas $C_{\mathbf{R}}^*$ is defined in terms of *hidden* RDP states and actions-observations, (q, ao). It is worth remarking that $C^*_{\mathbf{R}}$ could be equivalently defined in terms of state-action pairs (q, a), only. Finally, in the special case where the RDP is Markovian – in which case it coincides with an episodic MDP – we have $\mathcal{Q} \subseteq \mathcal{O} \times [H+2]$ and $C^*_{\mathbf{R}}$ coincides with the standard single-policy concentrability coefficient for MDPs in Rashidinejad, Zhu, et al. (2021). This fact will be also shown in the proof of proposition 6.17.

6.4 RegORL: Learning an Episodic RDP

In this section, we present an algorithm for learning the transition function of an unknown RDP $\mathbf{R}$ from a dataset $\mathcal{D}$ of episodes generated by an unknown regular behaviour policy π^{b}. To simplify the presentation, we treat $\mathcal{D}$ as a multiset of traces in $\mathcal{T}_{H+1}$. The learning agent has only access to the non-Markovian traces in $\mathcal{D}$, and needs prior knowledge of $\mathcal{A}$, $\mathcal{R}$ and $\mathcal{O}$, but no prior knowledge of π^{b} and $\mathbf{R}$. Our algorithm is an adaptation of AdaCT (Balle Pigem 2013) to episodic RDPs, and we thus refer to the algorithm as AdaCT–H.

```
Function AdaCT–H($\mathcal{D}$, $\delta$)
  Input: Dataset $\mathcal{D}$ containing $N$ traces in $\mathcal{T}_{H+1}$, failure probability $0 < \delta < 1$
  Output: Set $\mathcal{Q}$ of RDP states, transition function $\tau : \mathcal{Q} \times \mathcal{A}\mathcal{O} \to \mathcal{Q}$
  $\mathcal{Q}_0 \leftarrow \{q_0\}$, $\mathcal{X}(q_0) \leftarrow \mathcal{D}$                         // initial state
  for $t = 0, \dots, H$ do
      $\mathcal{Q}_{\mathsf{c},t+1} \leftarrow \{qao \mid q \in \mathcal{Q}_t, ao \in \mathcal{A}\mathcal{O}\}$          // make candidate states
      foreach $qao \in \mathcal{Q}_{\mathsf{c},t+1}$ do
          $\mathcal{X}(qao) \leftarrow \{e_{t+1:H} \mid aroe_{t+1:H} \in \mathcal{X}(q)\}$          // compute suffixes
      end
      $q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}} \leftarrow \arg\max_{qao \in \mathcal{Q}_{\mathsf{c},t+1}} |\mathcal{X}(qao)|$          // most common candidate
      $\mathcal{Q}_{t+1} \leftarrow \{q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}\}$, $\tau(q_{\mathsf{m}}, a_{\mathsf{m}}o_{\mathsf{m}}) = q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}$          // promote candidate
      $\mathcal{Q}_{\mathsf{c},t+1} \leftarrow \mathcal{Q}_{\mathsf{c},t+1} \setminus \{q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}\}$          // remove from candidate states
      foreach $qao \in \mathcal{Q}_{\mathsf{c},t+1}$ do
          $Similar \leftarrow \{q' \in \mathcal{Q}_{t+1} \mid \text{not TestDistinct}(t, \mathcal{X}(qao), \mathcal{X}(q'), \delta)\}$
          if $Similar = \emptyset$ then                                   // promote candidate
              $\mathcal{Q}_{t+1} \leftarrow \mathcal{Q}_{t+1} \cup \{qao\}$, $\tau(q, ao) = qao$
          else                                                           // merge states
              $q' \leftarrow$ element in $Similar$
              $\tau(q, ao) = q'$, $\mathcal{X}(q') \leftarrow \mathcal{X}(q') \cup \mathcal{X}(qao)$
      end
  end
  return $\mathcal{Q}_0 \cup \dots \cup \mathcal{Q}_{H+1}$, $\tau$
  Function TestDistinct($t$, $\mathcal{X}_1$, $\mathcal{X}_2$, $\delta$)
      return $L^{\mathsf{p}}_\infty(\mathcal{X}_1, \mathcal{X}_2) \geq \sqrt{2\log(8(ARO)^{H-t}/\delta)/\min(|\mathcal{X}_1|, |\mathcal{X}_2|)}$
```

The intuition behind AdaCT–H is that due to proposition 6.1, two histories h_1 and h_2 should map to the same RDP state if they induce the same probability distribution on suffixes. AdaCT–H starts by adding an initial RDP state q_0 to $\mathcal{Q}_0$, whose suffixes are the full traces in $\mathcal{D}$ (line 1). The algorithm then iteratively constructs the state sets $\mathcal{Q}_1, \dots, \mathcal{Q}_{H+1}$. In each iteration $t \in [H+1]$, AdaCT–H creates a set of candidate states $\mathcal{Q}_{\mathsf{c},t+1}$ by extending all states in $\mathcal{Q}_t$ with symbols in $\mathcal{AO}$ (line 3). We use qao to simultaneously refer to a candidate state and its state-symbol prefix (q, ao). We associate each candidate state qao with a multiset of suffixes $\mathcal{X}(qao)$, which are traces in $\mathcal{T}_{H-t}$, obtained by selecting all suffixes in $\mathcal{X}(q)$ that start with action a and observation o (line 5).

Next, AdaCT–H finds the candidate state whose suffix multiset has maximum cardinality, and promotes this candidate to $\mathcal{Q}_{t+1}$ by defining the transition function τ accordingly (lines from 7 to 9). The algorithm then iterates over each remaining candidate states $qao \in \mathcal{Q}_{\mathsf{c},t+1}$, comparing the distribution on suffixes in $\mathcal{X}(qao)$ to those of states in $\mathcal{Q}_{t+1}$ (line 11). If the suffix distribution is different from that of each state in $\mathcal{Q}_{t+1}$, qao is promoted to $\mathcal{Q}_{t+1}$ (line 11), else qao is merged with a state $q' \in \mathcal{Q}_{t+1}$ that has a similar suffix distribution (line 16). Finally, AdaCT–H returns the set of RDP states $\mathcal{Q}$ and the associated transition function τ.

The function TestDistinct compares two multisets $\mathcal{X}_1$ and $\mathcal{X}_2$ of traces in $\mathcal{T}_{H-t}$ using the metric L_∞^{p}. For $i \in \{1,2\}$ and each trace $e \in \mathcal{T}_{H-t}$, let $\widehat{p}_i(e) = \sum_{x \in \mathcal{X}_i} \mathbb{I}(x = e)/|\mathcal{X}_i|$ be the empirical estimate of p_i, as the proportion of elements in $\mathcal{X}_i$ equal to e. TestDistinct compares $L_\infty^{\mathsf{p}}(\mathcal{X}_1, \mathcal{X}_2) \coloneqq L_\infty^{\mathsf{p}}(\widehat{p}_1, \widehat{p}_2)$ to a confidence threshold.

Markov Transformation We are now ready to connect the RDP learning phase with the MDP learning phase. RDPs do not respect the Markov property over their observations and rewards, if automaton states remain hidden. However, we can use the reconstructed transition function τ returned by AdaCT–H, extended over histories $\bar{\tau} : \mathcal{H} \to \mathcal{Q}$, to recover the Markov property. In what follows, we formalize the notion of Markov transformation and the properties that its outputs satisfy.

Definition 6.2. Let $e_{0:H} \in \mathcal{T}_{H+1}$ be an episode collected from an RDP $\mathbf{R}$ and a policy π^{b} that is regular in $\mathbf{R}$. The *Markov transformation* of e_H with respect to $\mathbf{R}$ is the episode constructed as $a_0 r_0 q_1 \dots a_H r_H q_{H+1}$, where $q_t = \bar{\tau}(h_t)$ and $h_t = a_0 o_0 \cdots a_{t-1} o_{t-1} \in \mathcal{H}_t$, $t \in [H+1]$. The Markov transformation of a dataset $\mathcal{D}$ is the Markov transformation of all the episodes it contains.

A Markov transformation discards all observations from $\mathcal{D}$ and replaces them with RDP states generated by $\bar{\tau}$. The dataset so constructed can be seen as generated from an MDP, which we define next.

Definition 6.3. The episodic MDP *associated to* an episodic RDP $\mathbf{R}$ is $\mathbf{M}_{\mathbf{R}} = \langle \mathcal{Q}, \mathcal{A}, \mathcal{R}, T, \theta_{\mathsf{r}}, H \rangle$, where $T(q' \mid qa) = \sum_{o \in \mathcal{O}} \mathbb{I}(q' = \tau(q, ao))\, \theta_{\mathsf{o}}(o \mid q, a)$ for each $(q, a, q') \in \mathcal{Q} \times \mathcal{A} \times \mathcal{Q}$.

The associated MDP in definition 6.3 is the decision process that corresponds to the Markov transformation of definition 6.2: any episode produced with the Markov transformation can be equivalently seen as being generated from the associated MDP, in the sense of the following proposition.

Proposition 6.3. *Let $e_{0:H}$ be an episode sampled from an episodic RDP $\mathbf{R}$ under a regular policy $\pi \in \Pi_{\mathbf{R}}$, with $\pi(a \mid h) = \pi_{\mathsf{r}}(a \mid \bar{\tau}(h))$. If e'_H is the Markov transformation of e_H with respect to $\mathbf{R}$, then $\mathbb{P}(e'_H \mid \mathbf{R}, \pi) = \mathbb{P}(e'_H \mid \mathbf{M_R}, \pi_{\mathsf{r}})$, where $\mathbf{M_R}$ is the MDP associated to $\mathbf{R}$.*

Proof. See page 149. □

Rewards are not affected by the Markov transformation, only observations, implying the following.

Proposition 6.4. *Let $\pi \in \Pi_{\mathbf{R}}$ be a regular policy in $\mathbf{R}$ such that $\pi(a \mid h) = \pi_{\mathsf{r}}(a \mid \bar{\tau}(h))$. Then $V^{\pi}_{\mathbf{R}} = V^{\pi_{\mathsf{r}}}_{\mathbf{M_R}}$, where $V^{\pi}_{\mathbf{R}}$ and $V^{\pi_{\mathsf{r}}}_{\mathbf{M_R}}$ are the values from the initial distributions in the respective decision processes.*

Proof. See page 150. □

Corollary 6.5. *Given $\varepsilon \in (0, H]$, if $\pi_{\mathsf{r}} : \mathcal{Q} \to \Delta(\mathcal{A})$ is an ε-optimal policy of $\mathbf{M_R}$, the MDP associated to some RDP $\mathbf{R}$, then, $\pi(a \mid h) = \pi_{\mathsf{r}}(a \mid \bar{\tau}(h))$ is ε-optimal in $\mathbf{R}$.*

Summarizing, from proposition 6.3, if $\mathcal{D}_{\mathsf{m}}$ is the Markov transformation of a dataset $\mathcal{D}$ with respect to an RDP $\mathbf{R}$, then, $\mathcal{D}_{\mathsf{m}}$ can be seen as being generated from the associated MDP $\mathbf{M_R}$. Hence, any offline RL algorithm for MDPs can be used for learning in $\mathcal{D}_{\mathsf{m}}$. Moreover, according to corollary 6.5, any solution for $\mathbf{M_R}$ can be translated via $\bar{\tau}$ into a policy for the original RDP, with the same guarantees.

Complete Algorithm The complete procedure is illustrated in algorithm 6.1. Initially, the input dataset $\mathcal{D}$ is separated in two halves. The first portion is used for learning the transition function of the unknown RDP with AdaCT–H. If an upper bound $\overline{Q}$ on $|\mathcal{Q}|$ is available, it can optionally be provided to compute a more appropriate failure parameter for AdaCT–H. If not available, we adopt the upper bound of $2(AO)^H$ states, which is valid for any instance, due to histories having finite length. As we will see in theorem 6.6, this would only contribute linearly in H to the required dataset size. The output function computed by AdaCT–H is then used to compute a Markov transformation of the second phase, as specified in definition 6.2. The resulting dataset, now Markovian, can be passed to a generic offline RL algorithm, which we represent with the function OfflineRL$(\mathcal{D}, \varepsilon, \delta)$. In section 6.8.6, we instantiate it for a specific state-of-the-art offline RL algorithm.

Algorithm 6.1: Full procedure (RegORL)

```
Input: Dataset $\mathcal{D}$, accuracy $\varepsilon \in (0, H]$, failure probability $0 < \delta < 1$, (optionally)
       upper bound $\overline{Q}$ on $|\mathcal{Q}|$
Output: Policy $\hat{\pi} : \mathcal{H} \to \Delta(\mathcal{A})$
$\mathcal{D}_1, \mathcal{D}_2 \leftarrow$ separate $\mathcal{D}$ into two datasets of the same size
$\mathcal{Q}, \tau \leftarrow$ AdaCT–H($\mathcal{D}_1, \delta/(4AO\overline{Q})$), where $\overline{Q} = 2(AO)^H$ if not provided
$\mathcal{D}'_2 \leftarrow$ Markov transformation of $\mathcal{D}_2$ with respect to $\bar{\tau}$ as in definition 6.2
$\hat{\pi}_{\mathsf{m}} \leftarrow$ OfflineRL($\mathcal{D}'_2, \varepsilon, \delta/2$)
return $\hat{\pi} : h \mapsto \hat{\pi}_{\mathsf{m}}(\bar{\tau}(h))$
```

6.5 Theoretical Guarantees

We now turn to theoretical performance guarantees of RegORL. Our main performance result is a sample complexity bound in theorem 6.7, ensuring that, for any accuracy $\varepsilon \in (0, H]$, RegORL finds an ε-optimal policy. We also report a sample complexity bound for AdaCT–H in theorem 6.6, and an alternative bound in theorem 6.8. For comparison, the sample complexity bound for AdaCT from Balle, J. Castro, et al. (2013) is

$$\widetilde{O}\left(\frac{Q^4 A^2 O^2 H^5 \log(1/\delta)}{\varepsilon^2} \max\left\{\frac{1}{\mu_0^2}, \frac{H^4 O^2 A^2}{\varepsilon^4}\right\}\right) \tag{6.5}$$

We achieve a tighter bound by using Bernstein's inequalities and exploiting the finiteness of histories.

Theorem 6.6. *Consider a dataset $\mathcal{D}$ of episodes sampled from an RDP $\mathbf{R}$ and a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$. With probability $1-\delta$, the output of* AdaCT–H($\mathcal{D}, \delta/(2QAO)$) *is the transition function of the minimal RDP equivalent to $\mathbf{R}$, provided that $|\mathcal{D}| \geq N_\delta$, where*

$$N_\delta := \frac{21 \log(8QAO/\delta)}{d^{\mathsf{b}}_{\min}\, \mu_0} \sqrt{H \log(2ARO)} \in \widetilde{O}\left(\frac{\sqrt{H}}{d^{\mathsf{b}}_{\min}\, \mu_0}\right) \tag{6.6}$$

$d^{\mathsf{b}}_{\min} := \min\{d^{\mathsf{b}}_t(q, ao) \mid t \in [H+1], q \in \mathcal{Q}_t, ao \in \mathcal{AO}, d^{\mathsf{b}}_t(q, ao) > 0\}$ *is the minimal occupancy distribution, and μ_0 is the L^{p}_∞-distinguishability.*

Proof. See section 6.8.3. □

This theorem tells us that the sample complexity of AdaCT–H, to return a minimal RDP, is inversely proportional to μ_0, the L^{p}_∞-distinguishability of $\mathbf{R}$ and π^{b}, and the minimal occupancy $d^{\mathsf{b}}_{\min}$. Note that $d^{\mathsf{b}}_{\min} \leq 1/(QOA)$. The bound also depends on Q, the number of RDP states, implicitly through $d^{\mathsf{b}}_{\min}$, and explicitly via a logarithmic term. In the absence of prior knowledge of Q, one may use in the argument of algorithm 6.1 the worst-case upper bound $\overline{Q} = 2(AO)^H$. The sample complexity would then have an additional linear term in H, since $\overline{Q}$ is only used in the logarithmic term to set the appropriate value of δ. However, this will not impact the value of the $d^{\mathsf{b}}_{\min}$ term.

Theorem 6.6 is a sample complexity guarantee for the first phase of the algorithm, which learns τ, the structure of the minimal RDP underlying the domain. If δ is the desired failure probability of the complete algorithm, `RegORL` executes AdaCT–H so that its success probability is at least $1 - \delta/2$. This means that with the same probability, $\mathcal{D}'_2$ is an MDP dataset with the properties listed in section 6.4. As a consequence, provided that OfflineRL is some generic $(\varepsilon, \delta/2)$-PAC offline RL algorithm for MDPs, the output of `RegORL` is an ε-optimal policy with probability $1 - \delta$.

Theorem 6.7. *Consider a dataset $\mathcal{D}$ of episodes sampled from an RDP $\mathbf{R}$ and a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$. For any $\varepsilon \in (0, H]$ and $0 < \delta < 1$, if* OfflineRL *is an $(\varepsilon, \delta/2)$-PAC offline algorithm for MDPs with sample complexity N_{m}, then, the output of `RegORL`$(\mathcal{D}, \varepsilon, \delta)$ is an ε-optimal policy in $\mathbf{R}$, with probability at least $1 - \delta$, provided that $|\mathcal{D}| \geq 2 \max\{N_{\delta/2}, N_{\mathsf{m}}\}$.*

As we can see, the sample complexity requirement is separate for the two phases. While $N_{\delta/2}$ is due to the RDP learning component, defined in eq. (6.6), the quantity N_{m} completely depends on the offline RL algorithm for MDPs that is adopted. Among other terms, the performance guarantees of offline algorithms can often be characterized through the single-policy concentrability for MDPs C^*. However, since states become observations in the associated MDP, due to the properties of proposition 6.3, C^* coincides with $C^*_{\mathbf{R}}$, the RDP single-policy concentrability of definition 6.1.

In section 6.8.6, we demonstrate a specific instantiation of `RegORL` with an off-the-shelf offline RL algorithm from the literature by G. Li, L. Shi, et al. (2022). This yields the following requirement for N_{m}:

$$N_{\mathsf{m}} \geq \frac{c\, H^3 Q C^*_{\mathbf{R}} \log \frac{2HN_{\mathsf{m}}}{\delta}}{\varepsilon^2} \tag{6.7}$$

for a constant $c > 0$.

To eliminate the dependence that theorem 6.6 has on $d^{\mathsf{b}}_{\min}$, we develop a variant of AdaCT–H which does not learn a complete RDP. Rather, it only reconstructs a subset of states that are likely under the behaviour policy. The algorithm, which we call AdaCT–H–A (with ‘A’ standing for “approximation”), is defined at page 155. Theorem 6.8 is an upper bound on the sample complexity of AdaCT–H–A, that takes the accuracy ε as input and returns the transition function of an $\varepsilon/2$-approximate RDP $\mathbf{R}'$, whose optimal policy is $\varepsilon/2$-optimal for the original RDP $\mathbf{R}$. By performing a Markov transformation for $\mathbf{R}'$, and by using an $(\varepsilon/2, \delta/2)$-PAC offline algorithm for MDPs, we can compute an ε-optimal policy for $\mathbf{R}$. The total sample complexity can be combined in the same way as in theorem 6.7. Also, this theorem does not rely on assumption 6.1, because a finite $C^*_{\mathbf{R}}$ suffices.

Theorem 6.8. *Consider a dataset $\mathcal{D}$ of episodes sampled from an RDP $\mathbf{R}$ and a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$. With probability $1-\delta$, the output of* AdaCT–H–A*, called with $\mathcal{D}$, $\delta/(2QAO)$ and $\varepsilon \in (0, H]$ in input, is the transition function of an $\varepsilon/2$-approximate RDP $\mathbf{R}'$, provided that $|\mathcal{D}| \geq N'_\delta$, where*

$$N'_\delta \coloneqq \frac{504HQAOC^*_{\mathbf{R}'} \log(16QAO/\delta)}{\varepsilon\, \mu_0} \sqrt{H \log(2ARO)} \in \widetilde{O}\left(\frac{H^{3/2}QAOC^*_{\mathbf{R}'}}{\varepsilon\, \mu_0}\right)$$

Proof. See section 6.8.4. □

6.6 Sample Complexity Lower Bound

The main result of this section is theorem 6.9, a sample complexity lower bound for offline RL in RDPs. It shows that the dataset size required by *any* RL algorithm scales with the relevant parameters.

Theorem 6.9. *For any $(C^*_{\mathbf{R}}, H, \varepsilon, \mu_0)$ satisfying $C^*_{\mathbf{R}} \geq 2$, $H \geq 2$ and $\varepsilon \leq H\mu_0/64$, there exists an RDP with horizon H, L_1^{p}-distinguishability μ_0 and a regular behaviour policy π^{b} with RDP single-policy concentrability $C^*_{\mathbf{R}}$, such that if $\mathcal{D}$ has been generated using π^{b} and $\mathbf{R}$, and*

$$|\mathcal{D}| \notin \Omega\left(\frac{H}{\mu_0} + \frac{C^*_{\mathbf{R}} H^2}{\varepsilon^2}\right) \tag{6.8}$$

then, for any algorithm $\mathfrak{A} : \mathcal{D} \mapsto \widehat{\pi}$ returning non-Markov deterministic policies, the probability that $\widehat{\pi}$ is not ε-optimal is at least $1/4$.

Proof. See section 6.8.7. □

The proof relies on worst-case RDP instances that carefully combine two-armed bandits with noisy parity functions. This last component allows capturing the difficulty of learning in presence of temporal dependencies. Figure 6.1 shows an RDP in this class. At the beginning of each episode, the observation causes a transition towards either the bandit component (bottom branch) or the noisy parity function (top branches). Acting optimally in the two parity branches requires predicting the output of a parity function, which depends on some unknown binary code (of length 3, in the example). The first term in theorem 6.9 is due to this component, because the code scales linearly with H, while the amount of information revealed about the code is controlled by μ_0. The second term is caused by the required optimality in the bandit. Since the number of states scales linearly with H, the

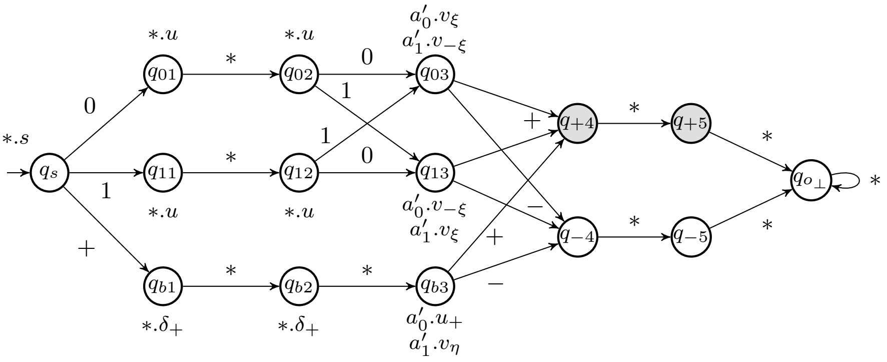


Figure 6.1. One episodic RDP instance $\mathbf{R}_{101,1} \in \mathbb{R}(L, H, \xi, \eta)$, associated to the parity function f_{101}, with code 101, and the optimal arm a'_1. Only the gray states are rewarding. The code length is $L = |101| = 3$, the horizon $H = 5$, the noise parameter is $\xi > 0$ and the bandit bonus parameter is $\eta > 0$. The transition function only depends on the observations, not the actions. The output distributions are: $u = \text{Unif}\{0, 1\}$, $u_+ = \text{Unif}\{+, -\}$, $v_\alpha(+) = (1 + \alpha)/2$, $v_\alpha(-) = (1 - \alpha)/2$. The star denotes any symbol. If the label of a state q is $a.d$, then the observation function is $\theta_\mathsf{o}(a \mid q) = d$. Refer to section 6.8.7 for details.

lower bound could be also expressed in terms of Q, instead of H. Moreover, by simply extending the number of action in the branch of the bandit, it should be possible to make A appear in the right term.

Differently from this lower bound, the parameter μ_0, appearing in the upper bounds of theorems 6.6 and 6.8, is a L^p_∞-distinguishability. However, the two are related, since we always have $L^\mathsf{p}_1(q, q') \geq L^\mathsf{p}_\infty(q, q')$. Intuitively, the L^p_1-distinguishability accounts for all the information that is available as differences for each prefix length. The L^p_∞-distinguishability, on the other hand, quantifies the maximum the difference in probability associated to one specific suffix, the one maximizing the distance. This is the information used by the algorithm and the one appearing in the two upper bounds. A similar lower bound also appears in the paper this chapter refers to (Cipollone, Jonsson, et al. 2024). However, the definition of L^p_1 that can be found here is slightly different from the one of this paper, as it is based on a double sum, $\sum_{u \in [H-t+1], e \in \mathcal{T}_{u+1}}$. In particular, there are RDPs in which the parameter used here is exponentially smaller in H, than the one found in the paper. As a result, the expression in (6.8) is stronger. An extended discussion on distinguishability parameters can be found in section 6.8.5

6.7 Discussion

In this chapter, we proposed an algorithm for Offline RL in episodic Regular Decision Processes, when both the RDP and the behaviour policy are unknown. Our algorithm exploits automata learning techniques to reduce the problem of offline RL in RDPs, in which observations and rewards are non-Markovian, into standard offline RL for MDPs. We provide the first high-probability sample complexity guarantees for this setting, as well as a new lower bound that shows how its complexity relates to the parameters that characterize the decision process and the behaviour policy. We identify the RDP single-policy concentrability as an analogous quantity to the one used for MDPs in the literature.

Although the results obtained in this chapter are specific for offline RL, we could loosely compare the lower bound obtained here with the one of theorem 5.14 in the previous chapter. We observe that, the general lower bound can be exponential for the general case. However, when the RDP can be characterized more precisely, as we did with distinguishability parameters here, for some instances, the required number of samples decreases significantly. Comparing the lower and the two upper bounds, instead, we notice that the most significant difference is that they use two different distinguishability parameters, defined through L_1^{p} and L_∞^{p}. This is a meaningful gap, that we aim to tighten in a future work. Specifically, we currently hypothesize that is the upper bound which could be significantly improved, with some non-trivial algorithmic changes. Finally, our results have strong implications for online learning in RDPs, which is a relevant setting to be explored.

6.8 Proofs

This section contains every proof for this chapter. The reader may skip this section and refer to it as needed.

6.8.1 Preliminaries

We first state Hoeffding's inequality for Bernoulli variables. In what follows we take log to be the natural logarithm.

Lemma 6.10 (Hoeffding's inequality)**.** *Let $X_1, \dots, X_N$ be N independent random Bernoulli variables with the same expected value $\mathbb{E}[X_1] = p$, and let $\widehat{p}_N = \sum_{i=1}^N X_i/N$ be an empirical estimate of p. Then, for any $\delta \in (0,1)$,*

$$\mathbb{P}\left(|\widehat{p}_N - p| \geq \sqrt{\frac{\log(2/\delta)}{2N}} \right) \leq \delta. \tag{6.9}$$

An alternative to Hoeffding's inequality is the empirical Bernstein inequality, which can be expressed as follows for Bernoulli variables (Maurer and Pontil 2009; Dann, Lattimore, et al. 2017).

Lemma 6.11 (Empirical Bernstein inequality)**.** *Let $X_1, \dots, X_N$ be N independent random Bernoulli variables with the same expected value $\mathbb{E}[X_1] = p$, and let $\widehat{p}_N = \sum_{i=1}^N X_i / N$ be an empirical estimate of p. Then, for any $\delta \in (0,1)$,*

$$\mathbb{P}\left(|\widehat{p}_N - p| \geq \sqrt{\frac{2\widehat{p}\log(4/\delta)}{N}} + \frac{14\log(4/\delta)}{3N} \right) \leq \delta. \tag{6.10}$$

If $X \sim p_X$ is a discrete random variable, the entropy of X is

$$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x) \tag{6.11}$$

Further, for $x \in (0,1)$, we define the binary entropy function as $H_2(x) = -x\log(x) - (1-x)\log(1-x)$. If $(X,Y) \sim p_{XY}$ are two discrete variables, the conditional entropy is $H(Y \mid X) = \sum_{x \in \mathcal{X}} p_X(x)\, H(Y \mid X = x)$. The mutual information is $I(X;Y) = I(Y;X) = D_{KL}(p_{XY} \parallel p_X \cdot p_Y)$, where D_{KL} is the Kullback–Leibler divergence. If X, Y, Z are three random variables, we write $X \to Y \to Z$ if the conditional distribution of Z does not depend on X, given Y. With these definitions, we state Fano's inequality, as one can find in Cover and J. A. Thomas (2006), (2.140).

Theorem 6.12 (Fano's inequality)**.** *Let $X \to Y \to \hat{X}$, for $X, \hat{X} \in \mathcal{X}$ and $P_e = \mathbb{P}(\hat{X} \neq X)$. Then,*

$$H_2(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X \mid Y). \tag{6.12}$$

6.8.2 RDP Properties

Proposition 6.1. *Consider an RDP $\mathbf{R}$, a regular policy $\pi \in \Pi_{\mathbf{R}}$ and two histories h_1 and h_2 in $\mathcal{H}_t$, $t \in [H]$, such that $\bar{\tau}(h_1) = \bar{\tau}(h_2)$. For each suffix $e_{t+1:H} \in \mathcal{T}_{H-t}$, the probability of generating $e_{t+1:H}$ is the same for h_1 and h_2, i.e. $\mathbb{P}(e_{t+1:H} \mid h_1, \pi, \mathbf{R}) = \mathbb{P}(e_{t+1:H} \mid h_2, \pi, \mathbf{R})$.*

Proof. By induction on t. For $t = H$, all histories in $\mathcal{H}_H$ generate the empty suffix in $(\mathcal{ARO})^0$ with probability 1 (the stop symbol is omitted). For $t < H$, the probability of generating a suffix $aroe_{t+2:H}$ is

$$\begin{aligned} \mathbb{P}(aroe_{t+2:H} \mid h_1, \pi) &= \pi(a \mid h_1)\, \mathbb{P}(r, o \mid \bar{\tau}(h_1), a, \mathbf{R})\, \mathbb{P}(e_{t+2:H} \mid h_1 ao, \pi) \\ &= \pi(a \mid h_2)\, \mathbb{P}(r, o \mid \bar{\tau}(h_2), a, \mathbf{R})\, \mathbb{P}(e_{t+2:H} \mid h_2 ao, \pi) = \mathbb{P}(aroe_{t+2:H} \mid h_2, \pi) \end{aligned} \tag{6.13}$$

where we have used the fact that π is regular, $\bar{\tau}(h_1) = \bar{\tau}(h_2)$, $\bar{\tau}(h_1ao) = \tau(\bar{\tau}(h_1), ao) = \tau(\bar{\tau}(h_2), ao) = \bar{\tau}(h_2ao)$, and the induction hypothesis. □

Proposition 6.2. *Each RDP* $\mathbf{R}$ *has at least one optimal policy* $\pi^* \in \Pi_{\mathbf{R}}$*.*

Proof. Given $\mathbf{R}$, consider any optimal policy $\pi^* : \mathcal{H} \to \Delta(\mathcal{A})$, not necessarily regular. We prove the statement by constructing a policy π and showing by induction on $t \in [H+1]$ that π is both optimal and regular. The base case is given by $t = H$. In this case, for an arbitrary $a \in \mathcal{A}$, define $\pi(h) \coloneqq \delta_a$ for each history $h \in \mathcal{H}_H$. Since $V_H^\pi(h) = 0$ by definition, π is optimal for each history $h \in \mathcal{H}_H$, and regular since it always selects the same action.

For $t \leq H$, we first construct a new policy π_{c} which is the composition of policies π^* and π. Concretely, for each history $h \in \mathcal{H}_u$ such that $u \leq t$, $\pi_{\mathsf{c}}(h) = \pi^*(h)$ acts according to π^*, while for each history $h \in \mathcal{H}_u$ such that $u > t$, $\pi_{\mathsf{c}}(h) = \pi(h)$ acts according to π. Clearly, π_{c} is an optimal policy for $\mathbf{R}$ since π^* is optimal and since by induction, π is optimal for histories in $\mathcal{H}_u$, $u > t$.

Consider a pair of histories h_1 and h_2 in $\mathcal{H}_t$ such that $\bar{\tau}(h_1) = \bar{\tau}(h_2)$ but $\pi_{\mathsf{c}}(h_1) \neq \pi_{\mathsf{c}}(h_2)$. Define $\pi(h_1) \coloneqq \pi(h_2) \coloneqq \pi_{\mathsf{c}}(h_1)$. Since the value function can be written as an expectation over suffixes, due to proposition 6.1 and the fact that π is regular for histories in $\mathcal{H}_u$, $u > t$, we have $V_t^\pi(h_1) = V_t^\pi(h_2)$. Since π_{c} is the same as π for histories in $\mathcal{H}_u$, $u > t$, this implies $V_t^\pi(h_1) = V_t^{\pi_{\mathsf{c}}}(h_1) \leq V_t^{\pi_{\mathsf{c}}}(h_2)$ since π_{c} is optimal for h_2. If we were to instead define $\pi(h_1) \coloneqq \pi(h_2) \coloneqq \pi_{\mathsf{c}}(h_2)$, we would obtain $V_t^{\pi_{\mathsf{c}}}(h_2) \leq V_t^{\pi_{\mathsf{c}}}(h_1)$. The only possibility is $V_t^{\pi_{\mathsf{c}}}(h_1) = V_t^{\pi_{\mathsf{c}}}(h_2)$, which is the same value achieved by the policy π. Hence, π is optimal for h_1 and h_2.

We now repeat the same procedure for each pair of histories h_1 and h_2 in $\mathcal{H}_t$ such that $\bar{\tau}(h_1) = \bar{\tau}(h_2)$ but $\pi_{\mathsf{c}}(h_1) \neq \pi_{\mathsf{c}}(h_2)$. If necessary, we complete the definition of π by copying the action choices of π_{c}. The resulting policy π is optimal for each history $h \in \mathcal{H}_t$, and regular since it makes the same action choices for each pair of histories h_1 and h_2 in $h \in \mathcal{H}_t$ such that $\bar{\tau}(h_1) = \bar{\tau}(h_2)$. □

Proposition 6.3. *Let* $e_{0:H}$ *be an episode sampled from an episodic RDP* $\mathbf{R}$ *under a regular policy* $\pi \in \Pi_{\mathbf{R}}$*, with* $\pi(a \mid h) = \pi_{\mathsf{r}}(a \mid \bar{\tau}(h))$*. If* e'_H *is the Markov transformation of* e_H *with respect to* $\mathbf{R}$*, then* $\mathbb{P}(e'_H \mid \mathbf{R}, \pi) = \mathbb{P}(e'_H \mid \mathbf{M}_{\mathbf{R}}, \pi_{\mathsf{r}})$*, where* $\mathbf{M}_{\mathbf{R}}$ *is the MDP associated to* $\mathbf{R}$*.*

Proof. For $t \in [H+2]$, let $e_t \in \mathcal{T}_t = (\mathcal{ARO})^t$ be an episode prefix in $\mathbf{R}$, $\phi(e_t) \in \mathcal{T}'_t = (\mathcal{ARQ})^t$ its Markov transformation and $e'_t \in \mathcal{T}'_t$ an episode of the associated MDP. The function $\phi : \mathcal{T} \to \mathcal{T}'$ transforms the observations according to $\bar{\tau}$, and preserves actions and rewards. The statement says that $\mathbb{P}(\phi(e_t) \mid \mathbf{R}, \pi) = \mathbb{P}(e'_t \mid \mathbf{M}_{\mathbf{R}}, \pi_{\mathsf{r}})$ (note that $\phi(e_t)$ and e'_t are distinct random variables). We prove this by induction. For $t = 0$, we recall that the irrelevant quantities a_0, r_0 are constant and,

$$\begin{aligned} \mathbb{P}(\phi(a_0 r_0 o_0) = a_0 r_0 q \mid \mathbf{R}, \pi) &= \sum_{o \in \mathcal{O}} \mathbb{I}(\tau(q_0, a_0 o) = q)\, \theta_{\mathsf{o}}(o \mid q_0, a_0) \\ &= T(q \mid q_0 a_0) \\ &= \mathbb{P}(e'_0 = a_0 r_0 q \mid \mathbf{M_R}, \pi_{\mathsf{r}}) \end{aligned} \tag{6.14}$$

where $T : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{Q})$ is the transition function of $\mathbf{M_R}$, from definition 6.3. Due to the role of the dummy action, $T(q_0 a_0)$ is the initial distribution of the MDP.

For the inductive step, assume that $\mathbb{P}(\phi(e_{t-1}) \mid \mathbf{R}, \pi) = \mathbb{P}(e'_{t-1} \mid \mathbf{M_R}, \pi_{\mathsf{r}})$. Then, for any $e' \in \mathcal{T}'_{t-1}$, $arq \in \mathcal{ARQ}$, if q' is the last element of e', we have

$$\begin{aligned} &\mathbb{P}(\phi(e_t) = e' arq \mid \mathbf{R}, \pi) = \\ &\quad = \mathbb{P}(\phi(e_{t-1}) = e' \mid \mathbf{R}, \pi)\, \mathbb{P}(a_t r_t q_{t+1} = arq \mid \phi(e_{t-1}) = e', \mathbf{R}, \pi) && (6.15) \\ &\quad = \mathbb{P}(e'_{t-1} = e' \mid \mathbf{M_R}, \pi_{\mathsf{r}})\, \mathbb{P}(a_t r_t q_{t+1} = arq \mid q_t = q', \mathbf{R}, \pi) && (6.16) \\ &\quad = \mathbb{P}(e'_{t-1} = e' \mid \mathbf{M_R}, \pi_{\mathsf{r}})\, \pi_{\mathsf{r}}(a \mid q')\, \theta_{\mathsf{r}}(r \mid q', a) \\ &\qquad \cdot \sum_{o \in \mathcal{O}} \theta_{\mathsf{o}}(o \mid q', a)\, \mathbb{I}(q = \tau(q', ao)) && (6.17) \\ &\quad = \mathbb{P}(e'_{t-1} = e' \mid \mathbf{M_R}, \pi_{\mathsf{r}})\, \pi_{\mathsf{r}}(a \mid q')\, \theta_{\mathsf{r}}(r \mid q', a)\, T(q \mid q'a) && (6.18) \\ &\quad = \mathbb{P}(e'_t = e' arq \mid \mathbf{M_R}, \pi_{\mathsf{r}}) && (6.19) \end{aligned}$$

where, in (6.16), we have used the induction hypothesis and the fact that $a_t r_t q_{t+1}$ are Markov in q' by regularity of the policy. □

Proposition 6.4. *Let $\pi \in \Pi_{\mathbf{R}}$ be a regular policy in $\mathbf{R}$ such that $\pi(a \mid h) = \pi_{\mathsf{r}}(a \mid \bar{\tau}(h))$. Then $V^{\pi}_{\mathbf{R}} = V^{\pi_{\mathsf{r}}}_{\mathbf{M_R}}$, where $V^{\pi}_{\mathbf{R}}$ and $V^{\pi_{\mathsf{r}}}_{\mathbf{M_R}}$ are the values from the initial distributions in the respective decision processes.*

Proof. The statement is composed of two parts. First, we show that $V^{\pi}_{\mathbf{R}} = V^{\pi_{\mathsf{r}}}_{\mathbf{M_R}}$, which is a direct consequence of proposition 6.3. Following the same convention as in the proof of proposition 6.3, we use $\mathcal{T}'_t = (\mathcal{ARQ})^t$ and ϕ for the Markov transformation. Then,

$$V^{\pi}_{\mathbf{R}} = \sum_{r_1 \dots r_H \in \mathcal{R}^{H+1}} \mathbb{P}(r_{1:H} = r_1 \dots r_H \mid \mathbf{R}, \pi) \sum_{i=1}^{H} r_i \tag{6.20}$$

$$= \sum_{e' \in \mathcal{T}'_{H+1}} \mathbb{P}(\phi(e_H) = e' \mid \mathbf{R}, \pi) \sum_{i=1}^{H} r_i \tag{6.21}$$

$$= \sum_{e' \in \mathcal{T}'_{H+1}} \mathbb{P}(e'_H = e' \mid \mathbf{M_R}, \pi_\mathsf{r}) \sum_{i=1}^{H} r_i \tag{6.22}$$

$$= V^{\pi_\mathsf{r}}_{\mathbf{M_R}} \tag{6.23}$$

For the second part of the statement, let $\Pi_\mathbf{R}$ and $\Pi_\mathbf{M}$ be the regular and the Markov policies in $\mathbf{R}$ and $\mathbf{M_R}$, respectively. Then, using proposition 6.2 and the first part of this statement,

$$V^*_\mathbf{R} = \max_{\pi \in \Pi_\mathbf{R}} V^\pi_\mathbf{R} = \max_{\pi \in \Pi_\mathbf{R}} V^{\pi_\mathsf{r}}_{\mathbf{M_R}} = \max_{\pi_\mathsf{r} \in \Pi_\mathbf{M}} V^{\pi_\mathsf{r}}_{\mathbf{M_R}} = V^*_{\mathbf{M_R}} \tag{6.24}$$

□

6.8.3 Sample Complexity of `AdaCT-H`

In this section, we prove theorem 6.6, which is a high-probability upper bound on the sample complexity of AdaCT–H. The first two lemmas are adaptations of lemmas 19 and 20 in Balle, J. Castro, et al. (2013) to the episodic setting.

Lemma 6.13. *For $t \in [H+1]$, let $\mathcal{X}_1$ and $\mathcal{X}_2$ be multisets sampled from distributions p_1 and p_2 in $\Delta(\mathcal{T}_{H-t})$. If $p_1 = p_2$, then* TestDistinct*$(t, \mathcal{X}_1, \mathcal{X}_2, \delta)$ returns False with probability $1-\delta$.*

Proof. For each $i \in \{1,2\}$ and each trace $e \in \mathcal{T}_{H-t}$, we can view each episode as a random Bernoulli variable with expected value $p_i(e)$ that takes value 1 if we observe e, and 0 otherwise. Let $\widehat{p}_i(e) = \sum_{x \in \mathcal{X}_i} \mathbb{I}(x = e)/|\mathcal{X}_i|$ be the empirical estimate of p_i, i.e. the proportion of elements in $\mathcal{X}_i$ equal to e. For each $i \in \{1,2\}$, each $u \in [H-t]$ and each prefix $e_{0:u} \in \mathcal{T}_{u+1}$, Hoeffding's inequality yields

$$\mathbb{P}\left(|\widehat{p}_i(e_{0:u}\,*) - p_i(e_{0:u}\,*)| \geq \sqrt{\frac{\log(2/\delta_\mathsf{s})}{2|\mathcal{X}_i|}} \right) \leq \delta_\mathsf{s} \tag{6.25}$$

The total number of non-empty prefixes of $\mathcal{T}_{H-t}$ equals a geometric sum:

$$(ARO)^1 + \cdots + (ARO)^{H-t} = \frac{(ARO)^{H+1-t} - 1}{ARO - 1} - 1 \leq 2(ARO)^{H-t} \tag{6.26}$$

Choosing $\delta_\mathsf{s} = \delta/4(ARO)^{H-t}$ and taking a union bound implies that the above inequality holds for each $i \in \{1,2\}$ and each $e_{0:u}$ simultaneously with probability $1 - 4(ARO)^{H-t}\delta_\mathsf{s} = 1 - \delta$, implying

$$L^\mathsf{p}_\infty(\mathcal{X}_1, \mathcal{X}_2) = \max_{u, e_{0:u}} |\widehat{p}_1(e_{0:u}*) - \widehat{p}_2(e_{0:u}*)| \tag{6.27}$$

$$\leq L^\mathsf{p}_\infty(p_1, p_2) + \sqrt{\frac{\log(2/\delta_\mathsf{s})}{2|\mathcal{X}_1|}} + \sqrt{\frac{\log(2/\delta_\mathsf{s})}{2|\mathcal{X}_2|}} \tag{6.28}$$

$$\leq 0 + 2\sqrt{\frac{\log(2/\delta_{\mathsf{s}})}{2\min(|\mathcal{X}_1|, |\mathcal{X}_2|)}} \tag{6.29}$$

$$= \sqrt{\frac{2\log(8(ARO)^{H-t}/\delta)}{\min(|\mathcal{X}_1|, |\mathcal{X}_2|)}}, \tag{6.30}$$

which is precisely the condition under which TestDistinct$(t, \mathcal{X}_1, \mathcal{X}_2, \delta)$ returns False. □

Lemma 6.14. *For $t \in [H+1]$, let $\mathcal{X}_1$ and $\mathcal{X}_2$ be multisets sampled from distributions p_1 and p_2 in $\Delta(\mathcal{T}_{H-t})$. If the L^{p}_{∞}-distinguishability of π^{b} is μ_0, then TestDistinct$(t, \mathcal{X}_1, \mathcal{X}_2, \delta)$ returns True with probability $1-\delta$, provided that*

$$\min(|\mathcal{X}_1|, |\mathcal{X}_2|) \geq \frac{8}{\mu_0^2}\Big(\log(2(ARO)^{H-t}) + \log(4/\delta)\Big) \tag{6.31}$$

Proof. Using the same argument as in the proof of lemma 6.13, Hoeffding's inequality yields

$$\mathbb{P}\left(|\widehat{p}_i(e_{0:u}\,*) - p_i(e_{0:u}\,*)| > \sqrt{\frac{\log(2/\delta_{\mathsf{s}})}{2|\mathcal{X}_i|}}\right) \leq \delta_{\mathsf{s}}, \tag{6.32}$$

with the inequality holding simultaneously for $i \in \{1,2\}$ and each prefix $e_{0:u}$ with probability $1-\delta$, by choosing $\delta_{\mathsf{s}} = \delta/4(ARO)^{H-t}$. Choosing $\mu_0 \geq 4\sqrt{\log(2/\delta_{\mathsf{s}})/2|\mathcal{X}_i|}$ for each $i \in \{1,2\}$ yields

$$|\mathcal{X}_i| \geq \min(|\mathcal{X}_1|, |\mathcal{X}_2|) \geq \frac{8}{\mu_0^2}\log(2/\delta_{\mathsf{s}}) = \frac{8}{\mu_0^2}\Big(\log(2(ARO)^{H-t}) + \log(4/\delta)\Big) \tag{6.33}$$

In this case we have

$$L^{\mathsf{p}}_{\infty}(\mathcal{X}_1, \mathcal{X}_2) = \max_{u, e_{0:u}} |\widehat{p}_1(e) - \widehat{p}_2(e)| \geq L^{\mathsf{p}}_{\infty}(p_1, p_2) - \sqrt{\frac{\log(2/\delta_{\mathsf{s}})}{2|\mathcal{X}_1|}} - \sqrt{\frac{\log(2/\delta_{\mathsf{s}})}{2|\mathcal{X}_2|}}$$
$$\geq \mu_0 - \frac{\mu_0}{4} - \frac{\mu_0}{4} = \frac{\mu_0}{2} \geq 2\sqrt{\frac{\log(2/\delta_{\mathsf{s}})}{2\min(|\mathcal{X}_1|, |\mathcal{X}_2|)}} = \sqrt{\frac{2\log(8(ARO)^{H-t}/\delta)}{\min(|\mathcal{X}_1|, |\mathcal{X}_2|)}}$$

which is precisely the condition under which TestDistinct$(t, \mathcal{X}_1, \mathcal{X}_2, \delta)$ returns True. □

We are now ready to prove theorem 6.6, which we restate below:

Theorem 6.6. *Consider a dataset $\mathcal{D}$ of episodes sampled from an RDP $\mathbf{R}$ and a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$. With probability $1-\delta$, the output of* AdaCT–H$(\mathcal{D}, \delta/(2QAO))$ *is the transition function of the minimal RDP equivalent to $\mathbf{R}$, provided that $|\mathcal{D}| \geq N_\delta$, where*

$$N_\delta := \frac{21\log(8QAO/\delta)}{d^{\mathsf{b}}_{\min}\,\mu_0}\sqrt{H\log(2ARO)} \in \widetilde{O}\left(\frac{\sqrt{H}}{d^{\mathsf{b}}_{\min}\,\mu_0}\right) \tag{6.6}$$

$d^{\mathsf{b}}_{\min} := \min\{d^{\mathsf{b}}_t(q,ao) \mid t \in [H+1], q \in \mathcal{Q}_t, ao \in \mathcal{AO}, d^{\mathsf{b}}_t(q,ao) > 0\}$ *is the minimal occupancy distribution, and* μ_0 *is the* L^{p}_∞*-distinguishability.*

Proof. The proof consists in choosing N and δ such that the condition in lemma 6.14 is true with high probability, for each application of TestDistinct. Consider an iteration $t \in [H+1]$ of AdaCT–H. For a candidate state $qao \in \mathcal{Q}_{\mathsf{c},t+1}$, its associated probability is $d^{\mathsf{b}}_t(q,ao)$ with empirical estimate $\widehat{p}_t(qao) = |\mathcal{X}(qao)|/N$, i.e. the proportion of episodes in $\mathcal{D}$ that are consistent with qao. We can apply the empirical Bernstein inequality in eq. (6.10) to show that

$$\mathbb{P}\left(\left|\widehat{p}_t(qao) - d^{\mathsf{b}}_t(q,ao)\right| \geq \sqrt{\frac{2\widehat{p}_t(qao)\ell}{N}} + \frac{14\ell}{3N} = \frac{\sqrt{2M\ell} + 14\ell/3}{N}\right) \leq \delta \tag{6.34}$$

where $M = |\mathcal{X}(qao)|$, $\ell = \log(4/\delta)$, and δ is the failure probability of AdaCT–H. To obtain a bound on M and N, assume that we can estimate $d^{\mathsf{b}}_t(q,ao)$ with accuracy $d^{\mathsf{b}}_t(q,ao)/2$, which yields

$$\frac{d^{\mathsf{b}}_t(q,ao)}{2} \geq \frac{\sqrt{2M\ell} + 14\ell/3}{N} \tag{6.35}$$

$$\widehat{p}_t(qao) \geq d^{\mathsf{b}}_t(q,ao) - \frac{\sqrt{2M\ell} + 14\ell/3}{N} \geq d^{\mathsf{b}}_t(q,ao) - \frac{d^{\mathsf{b}}_t(q,ao)}{2} = \frac{d^{\mathsf{b}}_t(q,ao)}{2} \tag{6.36}$$

Combining these two results, we obtain

$$\begin{aligned} M &= N\widehat{p}_t(qao) \geq N d^{\mathsf{b}}_t(q,ao)/2 \\ &\geq \frac{N}{2N}\left(\sqrt{2M\ell} + 14\ell/3\right) = \frac{1}{2}\left(\sqrt{2M\ell} + 14\ell/3\right) \end{aligned} \tag{6.37}$$

Solving for M yields $M \geq 4\ell$, which is subsumed by the bound on M in lemma 6.14, since $\mu_0 < 1$. Hence, the bound on M in lemma 6.14 is sufficient to ensure that we estimate $d^{\mathsf{b}}_t(q,ao)$ with accuracy $d^{\mathsf{b}}_t(q,ao)/2$. We can now insert the bound on M from lemma 6.14 into (6.35) to obtain a bound on N:

$$N \geq \frac{2(\sqrt{2M\ell} + 14\ell/3)}{d^{\mathsf{b}}_t(q,ao)} \tag{6.38}$$

$$\geq \frac{2\ell}{d^{\mathsf{b}}_t(q,ao)}\left(\frac{4}{\mu_0}\sqrt{\frac{(H-t)\log(2ARO)}{\ell} + 1} + \frac{14}{3}\right) =: N_1 \tag{6.39}$$

To simplify the bound, we can choose any value larger than N_1:

$$\begin{aligned} N_1 &\leq \frac{2\ell}{d^{\mathsf{b}}_t(q,ao)}\left(\frac{4}{\mu_0}\sqrt{H\log(2ARO) + H\log(2ARO)} + \frac{14}{3\mu_0}\sqrt{H\log(2ARO)}\right) \\ &< \frac{21\ell}{d^{\mathsf{b}}_{\min}\,\mu_0}\sqrt{H\log(2ARO)} =: N_0 \end{aligned} \tag{6.40}$$

where we have used $d^{\mathsf{b}}_t(q, ao) \ge d^{\mathsf{b}}_{\min}$, $\mu_0 < 1$, $\ell = \log 4 + \log(1/\delta) \ge 1$, $H \log(2ARO) \ge \log 4 \ge 1$ and $4\sqrt{2} + 14/3 < \frac{21}{2}$. Choosing $\delta \coloneqq \delta_0/2QAO$, a union bound implies that accurately estimating $d^{\mathsf{b}}_t(q, ao)$ for each candidate state qao, and accurately estimating $p(e_{0:u}*)$ for each prefix in the multiset $\mathcal{X}(qao)$ associated with qao, occurs with probability $1 - 2QAO\delta = 1 - \delta_0$, since there are at most QAO candidate states. Substituting the expression for δ in N_0 yields the bound in the theorem.

It remains to show that the resulting RDP is minimal. We show the result by induction. The base case is given by the set $\mathcal{Q}_0$, which is clearly minimal since it only contains the initial state q_0. For $t \in [H + 1]$, assume that the algorithm has learned a minimal RDP for sets $\mathcal{Q}_0, \dots, \mathcal{Q}_t$. Let $\mathcal{Q}_{t+1}$ be the set of states at layer $t + 1$ of a minimal RDP. Due to proposition 6.1, each pair of histories that map to a state $q_{t+1} \in \mathcal{Q}_{t+1}$ generate the same probability distribution over suffixes. Hence, by lemma 6.13, with high probability, TestDistinct$(t, \mathcal{X}(qao), \mathcal{X}(q'a'o'), \delta)$ returns false, for each pair of candidate states qao and $q'a'o'$ that map to q_{t+1}. Consequently, the algorithm merges qao and $q'a'o'$. On the other hand, by assumption, each pair of histories that map to different states of $\mathcal{Q}_{t+1}$ have L^{p}_∞-distinguishability μ_0. Hence, by lemma 6.14, with high probability, TestDistinct$(t, \mathcal{X}(qao), \mathcal{X}(q'a'o'), \delta)$ returns true, for each pair of candidate states qao and $q'a'o'$ that map to different states in $\mathcal{Q}_{t+1}$. Consequently, the algorithm does not merge qao and $q'a'o'$. It follows that with high probability, AdaCT–H will generate exactly the set $\mathcal{Q}_{t+1}$, which is that of a minimal RDP. □

6.8.4 Sample Complexity of `AdaCT-H-A`

In this section we prove theorem 6.8, which states an alternative upper bound on the sample complexity of AdaCT–H. The proof requires an alternative definition of the algorithm, which we call AdaCT–H–A, with A for "approximation".

Theorem 6.8. *Consider a dataset $\mathcal{D}$ of episodes sampled from an RDP $\mathbf{R}$ and a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$. With probability $1 - \delta$, the output of* AdaCT–H–A, *called with $\mathcal{D}$, $\delta/(2QAO)$ and $\varepsilon \in (0, H]$ in input, is the transition function of an $\varepsilon/2$-approximate RDP $\mathbf{R}'$, provided that $|\mathcal{D}| \ge N'_\delta$, where*

$$N'_\delta \coloneqq \frac{504 H QAO C^*_{\mathbf{R}'} \log(16QAO/\delta)}{\varepsilon\, \mu_0} \sqrt{H \log(2ARO)} \in \widetilde{O}\left(\frac{H^{3/2} QAO C^*_{\mathbf{R}'}}{\varepsilon\, \mu_0}\right)$$

Proof. AdaCT–H–A returns the set of RDP states $\mathcal{Q}'$ and transition function τ' of an approximate RDP $\mathbf{R}'$, taking as input the accuracy ε, an upper bound $\overline{Q}$ on $|\mathcal{Q}'|$, and an upper bound $\overline{C}$ on the concentrability $C^*_{\mathbf{R}'}$ of $\mathbf{R}'$. As a side note, for the relation between $C^*_{\mathbf{R}'}$ and $C^*_{\mathbf{R}}$, the reader can refer to lemma 6.15.

Function AdaCT–H–A($\mathcal{D}$, δ, ε, $\overline{Q}$, $\overline{C}$)

Input: Dataset $\mathcal{D}$, failure probability $0 < \delta < 1$, accuracy ε, upper bounds $\overline{Q}$ on $|\mathcal{Q}'|$ and $\overline{C}$ on $C^*_{\mathbf{R}'}$

Output: Set of states $\mathcal{Q}'$ and transition function $\tau' : \mathcal{Q}' \times \mathcal{AO} \to \mathcal{Q}'$ of an approximate RDP $\mathbf{R}'$

```
$\mathcal{Q}'_0 \leftarrow \{q_0\}$, $\mathcal{X}(q_0) \leftarrow \mathcal{D}$                                   // initial state
$\mathcal{Q}'_0 \leftarrow \mathcal{Q}'_0 \cup \{q^{\mathsf{e}}_0\}$, $\mathcal{X}(q^{\mathsf{e}}_0) \leftarrow \emptyset$                          // initial side state
for $t = 0, \ldots, H$ do
    $\mathcal{Q}'_{t+1} \leftarrow \{q^{\mathsf{e}}_{t+1}\}$                                          // make side state
    foreach $ao \in \mathcal{AO}$ do
        $\tau'(q^{\mathsf{e}}_t, ao) = q^{\mathsf{e}}_{t+1}$
        $\mathcal{X}(q^{\mathsf{e}}_{t+1}) \leftarrow \{e_{t+1:H} \mid aroe_{t+1:H} \in \mathcal{X}(q^{\mathsf{e}}_t)\}$
    end
    $\mathcal{Q}'_{\mathsf{c},t+1} \leftarrow \{qao \mid q \in \mathcal{Q}'_t, ao \in \mathcal{AO}\}$                  // make candidate states
    foreach $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ do
        $\mathcal{X}(qao) \leftarrow \{e_{t+1:H} \mid aroe_{t+1:H} \in \mathcal{X}(q)\}$                 // compute suffixes
    end
    $q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}} \leftarrow \arg\max_{qao \in \mathcal{Q}'_{\mathsf{c},t+1}} |\mathcal{X}(qao)|$                  // most common candidate
    $\mathcal{Q}'_{t+1} \leftarrow \mathcal{Q}'_{t+1} \cup \{q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}\}$, $\tau'(q_{\mathsf{m}}, a_{\mathsf{m}}o_{\mathsf{m}}) = q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}$        // promote candidate
    $\mathcal{Q}'_{\mathsf{c},t+1} \leftarrow \mathcal{Q}'_{\mathsf{c},t+1} \setminus \{q_{\mathsf{m}}a_{\mathsf{m}}o_{\mathsf{m}}\}$
    foreach $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N \geq \varepsilon/(4\overline{Q}AOH\overline{C})$ do
        $Similar \leftarrow \{q' \in \mathcal{Q}'_{t+1} \mid$ not TestDistinct$(t, \mathcal{X}(qao), \mathcal{X}(q'), \delta)\}$
        if $Similar = \emptyset$ then
            $\mathcal{Q}'_{t+1} \leftarrow \mathcal{Q}'_{t+1} \cup \{qao\}$, $\tau'(q, ao) = qao$           // promote candidate
        end
        else
            $q' \leftarrow$ element in $Similar$
            $\tau'(q, ao) = q'$, $\mathcal{X}(q') \leftarrow \mathcal{X}(q') \cup \mathcal{X}(qao)$              // merge states
        end
        if $|\mathcal{Q}'_0| + \cdots + |\mathcal{Q}'_{t+1}| > \overline{Q}$ then
            return Failure
        end
    end
    foreach $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N < \varepsilon/(4\overline{Q}AOH\overline{C})$ do
        $\tau'(q, ao) = q^{\mathsf{e}}_{t+1}$, $\mathcal{X}(q^{\mathsf{e}}_{t+1}) \leftarrow \mathcal{X}(q^{\mathsf{e}}_{t+1}) \cup \mathcal{X}(qao)$    // merge with side state
    end
end
return $\mathcal{Q}'_0 \cup \cdots \cup \mathcal{Q}'_{H+1}$, $\tau'$
```

If, at any moment, the number of RDP states $|\mathcal{Q}'|$ exceeds $\overline{Q}$, the algorithm returns Failure (line 26). AdaCT–H–A defines a sequence of side states $q_0^{\mathsf{e}}, \ldots, q_{H+1}^{\mathsf{e}}$ (lines 2 and 4), and defines $\tau'(q_t^{\mathsf{e}}, ao) = q_{t+1}^{\mathsf{e}}$ for each $t \in [H+1]$ and $ao \in \mathcal{AO}$ (line 6). For each candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N \geq \varepsilon/(4\overline{Q}AOH\overline{C})$, the definition of AdaCT–H–A is the same as that of AdaCT–H, including the call to TestDistinct (the lines in the block at 16). For each candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N < \varepsilon/(4\overline{Q}AOH\overline{C})$, instead of mapping (q, ao) to the correct RDP state, AdaCT–H–A maps (q, ao) to the side state q_{t+1}^{e} (line 30). Once in q_{t+1}^{e}, $\mathbf{R}'$ remains in a side state for the rest of the episode. We observe that the side states do not satisfy proposition 6.1, since the histories that map to side states may assign different probabilities to suffixes (and TestDistinct is never called).

We define an alternative occupancy measure $d'_t(q, ao)$ associated with the approximate RDP $\mathbf{R}'$ and the behaviour policy π^{b}. The new definition is given by $d'_0(q_0, a_0o_0) = \theta_{\mathsf{o}}(o_0 \mid q_0, a_0)$ and

$$d'_t(q_t, a_to_t) = \sum_{(q,ao)\in\tau'^{-1}(q_t)} d'_{t-1}(q, ao)\pi^{\mathsf{b}}(a_t \mid q_t)\theta_{\mathsf{o}}(o_t \mid q_t, a_t) \tag{6.41}$$

The only difference between d'_t and d^{b}_t is that d'_t is defined with respect to the transition function τ' of the approximate RDP $\mathbf{R}'$, instead of the transition function τ associated with the original RDP $\mathbf{R}$. Note that apart from the side states, $\mathbf{R}'$ will contain the same states as $\mathbf{R}$, as long as the candidate states satisfy the condition on line 16, and τ' will be the same as τ on those states. Because of this relationship, L^{p}_{∞}-distingishability μ_0 of $\mathbf{R}'$ is at least as that of $\mathbf{R}$.

First consider each candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N \geq \varepsilon/(4\overline{Q}AOH\overline{C})$. In this case, AdaCT–H–A calls TestDistinct, so lemmas 6.13 and 6.14 apply to these candidate states. The associated occupancy is $d'_t(q, ao)$ with empirical estimate $\widehat{p}_t(qao) = |\mathcal{X}(qao)|/N$. Hence, the empirical Bernstein inequality applies to $d'_t(q, ao)$ and $\widehat{p}_t(qao)$. Just as in the proof of theorem 6.6, we choose $\mathcal{X}(qao)$ large enough to accurately estimate $d'_t(q, ao)$ within a factor $d'_t(q, ao)/2$ with probability $1-\delta$. Thus, we obtain an alternative upper bound on $d'_t(q, ao)$ as follows:

$$d'_t(q, ao) \geq \frac{|\mathcal{X}(qao)|}{N} - \frac{d'_t(q, ao)}{2} \quad \Leftrightarrow \quad \frac{3d'_t(q, ao)}{2} \geq \frac{|\mathcal{X}(qao)|}{N} \geq \frac{\varepsilon}{4\overline{Q}AOH\overline{C}} \tag{6.42}$$

From here, we can use the proof of theorem 6.6 by substituting d'_t for d^{b}_t, up until the definition of the bound N_1 on $|\mathcal{D}|$ in (6.39). Inserting the bound on $d'_t(q, ao)$ into the expression for N_1 yields

$$N_1 \leq \frac{2\ell}{d'_t(q, ao)} \left(\frac{4}{\mu_0}\sqrt{H\log(2ARO) + H\log(2ARO)} + \frac{14}{3\mu_0}\sqrt{H\log(2ARO)} \right)$$

$$\leq \frac{126\overline{Q}AOH\overline{C}\ell}{\varepsilon\mu_0}\sqrt{H\log(2ARO)} =: N_2 \tag{6.43}$$

Next, consider each candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N < \varepsilon/(4\overline{Q}AOH\overline{C})$. In this case, we instead choose $\mathcal{X}(qao)$ large enough to estimate $d'_t(q,ao)$ with accuracy β with probability $1-\delta$. From the empirical Bernstein inequality, estimating $d'_t(q,ao)$ with accuracy β implies

$$\beta \geq \sqrt{\frac{2\widehat{p}_t(qao)\ell}{N}} + \frac{14\ell}{3N} \quad\Leftrightarrow\quad N \geq \frac{2\ell}{\beta}\left(\frac{14}{3} + \frac{\widehat{p}_t(qao)}{\beta}\right) =: N_3 \tag{6.44}$$

Choosing $\beta = \varepsilon/(4\overline{Q}AOH\overline{C})$ implies $\widehat{p}_t(qao) < \beta$, and we can thus simplify N_3 as

$$N_3 = \frac{2\ell}{\beta}\left(\frac{14}{3} + \frac{\widehat{p}_t(qao)}{\beta}\right) < \frac{12\ell}{\beta} = \frac{48\overline{Q}AOH\overline{C}\ell}{\varepsilon} =: N_4 \tag{6.45}$$

In addition, this choice of β yields the following bound on $d'_t(q,ao)$:

$$d'_t(q,ao) \leq \widehat{p}_t(qao) + \beta < \frac{\varepsilon}{4\overline{Q}AOH\overline{C}} + \frac{\varepsilon}{4\overline{Q}AOH\overline{C}} = \frac{\varepsilon}{2\overline{Q}AOH\overline{C}} \tag{6.46}$$

To conclude the proof, we verify that $\mathbf{R}'$ is an $\varepsilon/2$-approximation of the original RDP $\mathbf{R}$. We briefly overload notation by letting $d^*_t(q,ao)$ refer to the occupancy of an optimal policy π^* *with respect to the transition function* τ' *of* $\mathbf{R}'$. Consider a candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$ such that $|\mathcal{X}(qao)|/N < \varepsilon/(4\overline{Q}AOH\overline{C})$. The contribution to the expected optimal reward of $\mathbf{R}$ of all histories that map to qao is bounded as

$$d^*_t(q,ao)(H-t) \leq C^*_{\mathbf{R}'}\, d'_t(q,ao)H < \frac{\varepsilon}{2\overline{Q}AO} \tag{6.47}$$

since $(H-t)$ is the maximum reward obtained during the remaining time steps. Since qao is mapped to a side state of $\mathbf{R}'$, an optimal policy for $\mathbf{R}'$ may not accurately estimate the expected optimal value for qao, but the contribution of all such candidate states to the expected optimal value is at most

$$\sum_{t\in[H]}\sum_{q\in\mathcal{Q}_t}\sum_{ao\in\mathcal{AO}} d^*_t(q,ao)(H-t) \leq \sum_{t\in[H]}\sum_{q\in\mathcal{Q}_t}\sum_{ao\in\mathcal{AO}} \frac{\varepsilon}{2\overline{Q}AO} \leq \frac{\varepsilon}{2}, \tag{6.48}$$

since there can be at most $\overline{Q}AO$ such candidate states. Hence, any optimal policy for $\mathbf{R}'$ is an $\varepsilon/2$-optimal policy for $\mathbf{R}$, which implies that we can approximate an ε-optimal regular policy for the exact RDP $\mathbf{R}$ by finding an $\varepsilon/2$-optimal policy for the approximate RDP $\mathbf{R}'$.

It is easy to verify that the bound N_4 in (6.45) is less than the bound N_2 in (6.43). Hence, a worst-case bound is obtained by assuming that $|\mathcal{X}(qao)|/N \geq \varepsilon/(4\overline{Q}AOH\overline{C})$ for each $t \in [H+1]$ and each candidate state $qao \in \mathcal{Q}'_{\mathsf{c},t+1}$, which yields an upper bound N_2. Note that AdaCT–H–A takes as input an upper bound $\overline{Q}$ on the number of RDP states $|\mathcal{Q}'|$ of $\mathbf{R}'$, as well as an upper bound $\overline{C}$ of the concentrability coefficient $C^*_{\mathbf{R}'}$. If the learning agent has no prior knowledge of $\overline{Q}$ and $\overline{C}$, it could start with small estimates of $\overline{Q}$ and $\overline{C}$, and in the case that AdaCT–H–A returns Failure, or the resulting policy has larger concentrability than $\overline{C}$ for $\mathbf{R}'$, it could iteratively double the estimates $\overline{Q}$ and/or $\overline{C}$ and call the algorithm again. This only increases the computational complexity of AdaCT–H–A by a factor $O(\log QC^*_{\mathbf{R}'})$, and the resulting upper bounds $\overline{Q}$ and $\overline{C}$ do not exceed $2Q$ and $2C^*_{\mathbf{R}'}$. Since we already have an estimate $\overline{Q}$, in each iteration we can call AdaCT–H–A with $\delta = \delta_1/(2\overline{Q}AO)$ to ensure that the bound N_2 holds for each candidate state simultaneously with probability $1 - \delta_1$. Substituting this value of δ in the bound N_2 in (6.43) and using $\overline{Q} < 2Q$ and $\overline{C} < 2C^*_{\mathbf{R}'}$ yields the sample complexity bound stated in the theorem. □

Lemma 6.15. *The concentrability $C^*_{\mathbf{R}'}$ of the approximate RDP $\mathbf{R}'$ from theorem 6.8 satisfies*

$$C^*_{\mathbf{R}'} \leq C^*_{\mathbf{R}}(1 + 3\overline{Q}AO) \tag{6.49}$$

Proof. In this proof, we use the same conventions as the proof of theorem 6.8. For each $t > 0$, let $d'_t(q^{\mathsf{e}}_t)$ be the occupancy of the side state q^{e}_t in the approximate RDP $\mathbf{R}'$. We prove by induction on t that $d'_t(q^{\mathsf{e}}_t)$ satisfies

$$d'_t(q^{\mathsf{e}}_t) < \frac{\varepsilon \sum_{u=0}^{t-1} |\mathcal{Q}_u|}{2\overline{Q}H\overline{C}} \leq \frac{\varepsilon}{2H\overline{C}} \tag{6.50}$$

The base case is given by $t = 1$. In this case, a candidate state (q_0, ao) is mapped to q^{e}_1 if $d^{\mathsf{b}}_t(q_0, ao) = d'_t(q_0, ao) < \varepsilon/(2\overline{Q}AOH\overline{C})$. Since there can be at most $AO = |\mathcal{Q}_0|AO$ such candidate states, we have

$$d'_t(q^{\mathsf{e}}_t) < \frac{\varepsilon|\mathcal{Q}_0|AO}{2\overline{Q}AOH\overline{C}} = \frac{\varepsilon|\mathcal{Q}_0|}{2\overline{Q}H\overline{C}} \tag{6.51}$$

For $t > 1$, a candidate state (q_{t-1}, ao) is mapped to q^{e}_t if $d'_t(q_{t-1}, ao) < \varepsilon/(2\overline{Q}AOH\overline{C})$. Again, there can be at most $|Q_{t-1}|AO$ such candidate states. Since all occupancy of q^{e}_{t-1} is also mapped to q^{e}_t, we have

$$d'_t(q^{\mathsf{e}}_t) < d'_{t-1}(q^{\mathsf{e}}_{t-1}) + \frac{\varepsilon|\mathcal{Q}_{t-1}|AO}{2\overline{Q}AOH\overline{C}} < \frac{\varepsilon \sum_{u=0}^{t-2} |\mathcal{Q}_u|}{2\overline{Q}H\overline{C}} + \frac{\varepsilon|\mathcal{Q}_{t-1}|}{2\overline{Q}H\overline{C}} = \frac{\varepsilon \sum_{u=0}^{t-1} |\mathcal{Q}_u|}{2\overline{Q}H\overline{C}} \tag{6.52}$$

where we have used the induction hypothesis.

Consider a candidate state (q, ao) of $\mathbf{R}$ at time t. Due to approximation, some histories in $\bar{\tau}^{-1}(q)$ are mapped to side states in $\mathbf{R}'$ instead of q, and we can therefore write $d^{\mathsf{b}}_t(q, ao) = d'_t(q, ao) + \xi \leq d'_t(q, ao) + d'_t(q^{\mathsf{e}}_t)$, where ξ is the total occupancy of histories in $\bar{\tau}^{-1}(q)$ mapped to side states. In turn, this implies

$$d^*_t(q, ao) \le d^{\mathsf{b}}_t(q, ao) C^*_{\mathbf{R}} \le (d'_t(q, ao) + d'_t(q^{\mathsf{e}}_t)) C^*_{\mathbf{R}} < \left(d'_t(q, ao) + \frac{\varepsilon}{2H\overline{C}} \right) C^*_{\mathbf{R}} \quad (6.53)$$

The concentrability of a candidate state (q, ao) in the approximate RDP $\mathbf{R}'$ that is not mapped to a side state (i.e. $d'_t(q, ao) \ge \varepsilon/(6\overline{Q}AOH\overline{C})$) can now be bounded as

$$\begin{aligned} \frac{d^*_t(q, ao)}{d'_t(q, ao)} &< \frac{d'_t(q, ao) + \varepsilon/(2H\overline{C})}{d'_t(q, ao)} C^*_{\mathbf{R}} = \left(1 + \frac{\varepsilon}{2H\overline{C} d'_t(q, ao)} \right) C^*_{\mathbf{R}} \\ &\le C^*_{\mathbf{R}} (1 + 3\overline{Q}AO) \end{aligned} \quad (6.54)$$

This concludes the proof of the lemma. □

6.8.5 Distinguishability Parameters

As defined in the main text, for $t \in [H+1]$, we consider a metric L over distributions over the remaining part of the episode $\Delta(\mathcal{T}_\ell)$, for $\ell = H - t + 1$. Then, the L-distinguishability of an RDP $\mathbf{R}$ and a policy π is the maximum μ_0 such that, for any $t \in [H+1]$ and any two distinct $q, q' \in \mathcal{Q}_t$, the probability distributions over suffix traces $e_{t:H} \in \mathcal{T}_\ell$ from the two states satisfy

$$L(\mathbb{P}(e_{t:H} \mid q_t = q, \pi), \mathbb{P}(e_{t:H} \mid q_t = q', \pi)) \ge \mu_0 \quad (6.55)$$

So, μ_0 is a feature of the RDP and the policy combined, and it quantifies the distance between any two distinct states of the RDP with respect to the distributions they induce over the observable quantities. Distinguishability parameters have been first introduced in Ron, Singer, et al. (1998), later generalized for other metrics. They can be also found in Balle Pigem (2013), for PDFA learning, and in Ronca and De Giacomo (2021) and Ronca, Licks, et al. (2022), for RDP learning.

According to the definition we adopt, there exists an L-distinguishability for any RDP and policy. However, as stated in assumption 6.2, we require μ_0 to be strictly positive. This does not constitute a restriction for the RDP, since it can be always minimized while preserving all conditional probabilities. Though it implies that, in any state, the behaviour policy takes with positive probability all actions that are needed to observe episode suffixes that have different probability under the two states. Clearly if this was not the case for two distinct $q, q' \in \mathcal{Q}_t$ at some $t \in [H+1]$, $\mathbb{P}(e_{t:H} \mid q_t = q, \pi) = \mathbb{P}(e_{t:H} \mid q_t = q', \pi)$ and no information would be available for the algorithm to distinguish q and q'.

The metric selected also influences the actual value of the distinguishability parameter. In this work, we adopt L^{p}_∞, as it can be seen from the TestDistinct function in the two algorithms. A more standard distance would be L_∞. According to eq. (6.55), an L_∞-distinguishability of μ_0 implies that for any $t \in [H+1]$ and two distinct $q, q' \in \mathcal{Q}_t$,

$$\max_{e \in \mathcal{T}_{H-t+1}} |\mathbb{P}(e_{t:H} = e \mid q_t = q) - \mathbb{P}(e_{t:H} = e \mid q_t = q')| \geq \mu_0 \tag{6.56}$$

This means that some sequence until the end of the episode has a different probability of being generated from the two states. Although similar, the L_∞^{p} distance, maximizes for the full trace, as in the previous expression, as well as any of its prefixes:

$$\max_{u \in [H-t+1]} \max_{e \in \mathcal{T}_{u+1}} |\mathbb{P}(e_{t:H} = e* \mid q_t = q) - \mathbb{P}(e_{t:H} = e* \mid q_t = q')| \geq \mu_0 \tag{6.57}$$

As it has been discussed in Balle Pigem (2013, Appenix A.5), the prefix L_∞^{p} metric always upper bounds the L_∞ metric, up to a multiplicative factor, and there are pairs of distributions in which L_∞ is exponentially smaller than L_∞^{p} with respect to the expected suffix length. This motivates our choice. Moreover, in the specific case of our fixed horizon setting, we have that the L_∞^{p}-distinguishability is never lower than L_∞-distinguishability. Note that in the hard instance of fig. 6.1, the two coincide.

The lower bound is stated in terms of the L_1^{p}-distinguishability of the RDP, instead. While L_∞^{p} is achieved for one specific trace prefix, maximizing the difference in probability, L_1^{p} takes all traces prefixes for each length into account as

$$\max_{u \in [H-t+1]} \sum_{e \in \mathcal{T}_{u+1}} |\mathbb{P}(e_{t:H} = e* \mid q_t = q) - \mathbb{P}(e_{t:H} = e* \mid q_t = q')| \tag{6.58}$$

Due to this relation, the L_∞^{p}-distinguishability always lower bounds the L_1^{p}-distinguishability in the fixed horizon setting. Note that the definition of L_1^{p} used in this thesis differs from the one used in the paper this chapter is based from. In fact, in (Cipollone, Jonsson, et al. 2024), L_1^{p} was defined as:

$$\sum_{u \in [H-t+1]} \sum_{e \in \mathcal{T}_{u+1}} |\mathbb{P}(e_{t:H} = e* \mid q_t = q) - \mathbb{P}(e_{t:H} = e* \mid q_t = q')| \tag{6.59}$$

The motivation for this change is that with the new definition of L_1^{p} we were able to achieve here a much stronger lower bound, since the associated parameter appears in the denominator of eq. (6.8).

6.8.6 `RegORL` With Subsampled `VI-LCB`

In this section, we demonstrate the composition of our algorithm with a specific Offline Reinforcement Learning algorithm for MDPs. Specifically, we adopt Subsampled `VI-LCB`, from of G. Li, L. Shi, et al. (2022, Algorithm 3) and report the combined sample complexity of this choice, through a simple application of theorem 6.7.

First, we introduce the occupancy distribution and the single-policy conentrability coefficient for MDPs. Let $\mathbf{M} = \langle \mathcal{Q}, \mathcal{A}, \mathcal{R}, T, R, H \rangle$ be a finite-horizon MDP with states $\mathcal{Q}$, horizon H, transition function $T : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{Q})$ and reward function $R : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{R})$. The state-action occupancy distribution of a policy $\pi : \mathcal{Q} \to \Delta(\mathcal{A})$ in $\mathbf{M}$ at step $t \in [H+1]$ is $d^{\pi}_{\mathsf{m},t}(q, a) = \mathbb{P}(q_t = q, a_{t+1} = a \mid \mathbf{M}, \pi)$. For our purposes, it suffices to consider a fixed initial state q_0. Finally, the MDP single-policy concentrability of a behaviour policy π^{b} is (Rashidinejad, Zhu, et al. 2021):

$$C^* = \max_{t \in [H+1], q \in \mathcal{Q}, a \in \mathcal{A}} \frac{d^{\pi^*}_{\mathsf{m},t}(q, a)}{d^{\pi^{\mathsf{b}}}_{\mathsf{m},t}(q, a)} \tag{6.60}$$

We can now express the sample complexity of Subsampled `VI-LCB`.

Theorem 6.16 (G. Li, L. Shi, et al. (2022))**.** *Let $\mathcal{D}$ be a dataset of N_{m} episodes, sampled from an MDP $\mathbf{M}$ with a Markovian policy π^{b}. For any $\varepsilon \in (0, H]$ and $0 < \delta < 1/12$, with probability exceeding $1 - \delta$, the policy $\widehat{\pi}$ returned by Subsampled `VI-LCB` obeys $V^*_{\mu} - V^{\widehat{\pi}}_{\mu} \leq \varepsilon$, as long as:*

$$N_{\mathsf{m}} \geq \frac{c\, H^3 Q C^* \log \frac{N_{\mathsf{m}} H}{\delta}}{\varepsilon^2} \tag{6.61}$$

for a fixed, positive constant c.

The analysis in G. Li, L. Shi, et al. (2022) of Subsampled `VI-LCB` assumes that the reward function is deterministic and known. Thus, restricting our attention to this setting, we consider any episodic RDP with history-dependent, deterministic rewards. The reward function can be regarded as known, since it may be easily extracted from the dataset resulting from the Markov transformation of definition 6.2.

Proposition 6.17. *Let $\mathcal{D}$ be a dataset of N episodes, sampled with a regular policy $\pi^{\mathsf{b}} \in \Pi_{\mathbf{R}}$ from an RDP $\mathbf{R}$ with deterministic rewards. If Subsampled `VI-LCB` is the* OfflineRL *algorithm in algorithm 6.1, then, for any $\varepsilon \in (0, H]$ and $0 < \delta < 1/12$, with probability exceeding $1 - \delta$, the output of `RegORL`$(\mathcal{D}, \varepsilon, \delta)$ is an ε-optimal policy of $\mathbf{R}$, as long as*

$$N \geq 2 \max \left\{ \frac{21 \log(8QAO/\delta)}{d^{\mathsf{b}}_{\min}\, \mu_0} \sqrt{H \log(2ARO)},\ \frac{c\, H^3 Q C^*_{\mathbf{R}} \log \frac{2NH}{\delta}}{\varepsilon^2} \right\} \tag{6.62}$$

Proof. This statement follows as a direct application of theorem 6.16 to theorem 6.6. It only remains to verify that the single-policy concentrability of the MDP underlying the dataset $\mathcal{D}'$ that Subsampled `VI-LCB` receives is $C^*_{\mathbf{R}}$. The dataset $\mathcal{D}'$ is generated according to the Markov transformation $\bar{\tau}$ from definition 6.2. We only consider the cases in which AdaCT–H succeeds. Let $\pi \in \Pi_{\mathbf{R}}$ be any regular policy and q_t, q'_t the states reached at step t by $\mathbf{R}$ and $\mathbf{M_R}$, respectively. Then for $t > 0$,

$$d_t^\pi(q,a) \coloneqq \mathbb{P}(q_t = q \mid \mathbf{R}, \pi)\, \pi_{\mathsf{r}}(a \mid q) \tag{6.63}$$

$$= \mathbb{P}(\bar{\tau}(h_{t-1}) = q \mid \mathbf{R}, \pi)\, \pi_{\mathsf{r}}(a \mid q) \tag{6.64}$$

$$= \mathbb{P}(q'_t = q \mid \mathbf{M_R}, \pi_{\mathsf{r}})\, \pi_{\mathsf{r}}(a \mid q) \tag{6.65}$$

$$= d_{\mathsf{m},t}^{\pi_{\mathsf{r}}}(q,a) \tag{6.66}$$

This is valid for any regular policy, and for the optimal and behaviour policies in particular. Then,

$$C^*_{\mathbf{R}} = \max_{t \in [H+1], q \in \mathcal{Q}_t, ao \in \mathcal{AO}} \frac{d_t^*(q,ao)}{d_t^{\mathsf{b}}(q,ao)} \tag{6.67}$$

$$= \max_{t \in [H+1], q \in \mathcal{Q}_t, ao \in \mathcal{AO}} \frac{d_t^{\pi^*}(q,a)\, \theta_{\mathsf{o}}(o \mid q,a)}{d_t^{\pi^{\mathsf{b}}}(q,a)\, \theta_{\mathsf{o}}(o \mid q,a)} \tag{6.68}$$

$$= \max_{t \in [H+1], q \in \mathcal{Q}, a \in \mathcal{A}} \frac{d_{\mathsf{m},t}^{\pi_{\mathsf{r}}^*}(q,a)}{d_{\mathsf{m},t}^{\pi_{\mathsf{r}}^{\mathsf{b}}}(q,a)} \tag{6.69}$$

$$= C^* \tag{6.70}$$

□

Similarly to the previous proposition, it is also possible to combine theorem 6.16 with theorem 6.8. In this case, the sample complexity of Subsampled `VI-LCB` for learning an $\varepsilon/2$-accurate policy with probability $1 - \delta/2$ would be combined with $N'_{\delta/2}$ of theorem 6.8.

6.8.7 Sample Complexity Lower Bound

In this section, we prove the sample complexity lower bound of theorem 6.9. The proof is based on a suitable composition of a two-armed bandit and a learning problem associated to noisy parity functions. We first describe this latter class of problems and its sample-complexity lower bound below. Then, we compose a hard class of RDP instances at page 164, and prove the final statement at page 165.

Learning Parity With Noise

Let $\mathbb{B} \coloneqq \{0,1\}$ and $L \in \mathbb{N}$. For any string $x \in \mathbb{B}^L$, the parity function $f_x : \mathbb{B}^L \to \mathbb{B}$ is $f_x(y) = \oplus_{i \in [L]} x_i y_i$, where $\oplus$ is addition modulo 2. For noise parameter $\xi \in (0, 0.5)$, a noisy parity function $f_{x,\xi}$ returns $f_x(y)$ with probability $0.5 + \xi$ and $1 - f_x(y)$ otherwise. Consider the class of parity functions $\mathbb{F}(L) = \{f_x\}_{x \in \mathbb{B}^L}$ and the class of noisy parity functions $\mathbb{F}(L,\xi) = \{f_{x,\xi}\}_{x \in \mathbb{B}^L}$. Assume that x, y_1, y_2, ...$\sim$ Unif$(\mathbb{B}^L)$ are uniformly sampled. The success probability of a streaming algorithm $\mathfrak{A}$ for $\mathbb{F}(L,\xi)$

is the probability that $\mathfrak{A}$ recovers x, the hidden code, given in input a sequence of observations $(y_i, f_{x,\xi}(y_i))_i$.

Lemma 6.18. *Any streaming algorithm for $\mathbb{F}(L, \xi)$ with a success probability higher than $O(2^{-L})$ requires at least $\Omega(L/\xi)$ or $2^{\Omega(L)}$ input samples $(y_i, f_{x,\xi}(y_i))_i$.*

Proof. Learning in $\mathbb{F}(L, \xi)$ is the problem of recovering $x \in 2^{\mathbb{B}}$ from noisy data (y_i, b_i), where $b_i = f_x(y_i)$ with probability $0.5 + \xi$, and $b_i = 1 - f_x(y_i)$ otherwise. This is the problem of learning in $\mathbb{F}(L)$ with corruption rate $0.5 - \xi$. Hence, we focus on the problem of learning noiseless parity first.

The Statistical Query dimension $\text{SQ}\textsc{dim}(\mathcal{C}, d)$, characterizes the complexity of learning in the class $\mathcal{C}$ with respect to the prior distribution $d \in \Delta(\mathcal{C})$. As defined in Szörényi (2009), $\text{SQ}\textsc{dim}(\mathcal{C}, d)$ is the maximum $n \in \mathbb{N}$ such that there exist distinct $f_1, \ldots, f_n \in \mathcal{C}$, such that their pairwise correlations with respect to d are between $-1/n$ and $1/n$. For the class of parity functions, under the uniform distribution over x, $\text{SQ}\textsc{dim}(\mathbb{F}(L), \text{Unif}) = 2^L$. This was already observed in (Blum, Furst, et al. 1994), for a slightly different notion of SQ dimension. However, to verify this, we can consider a natural ordering over binary strings in $\mathcal{X}$, and represent the problem of learning $\mathbb{F}(L)$ as a matrix $M = (m_{ij}) \in \{1, -1\}^{2^L \times 2^L}$, defined as $m_{ij} = (-1)^{f_{x_j}(y_i)} = (-1)^{y_i \cdot x_j}$, where scalar product is modulo 2. We have that M is a Hadamard matrix. Then, since every row is orthogonal to the others, and the same is true for columns, every couple of parity functions are uncorrelated under the uniform distribution over x.

Regarding the noisy parity problem, since $\text{SQ}\textsc{dim}(\mathbb{F}(L), \text{Unif}) = 2^L$, we can apply Garg, Raz, et al. (2018, corollary 8) with $m = 2^L$, to have that the matrix M corresponding to the parity problem is a (k, l)-L_2-extractor with error 2^{-r}, for $k, l, r \in \Omega(L)$. Since M is a suitable extractor, we can apply Garg, Kothari, et al. (2021, theorem 1), which considers the problem of learning the extractor matrix M with the additional noise parameter ξ. We obtain that, in the streaming setting, any branching program B for $\mathbb{F}(L, \xi)$ whose depth is at most $2^{f_1(k,l,r)}$ and width is at most $2^{ckl/\xi}$ has a success probability of at most $O(2^{-f_1(k,l,r)})$, where c is a suitable constant and f_1 is from Garg, Kothari, et al. (2021, equation (1)).

Then, if the success probability of the program $\mathfrak{A}$ is not in $O(2^{-f_1(k,l,r)})$, meaning it is higher, we have that the depth of $\mathfrak{A}$ exceeds $2^{f_1(k,l,r)}$ or the width of $\mathfrak{A}$ exceeds $2^{ckl/\xi}$. Expanding f_1, since $k, l, r \in \Omega(L)$, if the success probability is not in $O(2^{-L})$, then the depth of $\mathfrak{A}$ is $2^{\Omega(L)}$ or the width of $\mathfrak{A}$ is $2^{\Omega(L^2)/\xi}$. Width and depth refer to the computational model that represents $\mathfrak{A}$ as a branching program. A branching program is a directed acyclic graph in which internal nodes have one outgoing edge for each possible input sample, that is $|\mathbb{B}^L \times \mathbb{B}| = 2^{L+1}$ in our problem, and leaves correspond to algorithm decisions. From the required width and depth we know that

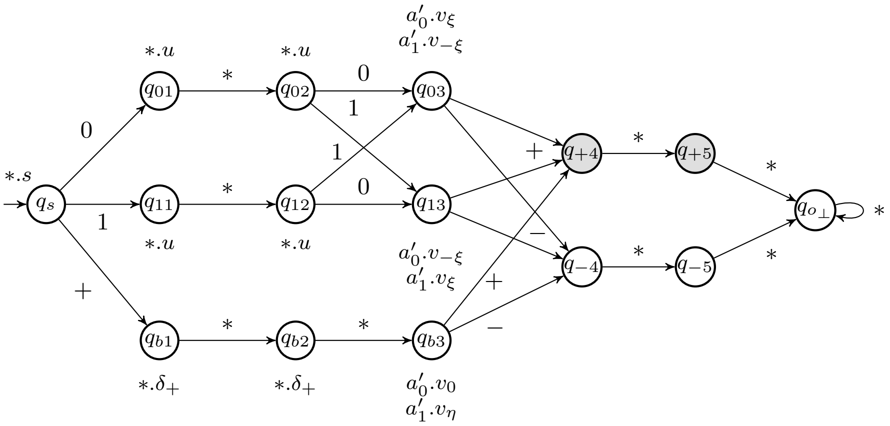


Figure 6.2. One episodic RDP instance $\mathbf{R}_{101,1} \in \mathbb{R}(L, H, \xi, \eta)$, associated to the parity function f_{101}, with code 101, and the optimal arm a_1'. The length is $L = |101| = 3$, the horizon $H = 5$, the noise parameter is $\xi > 0$ and the bandit bonus parameter is $\eta > 0$. The transition function only depends on the observations, not the actions. The star denotes any symbol. If the label of a state q is $a.d$, then the observation function is $\theta_\mathsf{o}(a \mid q) = d$, where $d \in \Delta(\mathcal{O})$ (some irrelevant outputs are omitted). Only the gray states are rewarding. More details are in the main body.

$\mathfrak{A}$ has a leaf in layer $2^{\Omega(L)}$ or in a layer that contains $2^{\Omega(L^2)/\xi}$ nodes. The former case implies a worst case sample complexity requirement that is exponential in L. For the latter, we observe that in order to reach that width, at least $\log_{2^{L+1}} 2^{\Omega(L^2/\xi)}$ transitions and input samples, are required. This is $\Omega(L/\xi)$. □

Class of Hard RDP Instances

For our main lower bound, we define a class of hard RDP instances. Figure 6.2 shows one possible instance in this class. This is the same as fig. 6.1, reported here for convenience, next to the proof where it is used. We will soon define it formally, but we can observe that its structure is organized in two main paths. The two branches in the top part encode a parity computation according to some hidden code $x \in \mathbb{B}^L$, so that behaving optimally in that region requires solving a parity problem (exactly one of lemma 6.18). The bottom part, instead, reaches a two-armed bandit whose optimal action is c. The right-most nodes are winning or losing states that provide a positive and null reward accordingly.

Formally, we define a class of hard RDP instances as $\mathbb{R}(L, H, \xi, \eta) = \{\mathbf{R}_{x,c}\}_{x \in \mathbb{B}^L, c \in \{0,1\}}$ where $\mathbf{R}_{x,c} = \langle \mathcal{Q}, \mathcal{AO}, \Omega, \tau, \theta, q_s, H \rangle$, for $\mathcal{Q} = \{q_s, q_{o_\perp}\} \cup \{q_{0i}, q_{1i}, q_{bi}\}_{i=1,\dots,L} \cup \{q_{+,i}, q_{-,i}\}_{i=L+1,\dots,H}$, $\mathcal{A} = \{a_0', a_1'\}$, $\mathcal{O} = \{0, 1, +, -\}$. Assume $L \geq 1$ and $H > L$. Rewards are zero everywhere, except in the winning states,

$$\theta_\mathsf{r}(r \mid q, a) = \delta_1, \text{ if } q = q_{+i} \text{ with } i > L, \text{ and } \delta_0, \text{ otherwise} \tag{6.71}$$

where we recall that δ_x represents the deterministic distribution on x. For observation probabilities, we denote the distributions $u(o) \coloneqq \mathrm{Unif}\{0,1\}$ and

$$v_\alpha(o) \coloneqq \begin{cases} \frac{1+\alpha}{2} & \text{if } o = + \\ \frac{1-\alpha}{2} & \text{if } o = - \\ 0 & \text{otherwise} \end{cases} \qquad s(o) \coloneqq \begin{cases} 1/4 & \text{if } o = 0 \\ 1/4 & \text{if } o = 1 \\ 1/2 & \text{if } o = + \end{cases} \tag{6.72}$$

Now define observations as

$$\theta_{\mathsf{o}}(q,a,o) = \begin{cases} s(o) & \text{if } q = q_s \\ u(o) & \text{if } q \in \{q_{0i}, q_{1i}\}_{i=1,\dots,L-1} \\ v_\xi(o) & \text{if } q = q_{0L} \wedge a = a'_0 \text{ or } q = q_{1L} \wedge a = a'_1 \\ v_{-\xi}(o) & \text{if } q = q_{0L} \wedge a = a'_1 \text{ or } q = q_{0L} \wedge a = a'_1 \\ \delta_+(o) & \text{if } q = q_{bi} \text{ with } i < L \\ v_0(o) & \text{if } q = q_{bL} \wedge a = a'_0 \\ v_\eta(o) & \text{if } q = q_{bL} \wedge a = a'_1 \wedge c = 1 \\ v_{-\eta}(o) & \text{if } q = q_{bL} \wedge a = a'_1 \wedge c = 0 \\ \delta_{o_\perp}(o) & \text{if } q = q_{o_\perp} \end{cases} \tag{6.73}$$

Finally, the transition function is defined such that $\bar{\tau}(q_s, h_{L-1}) = q_{iL}$ with $i = f_x(o_{0:L-1})$, and

$$\tau(q_{kL}, a+) = q_{+,L+1} \qquad \tau(q_{kL}, a-) = q_{-,L+1} \qquad \text{for } k = 1,2 \tag{6.74}$$

$$\tau(q_{bL}, a+) = q_{+,L+1} \qquad \tau(q_{bL}, a-) = q_{-,L+1} \tag{6.75}$$

$$\tau(q_{+i}, ao) = q_{+,i+1} \qquad \tau(q_{-i}, ao) = q_{-,i+1} \tag{6.76}$$

$$\tau(q_s, a+) = q_{b1} \qquad \tau(q_{bi}, ao) = q_{b,i+1} \qquad \text{for } i < L \tag{6.77}$$

$$\tau(q_{+H}, ao) = q_{o_\perp} \qquad \tau(q_{-H}, ao) = q_{o_\perp} \qquad \tau(q_{o_\perp}, a, o) = q_{o_\perp} \tag{6.78}$$

All the choices above reflect what is shown in the figure. In addition, the transitions $\tau(q_{0i}, ao)$ and $\tau(q_{1i}, ao)$, for $i < L$, are defined according to $o \in \{0,1\}$ and the parity code x. Namely, $\tau(q_{0i}, ao)$ equals $q_{1,i+1}$ iff $o \oplus x(i) = 1$, and $q_{0,i+1}$, otherwise. $\tau(q_{1i}, ao)$ is defined analogously.

Proof of theorem 6.9

Theorem 6.9. *For any $(C^*_{\mathbf{R}}, H, \varepsilon, \mu_0)$ satisfying $C^*_{\mathbf{R}} \geq 2$, $H \geq 2$ and $\varepsilon \leq H\mu_0/64$, there exists an RDP with horizon H, L_1^{p}-distinguishability μ_0 and a regular behaviour policy π^{b} with RDP single-policy concentrability $C^*_{\mathbf{R}}$, such that if $\mathcal{D}$ has been generated* *using* π^{b} *and* $\mathbf{R}$*, and*

$$|\mathcal{D}| \notin \Omega\left(\frac{H}{\mu_0} + \frac{C^*_{\mathbf{R}} H^2}{\varepsilon^2}\right) \tag{6.8}$$

then, for any algorithm $\mathfrak{A} : \mathcal{D} \mapsto \widehat{\pi}$ *returning non-Markov deterministic policies, the probability that* $\widehat{\pi}$ *is not* ε*-optimal is at least* $1/4$.

Proof. Denote with π^{b} a regular policy in $\mathbf{R}$ and $\mathcal{D} \in \mathbb{D}$ a dataset of episodes of length H, collected from $\mathbf{R}$ and the behaviour policy π^{b}. For an RDP $\mathbf{R}$, let $\Pi_{\mathsf{d}} = \mathcal{A}^{\mathcal{H}}$ be the set of deterministic non-Markov policies and $\mathfrak{A} \in (\mathbb{D} \to \Pi_{\mathsf{d}})$ an offline RL algorithm. For $\delta < 0.5$, we say that an algorithm $\mathfrak{A}$ is (ε, δ)-PAC for the class of RDPs $\mathbb{R}$ under condition φ, if, for every $\mathbf{R} \in \mathbb{R}$ and $\mathcal{D} \in \mathbb{D}$, if the condition $\varphi(\mathcal{D}, \pi^{\mathsf{b}})$ is verified, then the output policy $\mathfrak{A}(\mathcal{D})$ is ε-optimal in $\mathbf{R}$, with probability $1 - \delta$. One notable case is that of φ requiring a minimum dataset size.

Since the output of a generic algorithm might be any generic non-Markov deterministic policy, we cannot restrict our attention to regular policies. We expand the value of any history-dependent policy $\pi : \mathcal{H} \to \mathcal{A}$ in an RDP $\mathbf{R}_{x,c} \in \mathbb{R}(L, H, \xi, \eta)$ as follows:

$$\begin{aligned}
V^{\pi}_{\mu} &= \mathbb{E}\left[\sum_{i=1}^{H} r_i \mid \pi\right] && (6.79)\\
&= (H - L)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid \pi) && (6.80)\\
&= (H - L) \sum_{q \in \mathcal{Q}_L} \mathbb{P}(q_L = q \mid \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q, \pi) && (6.81)\\
&= (H - L)\, (\mathbb{P}(q_L = q_{0L} \mid \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{0L}, \pi)\\
&\qquad\quad + \mathbb{P}(q_L = q_{1L} \mid \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{1L}, \pi))\\
&\quad + (H - L)\, (\mathbb{P}(q_L = q_{bL} \mid \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{bL}, \pi)) && (6.82)\\
&= \frac{H - L}{2}\, (\mathbb{P}(q_L = q_{0L} \mid o_0 \in \{0, 1\}, \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{0L}, \pi)\\
&\qquad\quad + \mathbb{P}(q_L = q_{1L} \mid o_0 \in \{0, 1\}, \pi)\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{1L}, \pi))\\
&\quad + \frac{H - L}{2}\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{bL}, \pi) && (6.83)\\
&= \frac{H - L}{4}\, (\mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{0L}, \pi) + \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{1L}, \pi))\\
&\quad + \frac{H - L}{2}\, \mathbb{P}(q_{L+1} = q_{+,L+1} \mid q_L = q_{bL}, \pi) && (6.84)\\
&= \frac{H - L}{4}\, (\mathbb{P}(a_L = a'_0 \mid q_L = q_{0L}, \pi)\, \mathbb{P}(o_L = + \mid q_L = q_{0L}, a_L = a'_0)\\
&\qquad\quad + (1 - \mathbb{P}(a_L = a'_0 \mid q_L = q_{0L}, \pi))\, \mathbb{P}(o_L = + \mid q_L = q_{0L}, a_L = a'_1)\\
&\qquad\quad + (1 - \mathbb{P}(a_L = a'_1 \mid q_L = q_{1L}, \pi))\, \mathbb{P}(o_L = + \mid q_L = q_{1L}, a_L = a'_0)\\
&\qquad\quad + \mathbb{P}(a_L = a'_1 \mid q_L = q_{1L}, \pi)\, \mathbb{P}(o_L = + \mid q_L = q_{1L}, a_L = a'_1))
\end{aligned}$$

$$
\begin{aligned}
&+\frac{H-L}{2}\,(\mathbb{P}(a_L=a_0'\mid q_L=q_{bL},\pi)\,\mathbb{P}(o_L=+\mid q_L=q_{bL},a_L=a_0')\\
&\quad+\mathbb{P}(a_L=a_1'\mid q_L=q_{bL},\pi)\,\mathbb{P}(o_L=+\mid q_L=q_{bL},a_L=a_1')) \qquad (6.85)
\end{aligned}
$$

where in eq. (6.84) we have used the uniform probability over x. Now, for any history-dependent deterministic policy π in episodic RDPs, it is possible to identify an associated regular stochastic policy $\pi_{\mathsf{r}} : \mathcal{Q}' \to \Delta(\mathcal{A})$, where $\mathcal{Q}' \coloneqq \mathcal{Q} \setminus \{q_{o_\perp}\}$ and:

$$
\begin{aligned}
\pi_{\mathsf{r}}(a\mid q) &\coloneqq \mathbb{P}(\pi(h)=a\mid \bar{\tau}(h)=q) && (6.86)\\
&= \sum_{h'\in\bar{\tau}^{-1}(q)} \mathbb{I}(\pi(h')=a)\,\frac{\mathbb{P}(h=h'\mid\pi)}{\mathbb{P}(q\mid\pi)} && (6.87)
\end{aligned}
$$

In other words, π_{r} encodes the probability that π takes action a, given that some history has led to state q. With this convention, we resume from eq. (6.85)

$$
\begin{aligned}
V_\mu^\pi &= \frac{H-L}{4}\,(\pi_{\mathsf{r}}(a_0'\mid q_{0L})\,v_\xi(+)+(1-\pi_{\mathsf{r}}(a_0'\mid q_{0L}))\,v_\xi(-)\\
&\quad+(1-\pi_{\mathsf{r}}(a_1'\mid q_{1L}))\,v_\xi(-)+\pi_{\mathsf{r}}(a_1'\mid q_{1L})\,v_\xi(+)) \qquad (6.88)\\
&\quad+\frac{H-L}{2}\,(\pi_{\mathsf{r}}(a_0'\mid q_{bL})\,u(+)+\pi_{\mathsf{r}}(a_1'\mid q_{bL})\,(\mathbb{I}(c=a_1')\,v_\eta(+)+\mathbb{I}(c=a_0')\,v_\eta(-))\\
&= \frac{H-L}{8}\,(\pi_{\mathsf{r}}(a_0'\mid q_{0L})\,(1+\xi)+(1-\pi_{\mathsf{r}}(a_0'\mid q_{0L}))\,(1-\xi)\\
&\quad+(1-\pi_{\mathsf{r}}(a_1'\mid q_{1L}))\,(1-\xi)+\pi_{\mathsf{r}}(a_1'\mid q_{1L})\,(1+\xi)) \qquad (6.89)\\
&\quad+\frac{H-L}{4}\,(\pi_{\mathsf{r}}(a_0'\mid q_{bL})+\pi_{\mathsf{r}}(a_1'\mid q_{bL})\,(\mathbb{I}(c=a_1')\,(1+\eta)+\mathbb{I}(c=a_0')\,(1-\eta)))\\
&= \frac{H-L}{4}\,(1-\xi+\xi\,\pi_{\mathsf{r}}(a_0'\mid q_{0L})+\xi\,\pi_{\mathsf{r}}(a_1'\mid q_{1L})\\
&\quad+\pi_{\mathsf{r}}(a_0'\mid q_{bL})+\pi_{\mathsf{r}}(a_1'\mid q_{bL})\,(1+\eta\,\mathbb{I}(c=a_1')-\eta\,\mathbb{I}(c=a_0'))) \qquad (6.90)
\end{aligned}
$$

For the optimal policy, in particular, this becomes:

$$
V_\mu^* = \frac{H-L}{4}\,(1+\xi+\mathbb{I}(c=a_0')+(1+\eta)\,\mathbb{I}(c=a_1')) \qquad (6.91)
$$

From the ε-optimality of $\pi=\mathfrak{A}(\mathcal{D})$, then,

$$
\begin{aligned}
\varepsilon &\geq V_\mu^* - V_\mu^\pi && (6.92)\\
&= \frac{H-L}{4}(2\xi-\xi\,\pi_{\mathsf{r}}(a_0'\mid q_{0L})-\xi\,\pi_{\mathsf{r}}(a_1'\mid q_{1L})\\
&\quad+\eta\,\mathbb{I}(c=a_1')\,(1-\pi_{\mathsf{r}}(a_1'\mid q_{bL}))+\eta\,\mathbb{I}(c=a_0')\,\pi_{\mathsf{r}}(a_1'\mid q_{bL})) && (6.93)\\
&= \frac{H-L}{4}(\xi\,(2-\pi_{\mathsf{r}}(a_0'\mid q_{0L})-\pi_{\mathsf{r}}(a_1'\mid q_{1L}))+\eta\,(1-\pi_{\mathsf{r}}(c\mid q_{bL}))) && (6.94)\\
&\geq \frac{H-L}{4}\max\{\xi\,(1-\pi_{\mathsf{r}}(a_0'\mid q_{0L})),\xi\,(1-\pi_{\mathsf{r}}(a_1'\mid q_{1L})),\eta\,(1-\pi_{\mathsf{r}}(c\mid q_{bL}))\} && (6.95)
\end{aligned}
$$

Now, assume that

$$\min\{\xi, \eta\} \geq \frac{16\,\varepsilon}{H - L} \tag{6.96}$$

Then, all the following is true: $\pi_{\mathsf{r}}(a'_0 \mid q_{0L}) \geq 3/4$, $\pi_{\mathsf{r}}(a'_1 \mid q_{1L}) \geq 3/4$, $\pi_{\mathsf{r}}(c \mid q_{bL}) \geq 3/4$. This means that, for small ε, any ε-optimal policy must frequently select the optimal action for both the parity problem and the bandit. Let us represent the first two events with B_{p} and the third with B_{b}. Since $\mathfrak{A}$ is (ε, δ)-PAC for $\mathbb{R}(L, H, \xi, \eta)$ under φ, the probability of $B_{\mathsf{p}} \wedge B_{\mathsf{b}}$ is at least $1 - \delta$, for any $\mathcal{D}$ and π^{b} satisfying $\varphi(\mathcal{D}, \pi^{\mathsf{b}})$.

We proceed to compute the necessary data to satisfy both events with high probability. The dataset $\mathcal{D}$ can be partitioned in two subsets $\mathcal{D}_{\mathsf{p}}$ and $\mathcal{D}_{\mathsf{b}}$, containing any episode from $\mathcal{D}$ whose initial observation is $\{0, 1\}$ and $+$, respectively. The two datasets share no information and $\mathcal{D}_{\mathsf{p}}$ and $\mathcal{D}_{\mathsf{b}}$ are mutually independent. To see this, we observe that the sequence $a_{L+1} r_{L+1} o_{L+1} \dots o_H$ is independent of $a_0 r_0 o_0 \dots a_L$ given o_L, since the observations $+$ and $-$ at step L uniquely determine the rest of the episode. Also, for any two episodes e_H, e'_H, the sequence $a_1 r_1 o_1 \dots o_L$ is independent of $a'_1 r'_1 o'_1 \dots o'_L$ given o_0. Since, $o_0 \sim \mu = s$, that is the initial observation distribution in these RDPs, the two datasets are independent. Let $\mathcal{Q}_{\mathsf{p}} = \{q_s, q_{o_\perp}\} \cup \{q_{0i}, q_{1i}\}_{i=1,\dots,L} \cup \{q_{+,i}, q_{-,i}\}_{i=L+1,\dots,H}$ and $\mathcal{Q}_{\mathsf{b}} = \{q_s, q_{o_\perp}\} \cup \{q_{bi}\}_{i=1,\dots,L} \cup \{q_{+,i}, q_{-,i}\}_{i=L+1,\dots,H}$ be the reachable states in the two datasets. Then, we consider two separate classes $\mathbb{R}(L, H, \xi)$ and $\mathbb{R}(L, H, \eta)$ as the sets of RDPs in $\mathbb{R}(L, H, \xi, \eta)$, restricted to $\mathcal{Q}_{\mathsf{p}}$ and $\mathcal{Q}_{\mathsf{b}}$, respectively. To do so, we construct $\mathbf{R}_r \in \mathbb{R}(L, H, \xi)$ and $\mathbf{R}_c \in \mathbb{R}(L, H, \eta)$ such that the initial observation follows $\mathrm{Unif}(\{0, 1\})$ in $\mathbf{R}_r$ and δ_+ in $\mathbf{R}_c$. Now, from the independence of the two datasets and the fact that $\mathfrak{A}$ is (ε, δ)-PAC in $\mathcal{D}$, there must exist an algorithm $\mathfrak{A}_{\mathsf{p}} : \mathcal{D}_{\mathsf{p}} \mapsto \pi_{\mathsf{p}}$ that is $(2\varepsilon, \delta)$-PAC in $\mathbb{R}(L, H, \xi)$ under some φ_{p}, and $\mathfrak{A}_{\mathsf{b}} : \mathcal{D}_{\mathsf{b}} \mapsto \pi_{\mathsf{b}}$ that is $(2\varepsilon, \delta)$-PAC in $\mathbb{R}(L, H, \eta)$ under some φ_{b}. If this was not the case, $B_{\mathsf{p}} \wedge B_{\mathsf{b}}$ could not be verified in one of the two terms.

We analyse $\mathfrak{A}_{\mathsf{p}}$ first, and we show that its requirement φ_{p} is $|\mathcal{D}_{\mathsf{p}}| \in \Omega(L/\xi) \cup 2^{\Omega(L)}$. For a contradiction, assume this is not the case and that $|\mathcal{D}_{\mathsf{p}}| = g(L, \xi) \notin (\Omega(L/\xi) \cup 2^{\Omega(L)})$ is allowed. Then, we can use $\mathfrak{A}_{\mathsf{p}}$ to solve the noisy parity problem under the streaming setting with $g(L, \xi)$ samples (this setting has been introduced at page 162). We proceed as follows. Consider any noisy parity function $f_{x,\xi}$ with unknown x. Sample a sequence of strings $\{y_i\}_i \in 2^L$ from the uniform distribution and collect $g(L, \xi)$ pairs (y_i, p_i), sampling $p_i \sim f_{x,\xi}(y_i)$. Then, for $H > L$, compose a dataset of episodes $\{e_i\}_i$. All actions of e_i are selected uniformly in $\{a'_0, a'_1\}$. The observations $o_{0:L-1}$ are y_i and o_L equals p_i if $a_L = a'_0$, $1 - p_i$, otherwise (0 and 1 take roles of $+$ and $-$ symbols here). Rewards $r_{L+1:H}$ are equal to one if $o_L = 1$, null otherwise.

We obtain that dataset so constructed is equally likely under this procedure than under the uniform policy and the RDP $\mathbf{R}_x \in \mathbb{R}(L, H, \xi)$. Since $\mathfrak{A}_{\mathsf{p}}$ is $(2\varepsilon, \delta)$-PAC for $\mathbb{R}(L, H, \xi)$, with probability $1 - \delta$, the output policy π_{p} satisfies:

$$\min\{\pi_{\mathsf{pr}}(a'_0 \mid q_{0L}), \pi_{\mathsf{pr}}(a'_1 \mid q_{1L})\} \geq 3/4 \tag{6.97}$$

where π_{pr} is the stochastic regular policy for π_{p}. This can be seen by our assumption in eq. (6.96) and doubling both ε and the sub-optimality gap of eq. (6.95), due to the updated probability for the initial observation. Then, for any sequence $y \in 2^L$ and associated history h_{L-1} with $o_{0:L-1} = y$,

$$f_x(y) = \arg\max_{i=0,1} \pi_{\mathsf{p}}(a'_i \mid h_{L-1}) \tag{6.98}$$

which is the noiseless parity function based on x. This means that it is possible to reconstruct x solely by interacting with π_{p}, without collecting further samples. The solution we have described is a streaming algorithm with sample complexity $g(L, \xi)$. Since this contradicts lemma 6.18, we have proven $|\mathcal{D}_{\mathsf{p}}| \in \Omega(L/\xi) \cup 2^{\Omega(L)}$.

We now consider the bandit problem, which is solved by $\mathfrak{A}_{\mathsf{b}}$. Similarly to the previous case, from the ε-optimality of $\mathfrak{A}_{\mathsf{b}}(\mathcal{D}_{\mathsf{b}})$, we obtain the necessary condition: $\pi_{\mathsf{br}}(c \mid q_{bL}) \geq 3/4$ from eq. (6.95). This condition is expressed for the stochastic policy π_{br}. However, we notice that for q_{bL} in particular, the only possible history is $h_{L-1} = a_0{+}a_1{\dots}{+}$, where all actions must also be deterministic. Then,

$$\pi_{\mathsf{br}}(c \mid q_{bL}) = \mathbb{P}(\pi_{\mathsf{b}}(h) = c \mid \bar{\tau}(h) = q_{bL}) = \mathbb{I}(\pi_{\mathsf{b}}(h_{L-1}) = c) \tag{6.99}$$

implying that π_{br} can only be deterministic for q_{bL}. This means that $\mathfrak{A}_{\mathsf{b}}$ must solve best-arm identification in the two arm bandit at q_{bL}. We can compose a simplified dataset that is relevant for the bandit as:

$$\mathcal{D}'_{\mathsf{b}} = \{a_L o_L : e_H \in \mathcal{D}_{\mathsf{b}}\} \tag{6.100}$$

Since $\mathcal{D}_{\mathsf{b}}$ can be deterministically reconstructed from $\mathcal{D}'_{\mathsf{b}}$, we have the following conditional independence: $\pi_{\mathsf{b}} \perp c \mid \mathcal{D}'_{\mathsf{b}}$, where $c \in \{a'_0, a'_1\}$ is the optimal arm, and $\pi_{\mathsf{b}} = \mathfrak{A}_{\mathsf{b}}(\mathcal{D}_{\mathsf{b}})$ is the output of the algorithm. Denoting with $\hat{c} = \pi_{\mathsf{b}}(h_{L-1})$ the selected arm, the error probability is $P_e \coloneqq \mathbb{P}(\hat{c} \neq c)$. The application of Fano's inequality from theorem 6.12 to the variables $c \to \mathcal{D}'_{\mathsf{b}} \to \hat{c}$ gives:

$$\begin{aligned} H_2(P_e) &\geq H(c \mid \mathcal{D}'_{\mathsf{b}}) & (6.101)\\ &= H(c) - I(c; \mathcal{D}'_{\mathsf{b}}) = \log 2 - I(c; \mathcal{D}'_{\mathsf{b}}) & (6.102) \end{aligned}$$

where we have used the fact that $\hat{c}$ is a Bernoulli variable and the uniform prior over

c. Now, assuming $C \geq 2$, we construct a behaviour policy as $\pi^{\mathsf{b}}(a_0 \mid q_{bL}) = 1 - 1/C$ and $\pi^{\mathsf{b}}(a_1 \mid q_{bL}) = 1/C$. In the following, we write $N_{\mathsf{b}} \coloneqq |\mathcal{D}_{\mathsf{b}}|$ and omit the implicit dependency on π^{b}.

$$I(c\,;\mathcal{D}'_{\mathsf{b}}) = H(\mathcal{D}'_{\mathsf{b}}) - H(\mathcal{D}'_{\mathsf{b}} \mid c) \tag{6.103}$$

$$= N_{\mathsf{b}}(H(a_L o_L) - H(a_L o_L \mid c)) \tag{6.104}$$

$$= N_{\mathsf{b}} D_{KL}(\mathbb{P}(a_L o_L, c) \parallel \mathbb{P}(a_L o_L)\,\mathbb{P}(c)) \tag{6.105}$$

$$= \frac{N_{\mathsf{b}}}{2} \sum_{a,c' \in \mathcal{A}, o \in \mathcal{O}} \mathbb{P}(a, o \mid c') \log \frac{\mathbb{P}(a, o \mid c')}{\mathbb{P}(a, o)} \tag{6.106}$$

$$= \frac{N_{\mathsf{b}}}{2} \sum_{a,c' \in \mathcal{A}, o \in \mathcal{O}} \mathbb{P}(a, o \mid c') \log \frac{\mathbb{P}(a \mid c')\,\mathbb{P}(o \mid c', a)}{\sum_{c''} \mathbb{P}(a \mid c'')\,\mathbb{P}(o \mid c'', a)/2} \tag{6.107}$$

$$= \frac{N_{\mathsf{b}}}{2} \sum_{a,c' \in \mathcal{A}, o \in \mathcal{O}} \mathbb{P}(a, o \mid c') \log \frac{2\mathbb{P}(o \mid c', a)}{\sum_{c''} \mathbb{P}(o \mid c'', a)} \tag{6.108}$$

$$= \frac{N_{\mathsf{b}}}{2} \sum_{a,c' \in \mathcal{A}, o \in \mathcal{O}} \mathbb{P}(a, o \mid c') \log(2\mathbb{P}(o \mid c', a)) \tag{6.109}$$

$$= \frac{N_{\mathsf{b}}}{2} \sum_{c' \in \mathcal{A}, o \in \mathcal{O}} \mathbb{P}(a'_1, o \mid c') \log(2\mathbb{P}(o \mid c', a'_1)) \tag{6.110}$$

$$\begin{aligned} &= \frac{N_{\mathsf{b}}}{2} \sum_{o \in \mathcal{O}} (\mathbb{P}(a'_1, o \mid c = a'_0) \log(2\mathbb{P}(o \mid c = a'_0, a'_1)) \\ &\qquad + \mathbb{P}(a'_1, o \mid c = a'_1) \log(2\mathbb{P}(o \mid c = a'_1, a'_1))) \end{aligned} \tag{6.111}$$

$$\begin{aligned} &= N_{\mathsf{b}}(\mathbb{P}(a'_1, + \mid c = a'_0) \log(2\mathbb{P}(+ \mid c = a'_0, a'_1)) \\ &\qquad + \mathbb{P}(a'_1, + \mid c = a'_1) \log(2\mathbb{P}(+ \mid c = a'_1, a'_1))) \end{aligned} \tag{6.112}$$

$$= N_{\mathsf{b}}\Big(\frac{1-\eta}{2C} \log(1-\eta) + \frac{1+\eta}{2C} \log(1+\eta)\Big) \tag{6.113}$$

$$= \frac{N_{\mathsf{b}}}{C} D_{KL}(v_\eta \parallel v_0) \tag{6.114}$$

$$\leq \frac{N_{\mathsf{b}}\,\eta^2}{C} \tag{6.115}$$

Then from eq. (6.102), and the fact that $\mathfrak{A}_{\mathsf{b}}$ is $(2\varepsilon, \delta)$-PAC,

$$H(\delta) \geq H_2(P_e) \geq \log 2 - \frac{N_{\mathsf{b}}\,\eta^2}{C} \tag{6.116}$$

$$\implies N_{\mathsf{b}} \geq \frac{C}{\eta^2}(\log 2 - H(\delta)) \tag{6.117}$$

Which means that this must be φ_{b}, the requirement for $\mathfrak{A}_{\mathsf{b}}$.

Finally, to compose the results from both branches, we observe that $|\mathcal{D}| = |\mathcal{D}_{\mathsf{p}}| + |\mathcal{D}_{\mathsf{b}}|$. Also, for any $\delta \in (0, 0.5)$, say $1/4$, $(\log 2 - H(\delta))$ becomes a positive constant, and we can add both sizes asymptotically:

$$|\mathcal{D}| \in \Omega\left(\frac{H}{\xi} + \frac{C}{\eta^2}\right) \tag{6.118}$$

To relate the parameters to features of the RDP, we observe that the number of states of any RDP in $\mathbb{R}(L, H, \xi, \eta)$ is $Q \leq 3H$. Also, the behaviour policy is uniform everywhere except in q_{bL}. Assuming $C \geq 2$, the computation of the single-policy concentrability coefficient yields $C^*_{\mathbf{R}} = C$, for any $c \in \{a'_0, a'_1\}$. Next, we compute the L_1^p-distinguishability of any RDP in this class. The L_1^p-distinguishability of a set of states $\mathcal{Q}$ is the minimum L_1 distance in distribution between episodes prefixes that are generated starting from any two states in $\mathcal{Q}$. Let us consider the L_1 norm for the pair q_{01} and q_{11},

$$\begin{aligned}
&\|\mathbb{P}(e_{1:H} \mid q_{01}, \pi^{\mathsf{b}}) - \mathbb{P}(e_{1:H} \mid q_{01}, \pi^{\mathsf{b}})\|_1 = && (6.119)\\
&\quad = \sum_{e \in \mathcal{T}_H} |\mathbb{P}(e_{1:H} = e \mid q_{01}) - \mathbb{P}(e_{1:H} = e \mid q_{11})| && (6.120)\\
&\quad = \sum_{earo \in \mathcal{T}_{L+1}} \mathbb{P}(e_{1:L-1} = e)|\mathbb{P}(a_L = a, r_L = r, o_L = o \mid q_{01}, e) \\
&\qquad\quad - \mathbb{P}(a_L = a, r_L = r, o_L = o \mid q_{11}, e)| && (6.121)\\
&\quad = \sum_{ao \in \mathcal{AO}} |\mathbb{P}(a_L = a, o_L = o \mid q_{0L}) - \mathbb{P}(a_L = a, o_L = o \mid q_{1L})| && (6.122)\\
&\quad = (1/2) \sum_{o \in \mathcal{O}} |\mathbb{P}(o_L = o \mid a_L = a'_0, q_{0L}) - \mathbb{P}(o_L = o \mid a_L = a'_0, q_{1L})| && (6.123)\\
&\qquad\quad + (1/2) \sum_{o \in \mathcal{O}} |\mathbb{P}(o_L = o \mid a_L = a'_1, q_{0L}) - \mathbb{P}(o_L = o \mid a_L = a'_1, q_{1L})| && (6.124)\\
&\quad = \sum_{o \in \mathcal{O}} |\mathbb{P}(o_L = o \mid a_L = a'_0, q_{0L}) - \mathbb{P}(o_L = o \mid a_L = a'_0, q_{1L})| && (6.125)\\
&\quad = 2|\mathbb{P}(o_L = + \mid a_L = a'_0, q_{0L}) - \mathbb{P}(o_L = + \mid a_L = a'_0, q_{1L})| && (6.126)\\
&\quad = 2\xi && (6.127)
\end{aligned}$$

The L_1 distance of suffixes from q_{01}, q_{11} that are longer than L have also a distance of 2ξ. On the other hand, any shorter prefix has a distance of 0. Since, L_1^{p} minimizes across all these distances, the minimum is attained for any q_{0i} and q_{1i}, which determines $\mu_0 \geq 2\xi$. In fact, the distance between any other pair of states in the same layer is strictly higher, since they differ deterministically in some reward or observation. Hence, the L_1^{p}-distinguishability of the entire RDP is $\mu_0 = 2\xi$. Now, we choose $L = H/2$, $\eta = 32\,\varepsilon/H$ and we assume $\varepsilon \leq H\mu_0/64$, $H \geq 2$. We can verify that these choices are consistent with the previous assumption $\min\{\xi, \eta\} \geq \frac{16\,\varepsilon}{H-L}$. Substituting, the final requirement φ for the complete algorithm $\mathfrak{A}$ is an exponential number of episodes in H or:

$$|\mathcal{D}| \in \Omega\left(\frac{H}{\mu_0} + \frac{C^*_{\mathbf{R}} H^2}{\varepsilon^2}\right) \tag{6.128}$$

Now, for any $H, \mu_0, C^*_{\mathbf{R}}, \varepsilon$ satisfying the previous assumptions, any algorithm cannot be $(\varepsilon, 1/4)$-optimal for the instances in $\mathbb{R}(H/2, H, \mu_0, 32\,\varepsilon/H)$ if eq. (6.128) is not satisfied. □

Note that in our RDP instance, the number of states and the horizon length scale linearly. So, we might equivalently write HQ instead of H^2.

Chapter 7

Conclusion

Thanks to the many successes achieved in complex environments, Reinforcement Learning is now the leading AI research field for the development of intelligent agents for decision-making. In simulated environments, specifically, there has been impressive progress in many benchmarks, including ATARI games, physical simulations, games with first-person views, and open, strategic games, (Mnih, Kavukcuoglu, et al. 2015; Schulman, Levine, et al. 2015; Kempka, Wydmuch, et al. 2016; Mnih, Badia, et al. 2016; Pohlen, Piot, et al. 2018; Lample and Chaplot 2016; Oh, Chockalingam, et al. 2016). This is significant, especially considering the little prior knowledge that RL requires. However, despite the significant progress, two important features remain unsatisfactory in AI agents. Namely, RL algorithms should be both generally applicable and efficient. Although these two features are arguably part of an inevitable compromise and may not be perfectly optimised at the same time, there still seem to be plenty of opportunities to improve RL algorithms in both directions.

This dissertation summarises a progressive effort whose purpose is improving RL methods in both the directions just described: efficiency and general applicability. Regarding efficiency, a series of excellent results showed that it is possible to apply RL algorithms in large MDPs (C. Jin, Z. Yang, et al. 2020; François-Lavet, Henderson, et al. 2018). However, despite the increased efficiency, the classic "flat" RL algorithms often fail to exploit specific structures in the environment dynamics. For example, by recognising that a complex problem can be regarded as the composition of two connected subtasks, a person would easily approach each subtask independently and, when possible, would reconstruct the global solution by composing its parts. The whole field of Hierarchical RL aims at developing RL agents that are capable of this reasoning. This would also allow us to reuse previous solutions, when available. As a whole, part II of this dissertation proceeds in this research direction.

The second general objective that we identified above is the achievement of a more general applicability of the RL algorithms. Specifically, many theoretical results and algorithms are available for fully observable environments. However, despite the pervasive presence of partial information in realistic scenarios, comparatively little progress has been made with respect to MDPs. This does not mean that RL algorithms are not currently employed in the presence of partial observations. As a significant example, some simulators with first-person views provide very limited perceptions. Nonetheless, some excellent results are already available (Lample and Chaplot 2017; Baker, Kanitscheider, et al. 2020). However, instead of directly accounting for partial observations and non-Markovian dependencies in the algorithm, most RL methods employ techniques that have been originally developed for MDPs and delegate all the temporal complexity to the internal neural network architecture. Although recurrent architectures do have the expressive power for capturing such dependencies, in principle, complex non-Markovian relations strongly complicate the optimisation landscape and may lead to well-known instabilities for RNNs. Unlike these works, in part III, we directly target the Reinforcement Learning problem in non-Markovian environments.

Together, the two parts of this thesis share the common need of developing algorithms that target the most appropriate state representation for each environment. In fact, selecting appropriate representations should be an important component for flexible AI agents (G. Konidaris 2019). For part II, this would be a representation that allows near-optimal behaviours, while avoiding non-stationarity effects. For part III, instead, this would be a representation that allows the agents to accurately predict future rewards and plan for them. Despite the similarities, each part operates in a different context and requires some specific techniques. Therefore, we summarise some specific conclusions for each of them in the following. A detailed list of contributions can be found at the beginning of this thesis and in the opening and closing of each chapter. Instead, the purpose of the next sections is to summarise the general state of this work and to suggest directions for future work.

Learning With MDP Abstractions

In chapter 3, we proposed a new RL algorithm for incorporating additional prior knowledge, in the form of an abstract MDP simulator, into the learning routine. Specifically, the algorithm persistently influences the exploration policy of the ground MDP, while retaining the original optimal convergence guarantees. Furthermore, we identified a relationship between abstraction and ground MDP, through a comparison of the induced exploration policy.

Beyond the specific results, it is worth noting that the associated theoretical results had required the identification of two very intuitive parameters, abstract value approximation and abstract similarity, that characterise the quality of the state partitioning and the ground options for those partitions. In fact, these parameters have been the core motivation for seeking a more elegant and direct relationship between the two models in chapter 4. In this follow-up work, we step back from the specific algorithms and try to answer a more general question for HRL. Namely, the purpose of this chapter is to identify sufficient conditions that would enable the translation of abstract policies into ground policies by learning in a truly compositional way. This chapter is a major step forward, with respect to more sound definitions of MDP abstractions, since it gives important insights related to the quality of different state partitions, the nature of values at exit states, the constrained MDP formulation of the realizability problem, and the conditions allowing for a compressed effective horizon in the abstraction.

Although we were able to conclude this part of the dissertation with a good number of insights related to HRL theory, the specific applicability of realizable abstractions remains open-ended. In fact, we have identified some specific components and solutions related to the realizability problem. However, a complete RL algorithm based on realizable abstractions has not been developed yet. There are two very natural directions for future work that extend from this dissertation. The first is the development of a highly sample-efficient algorithm that takes advantage of the nature of realizable abstractions. The second is the development of learning algorithms that are capable of finding realizable abstractions from ground MDPs.

Learning in Non-Markov Decision Processes

In chapter 5, we studied the expressive power of Regular Decision Processes. This is a very relevant topic for RL in non-Markovian environments, because it is motivated by the general intractability of RL in POMDPs. In this thesis, we have shown that the class of RDPs is distinct from POMDPs and is placed in the strict relation: k-MDP $\subset$ RDP $\subset$ POMDP. So, RDPs have the potential to be a very interesting middle ground between relatively limited models, such as k-MDPs, and the full expressiveness of POMDPs. We also showed that RDPs can approximate a large subset of POMDPs. In chapter 6, on the other hand, we have provided an original offline RL algorithm for RDPs, with associated sample-efficiency guarantees.

Although this dissertation contributes significantly to the current understanding of RDPs, many questions remain open. With respect to expressive power, we have not yet proved whether RDPs can act as general approximators for POMDPs. In both cases of a positive or negative response, this result would have an interesting impact in shaping the landscape of the existing models for non-Markovian RL.

With respect to the theoretical results regarding the complexity of RDP learning, both chapters showed lower bounds on the sample complexity of RL in RDPs, for the online and the offline setting, respectively. The lower bound of chapter 5 indicates that, in the general case, RL in RDPs can be intractable. However, as we saw from the lower bound in chapter 6, the complexity of (offline) RL in RDPs can also be characterised with more precision, if some notion related to temporal complexity is introduced. In this work, we have used "distinguishability" parameters, which come from the theory of probabilistic automata. Interestingly, this parameter has not been previously used for POMDPs. In fact, RDPs might admit different parameterizations compared to those currently available for POMDPs.

In conclusion, the algorithm proposed in chapter 6 has the advantage of being extremely modular, thanks to its internal reduction to a Markovian environment. However, this two-step approach precludes a series of optimisations that are also worth exploring. This is especially important when developing RL algorithms for the online setting, in which careful exploration is of paramount importance.

> Both parts of this dissertation are related to two very active branches of the RL literature. I am confident that many of the open questions that we discussed in this conclusion will be solved in the near future, thanks to the joint effort of this active community.

Bibliography

Abadi, Eden and Ronen I. Brafman (2020). "Learning and Solving Regular Decision Processes". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, pp. 1948–1954.

Abel, David (2020). "A Theory of Abstraction in Reinforcement Learning". PhD thesis. Brown University, USA.

Abel, David, David Hershkowitz, and Michael Littman (2016). "Near Optimal Behavior via Approximate State Abstraction". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR, pp. 2915–2923.

Abel, David, Nate Umbanhowar, Khimya Khetarpal, Dilip Arumugam, Doina Precup, and Michael Littman (2020). "Value Preserving State-Action Abstractions". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 1639–1650.

Achiam, Joshua, David Held, Aviv Tamar, and Pieter Abbeel (2017). "Constrained Policy Optimization". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 22–31.

Agarwal, Alekh, Nan Jiang, Sham M. Kakade, and Wen Sun (2021). *Reinforcement Learning: Theory and Algorithms*. 205 pp.

Altman, Eitan (1999). *Constrained Markov Decision Processes*. 1st ed. 256 pp. ISBN: 978-1-315-14022-3.

Åström, K.J (1965). "Optimal Control of Markov Processes with Incomplete State Information". In: *Journal of Mathematical Analysis and Applications* 10.1, pp. 174–205. ISSN: 0022-247X.

Auer, Peter, Nicolò Cesa-Bianchi, and Paul Fischer (2002). "Finite-Time Analysis of the Multiarmed Bandit Problem". In: *Machine Learning* 47.2, pp. 235–256. ISSN: 1573-0565.

Azar, Mohammad Gheshlaghi, Ian Osband, and Rémi Munos (2017). "Minimax Regret Bounds for Reinforcement Learning". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 263–272.

Azizzadenesheli, Kamyar, Alessandro Lazaric, and Animashree Anandkumar (2016). "Reinforcement Learning of POMDPs Using Spectral Methods". In: *COLT*. Vol. 49. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 193–256.

Bacchus, Fahiem, Craig Boutilier, and Adam Grove (1996). "Rewarding Behaviors". In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2* (Portland, Oregon). AAAI'96. AAAI Press, pp. 1160–1167. ISBN: 0-262-51091-X.

Bacon, Pierre-Luc, Jean Harb, and Doina Precup (2017). "The Option-Critic Architecture". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 1726–1734.

Baker, Bowen, Ingmar Kanitscheider, Todor M. Markov, Yi Wu, et al. (2020). "Emergent Tool Use from Multi-Agent Autocurricula". In: *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Bakker, Bram (2001). "Reinforcement Learning with Long Short-Term Memory". In: *Advances in Neural Information Processing Systems 14, NIPS 2001*. MIT Press, pp. 1475–1482.

Balle, Borja, Jorge Castro, and Ricard Gavaldà (2013). "Learning Probabilistic Automata: A Study in State Distinguishability". In: *Theoretical Computer Science* 473, pp. 46–60. DOI: 10.1016/j.tcs.2012.10.009.

— (2014). "Adaptively Learning Probabilistic Deterministic Automata from Data Streams". In: *Machine Learning* 96.1, pp. 99–127. ISSN: 1573-0565. DOI: 10.1007/s10994-013-5408-x.

Balle Pigem, Borja de (2013). "Learning Finite-State Machines: Statistical and Algorithmic Aspects". Universitat Politècnica de Catalunya. 175 pp.

Bellemare, Marc G., Will Dabney, and Rémi Munos (2017). "A Distributional Perspective on Reinforcement Learning". In: *ICML*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 449–458.

Bellman, Richard (1956). "Dynamic Programming and Lagrange Multipliers". In: *Proceedings of the National Academy of Sciences* 42.10, pp. 767–769.

— (1958). "Dynamic Programming and Stochastic Control Processes". In: *Inf. Control.* 1.3, pp. 228–239.

Bertsekas, Dimitri P. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific.

Biza, Ondrej and Robert Platt Jr. (2019). “Online Abstraction with MDP Homomorphisms for Deep Learning”. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’19*. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1125–1133.

Blum, Avrim, Merrick L. Furst, Jeffrey C. Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich (1994). “Weakly Learning DNF and Characterizing Statistical Query Learning Using Fourier Analysis”. In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*. ACM, pp. 253–262. DOI: 10.1145/195058.195147.

Bowling, Michael H., Peter McCracken, Michael James, James Neufeld, and Dana F. Wilkinson (2006). “Learning Predictive State Representations Using Non-Blind Policies”. In: *Machine Learning, Proceedings of the Twenty-Third International Conference, ICML 2006*. Vol. 148. ACM International Conference Proceeding Series. ACM, pp. 129–136. DOI: 10.1145/1143844.1143861.

Boyen, Xavier and Daphne Koller (1998). “Tractable Inference for Complex Stochastic Processes”. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI ’98*. Morgan Kaufmann, pp. 33–42.

Brafman, Ronen I. and Giuseppe De Giacomo (2019). “Regular Decision Processes: A Model for Non-Markovian Domains”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 5516–5522. ISBN: 978-0-9992411-4-1.

Brafman, Ronen I., Giuseppe De Giacomo, and Fabio Patrizi (2018). “LTLf / LDLf Non-Markovian Rewards”. In: *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 1771–1778.

Brafman, Ronen I. and Moshe Tennenholtz (2003). “R-Max - a General Polynomial Time Algorithm for near-Optimal Reinforcement Learning”. In: *J. Mach. Learn. Res.* 3 (null), pp. 213–231. ISSN: 1532-4435.

Camacho, Alberto, Oscar Chen, Scott Sanner, and Sheila A. McIlraith (2017). “Non-Markovian Rewards Expressed in LTL: Guiding Search via Reward Shaping”. In: *Proceedings of the Tenth International Symposium on Combinatorial Search, SOCS 2017*. AAAI Press, pp. 159–160.

Camacho, Alberto, Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Anthony Valenzano, and Sheila A. McIlraith (2019). “LTL and beyond: Formal Languages for Reward Function Specification in Reinforcement Learning”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, pp. 6065–6073. DOI: 10.24963/IJCAI.2019/840.

Castro, Pablo Samuel and Doina Precup (2011). “Automatic Construction of Temporally Extended Actions for MDPs Using Bisimulation Metrics”. In: *Recent Advances in Reinforcement Learning - 9th European Workshop, EWRL 2011*. Vol. 7188. Lecture Notes in Computer Science. Springer, pp. 140–152.

Chen, Jinglin and Nan Jiang (2019). “Information-Theoretic Considerations in Batch Reinforcement Learning”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 1042–1051.

Chen, Yichen and Mengdi Wang (2016). “Stochastic Primal-Dual Methods and Sample Complexity of Reinforcement Learning”. In: *CoRR* abs/1612.02516. arXiv: `1612.02516`.

Cipollone, Roberto, Giuseppe De Giacomo, Marco Favorito, Luca Iocchi, and Fabio Patrizi (2023a). “Exploiting Multiple Abstractions in Episodic RL via Reward Shaping”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37, pp. 7227–7234.

— (2023b). “Exploiting Multiple Abstractions in Episodic RL via Reward Shaping”. In: *CoRR* abs/2303.00516.

Cipollone, Roberto, Anders Jonsson, Alessandro Ronca, and Mohammad Sadegh Talebi (2024). “Provably Efficient Offline Reinforcement Learning in Regular Decision Processes”. In: *Thirty-Seventh Conference on Neural Information Processing Systems, NeurIPS 2024*.

Clark, Alexander and Franck Thollard (2004). “PAC-learnability of Probabilistic Deterministic Finite State Automata”. In: *Journal of Machine Learning Research* 5, pp. 473–497.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory (2. Ed.)* Wiley. ISBN: 978-0-471-24195-9.

Csiszár, Imre and Zsolt Talata (2006). “Context Tree Estimation for Not Necessarily Finite Memory Processes, via BIC and MDL”. In: *IEEE Trans. Inf. Theory* 52.3, pp. 1007–1016. DOI: `10.1109/TIT.2005.864431`.

Dann, Christoph and Emma Brunskill (2015). “Sample Complexity of Episodic Fixed-Horizon Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 28, NIPS 2015*, pp. 2818–2826.

Dann, Christoph, Tor Lattimore, and Emma Brunskill (2017). “Unifying PAC and Regret: Uniform PAC Bounds for Episodic Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 5713–5723.

Dayan, Peter (1993). “Improving Generalization for Temporal Difference Learning: The Successor Representation”. In: *Neural Computation* 5.4, pp. 613–624. DOI: 10.1162/neco.1993.5.4.613.

Dayan, Peter and Geoffrey E. Hinton (1992). “Feudal Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 5, NIPS 1992*. Morgan Kaufmann, pp. 271–278.

De Giacomo, Giuseppe, Marco Favorito, Luca Iocchi, Fabio Patrizi, and Alessandro Ronca (2020). “Temporal Logic Monitoring Rewards via Transducers”. In: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, pp. 860–870. DOI: 10.24963/KR.2020/89.

De Giacomo, Giuseppe, Luca Iocchi, Marco Favorito, and Fabio Patrizi (2019). “Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications”. In: *Proceedings International Conference on Automated Planning and Scheduling, ICAPS* (Brooks 1991), pp. 128–136. ISSN: 23340843.

De Farias, Daniela Pucci and Benjamin Van Roy (2003). “The Linear Programming Approach to Approximate Dynamic Programming”. In: *Operations Research* 51.6, pp. 850–865. DOI: 10.1287/OPRE.51.6.850.24925.

Devlin, Sam and Daniel Kudenko (2012). “Dynamic Potential-Based Reward Shaping”. In: *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012*. IFAAMAS, pp. 433–440.

Dietterich, Thomas G. (2000). “Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition”. In: *J. Artif. Intell. Res.* 13, pp. 227–303.

Eckstein, Maria K. and Anne G. E. Collins (2020). “Computational Evidence for Hierarchically Structured Reinforcement Learning in Humans”. In: *Proceedings of the National Academy of Sciences* 117.47, pp. 29381–29389.

Efroni, Yonathan, Chi Jin, Akshay Krishnamurthy, and Sobhan Miryoosefi (2022). “Provable Reinforcement Learning with a Short-Term Memory”. In: *ICML*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 5832–5850.

Even-Dar, Eyal, Sham M. Kakade, and Yishay Mansour (2007). “The Value of Observation for Monitoring Dynamic Systems”. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007*, pp. 2474–2479.

Fiechter, Claude-Nicolas (1994). “Efficient Reinforcement Learning”. In: *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory, COLT 1994*. ACM, pp. 88–97.

François-Lavet, Vincent, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau (2018). “An Introduction to Deep Reinforcement Learning”. In:

Foundations and Trends in Machine Learning 11.3-4, pp. 219–354. ISSN: 1935-8237. DOI: 10.1561/2200000071.

Furelos-Blanco, Daniel, Mark Law, Anders Jonsson, Krysia Broda, and Alessandra Russo (2022). “Hierarchies of Reward Machines”. In: *CoRR* abs/2205.15752.

Gabbianelli, Germano, Gergely Neu, Nneka Okolo, and Matteo Papini (2023). “Offline Primal-Dual Reinforcement Learning for Linear MDPs”. In: *CoRR* abs/2305.12944. DOI: 10.48550/ARXIV.2305.12944. arXiv: 2305.12944.

Gao, Yang and Francesca Toni (2015). “Potential Based Reward Shaping for Hierarchical Reinforcement Learning”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*. AAAI Press, pp. 3504–3510.

Gaon, Maor and Ronen I. Brafman (2019). “Reinforcement Learning with Non-Markovian Rewards”. arXiv: 1912.02552 [cs].

García, Javier, Álvaro Visús, and Fernando Fernández (2022). “A Taxonomy for Similarity Metrics between Markov Decision Processes”. In: *Machine Learning* 111.11, pp. 4217–4247.

Garg, Sumegha, Pravesh K. Kothari, Pengda Liu, and Ran Raz (2021). “Memory-Sample Lower Bounds for Learning Parity with Noise”. In: *CoRR* abs/2107.02320. arXiv: 2107.02320.

Garg, Sumegha, Ran Raz, and Avishay Tal (2018). “Extractor-Based Time-Space Lower Bounds for Learning”. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*. ACM, pp. 990–1002. DOI: 10.1145/3188745.3188962.

Geffner, Hector and Blai Bonet (2013). *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers. ISBN: 978-1-60845-969-8. DOI: 10.2200/S00513ED1V01Y201306AIM022.

Golowich, Noah, Ankur Moitra, and Dhruv Rohatgi (2022a). “Learning in Observable POMDPs, without Computationally Intractable Oracles”. In: *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

— (2022b). “Planning in Observable POMDPs in Quasipolynomial Time”. In: *CoRR* abs/2201.04735. arXiv: 2201.04735.

Grzes, Marek (2017). “Reward Shaping in Episodic Reinforcement Learning”. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017*. ACM, pp. 565–573.

Guo, Hongyi, Qi Cai, Yufeng Zhang, Zhuoran Yang, and Zhaoran Wang (2022). “Provably Efficient Offline Reinforcement Learning for Partially Observable Markov Decision Processes”. In: *International Conference on Machine Learn-*

ing, ICML 2022. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 8016–8038.

Guo, Zhaohan Daniel, Shayan Doroudi, and Emma Brunskill (2016). "A PAC RL Algorithm for Episodic POMDPs". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*. Vol. 51. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 510–518.

Gürtler, Nico, Dieter Büchler, and Georg Martius (2021). "Hierarchical Reinforcement Learning with Timed Subgoals". In: *CoRR* abs/2112.03100.

Ha, David and Jürgen Schmidhuber (2018). "Recurrent World Models Facilitate Policy Evolution". In: *Advances in Neural Information Processing Systems 31, NeurIPS 2018*, pp. 2455–2467.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine (2018). "Soft Actor-Critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1856–1865.

Hansen, Eric A. (1998). "Solving POMDPs by Searching in Policy Space". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI '98*. Ed. by Gregory F. Cooper and Serafín Moral. Morgan Kaufmann, pp. 211–219.

Hausknecht, Matthew J. and Peter Stone (2015). "Deep Recurrent Q-learning for Partially Observable MDPs". In: *2015 AAAI Fall Symposia*. AAAI Press, pp. 29–37.

Hauskrecht, Milos (2000). "Value-Function Approximations for Partially Observable Markov Decision Processes". In: *Journal of Artificial Intelligence Research* 13, pp. 33–94. DOI: `10.1613/jair.678`.

Heess, Nicolas, Jonathan J. Hunt, Timothy P. Lillicrap, and David Silver (2015). "Memory-Based Control with Recurrent Neural Networks". In: *CoRR* abs/1512.04455.

Hessel, Matteo, Joseph Modayil, Hado van Hasselt, Tom Schaul, et al. (2018). "Rainbow: Combining Improvements in Deep Reinforcement Learning". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1.

Howard, Ronald A. (1960). *Dynamic Programming and Markov Processes*. Dynamic Programming and Markov Processes. John Wiley, pp. viii, 136. viii, 136.

Hsu, David, Wee Sun Lee, and Nan Rong (2007). "What Makes Some POMDP Problems Easy to Approximate?" In: *Advances in Neural Information Processing Systems 20, NIPS 2007*. Curran Associates, Inc., pp. 689–696.

Hutsebaut-Buysse, Matthias, Kevin Mets, and Steven Latré (2022). "Hierarchical Reinforcement Learning: A Survey and Open Research Challenges". In: *Machine Learning and Knowledge Extraction* 4.1, pp. 172–221. ISSN: 2504-4990.

Hutter, Marcus (2009). "Feature Reinforcement Learning: Part I. Unstructured MDPs". In: *J. Artif. Gen. Intell.* 1.1, pp. 3–24.

— (2014). "Extreme State Aggregation beyond MDPs". In: *Algorithmic Learning Theory - 25th International Conference, ALT 2014*. Vol. 8776. Lecture Notes in Computer Science. Springer, pp. 185–199. DOI: `10.1007/978-3-319-11662-4\_14`.

— (2016). "Extreme State Aggregation beyond Markov Decision Processes". In: *Theor. Comput. Sci.* 650, pp. 73–91. DOI: `10.1016/j.tcs.2016.07.032`.

Icarte, Rodrigo Toro, Toryn Klassen, Richard Valenzano, and Sheila McIlraith (2018). "Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2107–2116.

Icarte, Rodrigo Toro, Toryn Q. Klassen, Richard Anthony Valenzano, and Sheila A. McIlraith (2022). "Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning". In: *Journal of Artificial Intelligence Research* 73, pp. 173–208.

Icarte, Rodrigo Toro, Ethan Waldie, Toryn Q. Klassen, Richard Anthony Valenzano, Margarita P. Castro, and Sheila A. McIlraith (2019). "Learning Reward Machines for Partially Observable Reinforcement Learning". In: *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pp. 15497–15508.

Igl, Maximilian, Luisa M. Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson (2018). "Deep Variational Reinforcement Learning for POMDPs". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2122–2131.

Infante, Guillermo, Anders Jonsson, and Vicenç Gómez (2022). "Globally Optimal Hierarchical Reinforcement Learning for Linearly-Solvable Markov Decision Processes". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. AAAI Press, pp. 6970–6977.

Jaksch, Thomas, Ronald Ortner, and Peter Auer (2010). "Near-Optimal Regret Bounds for Reinforcement Learning". In: *Journal of Machine Learning Research* 11, pp. 1563–1600.

James, Michael R. and Satinder Singh (2004). "Learning and Discovery of Predictive State Representations in Dynamical Systems with Reset". In: *Machine Learning, Proceedings of the Twenty-First International Conference, ICML 2004*. Vol. 69.

ACM International Conference Proceeding Series. ACM. DOI: `10.1145/1015330.1015359`.

Jiang, Yiding, Shixiang Gu, Kevin Murphy, and Chelsea Finn (2019). “Language as an Abstraction for Hierarchical Deep Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9414–9426.

Jiang, Yiding, Evan Zheran Liu, Benjamin Eysenbach, J. Zico Kolter, and Chelsea Finn (2022). “Learning Options via Compression”. In: *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

Jin, Chi, Sham M. Kakade, Akshay Krishnamurthy, and Qinghua Liu (2020). “Sample-Efficient Reinforcement Learning of Undercomplete POMDPs”. In: *Advances in Neural Information Processing Systems 33, NeurIPS 2020*.

Jin, Chi, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan (2020). “Provably Efficient Reinforcement Learning with Linear Function Approximation”. In: *Conference on Learning Theory, COLT 2020*. Vol. 125. Proceedings of Machine Learning Research. PMLR, pp. 2137–2143.

Jin, Ying, Zhuoran Yang, and Zhaoran Wang (2021). “Is Pessimism Provably Efficient for Offline RL?” In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5084–5096.

Jinnai, Yuu, Jee Won Park, David Abel, and George Dimitri Konidaris (2019). “Discovering Options for Exploration by Minimizing Cover Time”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 3130–3139.

Jinnai, Yuu, Jee Won Park, Marlos C. Machado, and George Dimitri Konidaris (2020). “Exploration in Reinforcement Learning with Deep Covering Options”. In: *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Jong, Nicholas K., Todd Hester, and Peter Stone (2008). “The Utility of Temporal Abstraction in Reinforcement Learning”. In: *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*. IFAAMAS, pp. 299–306.

Jong, Nicholas K. and Peter Stone (2008). “Hierarchical Model-Based Reinforcement Learning: R-Max + MAXQ”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Association for Computing Machinery, pp. 432–439. ISBN: 978-1-60558-205-4.

Jothimurugan, Kishor, Osbert Bastani, and Rajeev Alur (2021). “Abstract Value Iteration for Hierarchical Reinforcement Learning”. In: *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021*. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1162–1170.

Kakade, Sham Machandranath (2013). “On the Sample Complexity of Reinforcement Learning”.

Kallus, Nathan and Masatoshi Uehara (2020). “Double Reinforcement Learning for Efficient Off-Policy Evaluation in Markov Decision Processes”. In: *Journal of Machine Learning Research* 21, 167:1–167:63.

Kapturowski, Steven, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney (2019). “Recurrent Experience Replay in Distributed Reinforcement Learning”. In: *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Kara, Ali Devran and Serdar Yüksel (2022). “Near Optimality of Finite Memory Feedback Policies in Partially Observed Markov Decision Processes”. In: *Journal of Machine Learning Research* 23, 11:1–11:46.

Kearns, Michael J. and Satinder Singh (2002). “Near-Optimal Reinforcement Learning in Polynomial Time”. In: *Machine Learning* 49.2-3, pp. 209–232.

Kempka, Michal, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaskowski (2016). “ViZDoom: A Doom-based AI Research Platform for Visual Reinforcement Learning”. In: *IEEE Conference on Computational Intelligence and Games, CIG 2016*. IEEE, pp. 1–8. DOI: 10.1109/CIG.2016.7860433.

Khetarpal, Khimya, Martin Klissarov, Maxime Chevalier-Boisvert, Pierre-Luc Bacon, and Doina Precup (2020). “Options of Interest: Temporal Abstraction with Interest Functions”. In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*. AAAI Press, pp. 4444–4451.

Koenig, Sven and Reid G. Simmons (1996). “The Effect of Representation and Knowledge on Goal-Directed Exploration with Reinforcement-Learning Algorithms”. In: *Machine Learning* 22.1, pp. 227–250. ISSN: 1573-0565.

Konidaris, George (2019). “On the Necessity of Abstraction”. In: *Current Opinion in Behavioral Sciences* 29, pp. 1–7. ISSN: 2352-1546. DOI: 10.1016/j.cobeha.2018.11.005.

Konidaris, George Dimitri, Leslie Pack Kaelbling, and Tomás Lozano-Pérez (2018). “From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning”. In: *Journal of Artificial Intelligence Research* 61, pp. 215–289.

Krishnamurthy, Akshay, Alekh Agarwal, and John Langford (2016). “PAC Reinforcement Learning with Rich Observations”. In: *NIPS*, pp. 1840–1848.

Kulesza, Alex, Nan Jiang, and Satinder Singh (2015). "Spectral Learning of Predictive State Representations with Insufficient Statistics". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 2715–2721. DOI: 10.1609/AAAI.V29I1.9635.

Lample, Guillaume and Devendra Singh Chaplot (2016). "Playing FPS Games with Deep Reinforcement Learning". In: *CoRR* abs/1609.05521. arXiv: 1609.05521.

— (2017). "Playing FPS Games with Deep Reinforcement Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 2140–2146.

Lattimore, Tor, Marcus Hutter, and Peter Sunehag (2013). "The Sample-Complexity of General Reinforcement Learning". In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 28–36.

Lattimore, Tor and Csaba Szepesvári (2020). *Bandit Algorithms*. Cambridge University Press. ISBN: 978-1-108-57140-1.

Lee, Junkyu, Michael Katz, Don Joven Agravante, Miao Liu, et al. (2022). "AI Planning Annotation for Sample Efficient Reinforcement Learning". In: *CoRR* abs/2203.00669.

Lee, Seungjae, Jigang Kim, Inkyu Jang, and H. Jin Kim (2022). "DHRL: A Graph-Based Approach for Long-Horizon and Sparse Hierarchical Reinforcement Learning". In: *Advances in Neural Information Processing Systems 35, NeurIPS 2022*.

Levy, Kfir Y. and Nahum Shimkin (2011). "Unified Inter and Intra Options Learning Using Policy Gradient Methods". In: *Recent Advances in Reinforcement Learning - 9th European Workshop, EWRL 2011*. Vol. 7188. Lecture Notes in Computer Science. Springer, pp. 153–164.

Li, Gen, Laixi Shi, Yuxin Chen, Yuejie Chi, and Yuting Wei (2022). "Settling the Sample Complexity of Model-Based Offline Reinforcement Learning". In: *CoRR* abs/2204.05275. DOI: 10.48550/ARXIV.2204.05275. arXiv: 2204.05275.

Li, Lihong, Thomas J. Walsh, and Michael L. Littman (2006). "Towards a Unified Theory of State Abstraction for MDPs". In: *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2006*.

Li, Xiujun, Lihong Li, Jianfeng Gao, Xiaodong He, et al. (2015). "Recurrent Reinforcement Learning: A Hybrid Approach". In: *CoRR* abs/1509.03044. arXiv: 1509.03044.

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, et al. (2016). "Continuous Control with Deep Reinforcement Learning". In: *4th International Conference on Learning Representations, ICLR 2016*.

Lin, Long-Ji and Tom M. Mitchell (1993). “Reinforcement Learning with Hidden States”. In: *Proceedings of the Second International Conference on from Animals to Animats*. MIT Press, pp. 271–280. ISBN: 0-262-63149-0.

Littman, Michael L., Thomas L. Dean, and Leslie Pack Kaelbling (1995). “On the Complexity of Solving Markov Decision Problems”. In: *UAI ’95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, pp. 394–402.

Littman, Michael L., Richard S. Sutton, and Satinder P. Singh (2001). “Predictive Representations of State”. In: *Advances in Neural Information Processing Systems 14, NIPS 2001*. MIT Press, pp. 1555–1561.

Liu, Qinghua, Alan Chung, Csaba Szepesvári, and Chi Jin (2022). “When Is Partially Observable Reinforcement Learning Not Scary?” In: *Conference on Learning Theory, COLT*. Vol. 178. Proceedings of Machine Learning Research. PMLR, pp. 5175–5220.

Lusena, Christopher, Judy Goldsmith, and Martin Mundhenk (2001). “Nonapproximability Results for Partially Observable Markov Decision Processes”. In: *Journal of Artificial Intelligence Research* 14, pp. 83–103. DOI: 10.1613/jair.714.

Machado, Marlos C., André Barreto, Doina Precup, and Michael Bowling (2023). “Temporal Abstraction in Reinforcement Learning with the Successor Representation”. In: *Journal of Machine Learning Research* 24, 80:1–80:69.

Machado, Marlos C., Marc G. Bellemare, and Michael H. Bowling (2017). “A Laplacian Framework for Option Discovery in Reinforcement Learning”. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 2295–2304.

Madani, Omid, Steve Hanks, and Anne Condon (1999). “On the Undecidability of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems”. In: *AAAI/IAAI*. AAAI Press / The MIT Press, pp. 541–548.

Maei, Hamid Reza, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S. Sutton (2010). “Toward Off-Policy Learning Control with Function Approximation”. In: *Proceedings of the 27th International Conference on Machine Learning, ICML-10*. Omnipress, pp. 719–726.

Mahadevan, Sridhar, Bo Liu, Philip S. Thomas, William Dabney, et al. (2014). “Proximal Reinforcement Learning: A New Theory of Sequential Decision Making in Primal-Dual Spaces”. In: *CoRR* abs/1405.6757. arXiv: 1405.6757.

Mahmud, M. M. Hassan (2010). “Constructing States for Reinforcement Learning”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*. Omnipress, pp. 727–734.

Maillard, Odalric-Ambrym, Rémi Munos, and Daniil Ryabko (2011). "Selecting the State-Representation in Reinforcement Learning". In: *Advances in Neural Information Processing Systems 24, NIPS 2011*, pp. 2627–2635.

Maillard, Odalric-Ambrym, Phuong Nguyen, Ronald Ortner, and Daniil Ryabko (2013). "Optimal Regret Bounds for Selecting the State Representation in Reinforcement Learning". In: *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 543–551.

Majeed, Sultan Javed and Marcus Hutter (2018). "On Q-learning Convergence for Non-Markov Decision Processes". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*. Ed. by Jérôme Lang. ijcai.org, pp. 2546–2552. DOI: 10.24963/IJCAI.2018/353.

Maurer, Andreas and Massimiliano Pontil (2009). "Empirical Bernstein Bounds and Sample-Variance Penalization". In: *The 22nd Conference on Learning Theory, COLT 2009*.

McCallum, Andrew Kachites (1996). "Reinforcement Learning with Selective Perception and Hidden State". PhD thesis. University of Rochester.

Mirowski, Piotr, Razvan Pascanu, Fabio Viola, Hubert Soyer, et al. (2017). "Learning to Navigate in Complex Environments". In: *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, et al. (2016). "Asynchronous Methods for Deep Reinforcement Learning". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1928–1937.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, et al. (2015). "Human-Level Control through Deep Reinforcement Learning". In: *Nature* 518.7540, pp. 529–533. ISSN: 14764687.

Mundhenk, Martin (2000). "The Complexity of Optimal Small Policies". In: *Mathematics of Operations Research* 25.1, pp. 118–129. DOI: 10.1287/moor.25.1.118.15214.

Murphy, K.P. (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series. MIT Press. ISBN: 978-0-262-01802-9.

Nachum, Ofir, Shixiang Gu, Honglak Lee, and Sergey Levine (2018). "Data-Efficient Hierarchical Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc.

— (2019). "Near-Optimal Representation Learning for Hierarchical Reinforcement Learning". In: *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

Neu, Gergely and Nneka Okolo (2023). “Efficient Global Planning in Large MDPs via Stochastic Primal-Dual Optimization”. In: *International Conference on Algorithmic Learning Theory*. Vol. 201. Proceedings of Machine Learning Research. PMLR, pp. 1101–1123.

Ng, Andrew Y., Daishi Harada, and Stuart J. Russell (1999). “Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping”. In: *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)*. Morgan Kaufmann, pp. 278–287.

Nguyen, Phuong, Odalric-Ambrym Maillard, Daniil Ryabko, and Ronald Ortner (2013). “Competing with an Infinite Set of Models in Reinforcement Learning”. In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013*. Vol. 31. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 463–471.

Oh, Junhyuk, Valliappa Chockalingam, Satinder P. Singh, and Honglak Lee (2016). “Control of Memory, Active Perception, and Action in Minecraft”. In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 2790–2799.

Ortner, Ronald, Matteo Pirotta, Alessandro Lazaric, Ronan Fruit, and Odalric-Ambrym Maillard (2019). “Regret Bounds for Learning State Representations in Reinforcement Learning”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.

Palmer, Nick and Paul W. Goldberg (2007). “PAC-learnability of Probabilistic Deterministic Finite State Automata in Terms of Variation Distance”. In: *Theoretical Computer Science* 387.1, pp. 18–31. DOI: 10.1016/J.TCS.2007.07.023.

Papadimitriou, Christos H. and John N. Tsitsiklis (1987). “The Complexity of Markov Decision Processes”. In: *Mathematics of Operations Research* 12.3, pp. 441–450.

Pineau, Joelle, Geoffrey J. Gordon, and Sebastian Thrun (2003). “Point-Based Value Iteration: An Anytime Algorithm for POMDPs”. In: *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI-03*. Ed. by Georg Gottlob and Toby Walsh. Morgan Kaufmann, pp. 1025–1032.

Pohlen, Tobias, Bilal Piot, Todd Hester, Mohammad Gheshlaghi Azar, et al. (2018). “Observe and Look Further: Achieving Consistent Performance on Atari”. In: *CoRR* abs/1805.11593. arXiv: 1805.11593.

Precup, Doina and Richard S. Sutton (1997). “Multi-Time Models for Temporally Abstract Planning”. In: *Advances in Neural Information Processing Systems 10, NIPS 1997*. The MIT Press, pp. 1050–1056.

Puterman, Martin L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley Series in Probability and Statistics. Wiley. ISBN: 978-0-471-61977-2.

Rabin, Michael O. and Dana S. Scott (1959). “Finite Automata and Their Decision Problems”. In: *IBM J. Res. Dev.* 3.2, pp. 114–125.

Rashidinejad, Paria, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell (2021). “Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism”. In: *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 11702–11716.

Ravindran, Balaraman and Andrew G Barto (2004). “Approximate Homomorphisms: A Framework for Non-Exact Minimization in Markov Decision Processes”. In: p. 10.

— (2002). “Model Minimization in Hierarchical Reinforcement Learning”. In: *Abstraction, Reformulation and Approximation, 5th International Symposium, SARA 2002.* Vol. 2371. Lecture Notes in Computer Science. Springer, pp. 196–211.

— (2003). “Relativized Options: Choosing the Right Transformation”. In: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003).* AAAI Press, pp. 608–615.

Ren, Tongzheng, Jialian Li, Bo Dai, Simon S. Du, and Sujay Sanghavi (2021). “Nearly Horizon-Free Offline Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 15621–15634.

Rissanen, Jorma (1983). “A Universal Data Compression System”. In: *IEEE Trans. Inf. Theory* 29.5, pp. 656–663. DOI: `10.1109/TIT.1983.1056741`.

Ron, Dana, Yoram Singer, and Naftali Tishby (1996). “The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length”. In: *Machine Learning* 25.2-3, pp. 117–149. DOI: `10.1023/A:1026490906255`.

— (1998). “On the Learnability and Usage of Acyclic Probabilistic Finite Automata”. In: *Journal of Computer and System Sciences* 56.2, pp. 133–152. DOI: `10.1006/JCSS.1997.1555`.

Ronca, Alessandro and Giuseppe De Giacomo (2021). “Efficient PAC Reinforcement Learning in Regular Decision Processes”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021.* ijcai.org, pp. 2026–2032.

Ronca, Alessandro, Gabriel Paludo Licks, and Giuseppe De Giacomo (2022). “Markov Abstractions for PAC Reinforcement Learning in Non-Markov Decision Processes”. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022.* ijcai.org, pp. 3408–3415.

Ross, Keith Wimberly (1985). *Constrained Markov Decision Processes with Queueing Applications.* University of Michigan.

Russell, Stuart and Peter Norvig (2009). *Artificial Intelligence: A Modern Approach.* 3rd. Prentice Hall Press. ISBN: 0-13-604259-7.

Schubert, Ingmar, Ozgur S. Oguz, and Marc Toussaint (2021). "Plan-Based Relaxed Reward Shaping for Goal-Directed Tasks". In: *9th International Conference on Learning Representations, ICLR 2021.* OpenReview.net.

Schulman, John, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz (2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015.* Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1889–1897.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal Policy Optimization Algorithms". In: *CoRR* abs/1707.06347.

Silver, David and Joel Veness (2010). "Monte-Carlo Planning in Large POMDPs". In: *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a Meeting Held 6-9 December 2010, Vancouver, British Columbia, Canada.* Curran Associates, Inc., pp. 2164–2172.

Simsek, Özgür and Andrew G. Barto (2004). "Using Relative Novelty to Identify Useful Temporal Abstractions in Reinforcement Learning". In: *Machine Learning, Proceedings of the Twenty-First International Conference (ICML 2004).* Vol. 69. ACM International Conference Proceeding Series. ACM.

Singh, Satinder P., Michael L. Littman, Nicholas K. Jong, David Pardoe, and Peter Stone (2003). "Learning Predictive State Representations". In: *Machine Learning, Proceedings of the Twentieth International Conference, ICML 2003.* AAAI Press, pp. 712–719.

Smallwood, Richard D. and Edward J. Sondik (1973). "The Optimal Control of Partially Observable Markov Processes over a Finite Horizon". In: *Operations Research* 21.5, pp. 1071–1088. DOI: 10.1287/opre.21.5.1071.

Sondik, Edward J. (1978). "The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs". In: *Operations Research* 26.2, pp. 282–304. ISSN: 0030364X, 15265463. JSTOR: 169635.

Steccanella, Lorenzo (2023). "Representation Learning for Hierarchical Reinforcement Learning". Universitat Pompeu Fabra. 147 pp.

Steccanella, Lorenzo, Simone Totaro, and Anders Jonsson (2021). "Hierarchical Representation Learning for Markov Decision Processes". In: *CoRR* abs/2106.01655.

Strehl, Alexander L., Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman (2006). "PAC Model-Free Reinforcement Learning". In: *Proceedings of* *the 23rd International Conference on Machine Learning - ICML '06*. The 23rd International Conference. ACM Press, pp. 881–888. ISBN: 978-1-59593-383-6.

Subramanian, Jayakumar, Amit Sinha, Raihan Seraj, and Aditya Mahajan (2022). "Approximate Information State for Approximate Planning and Reinforcement Learning in Partially Observed Systems". In: *Journal of Machine Learning Research* 23, 12:1–12:83.

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning Series. The MIT Press. 526 pp. ISBN: 978-0-262-03924-6.

Sutton, Richard S., Doina Precup, and Satinder Singh (1999). "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning". In: *Artificial Intelligence* 112.1-2, pp. 181–211.

Sutton, Richard S., Doina Precup, and Satinder P. Singh (1998). "Intra-Option Learning about Temporally Abstract Actions". In: *Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998*. Morgan Kaufmann, pp. 556–564.

Szörényi, Balázs (2009). "Characterizing Statistical Query Learning: Simplified Notions and Proofs". In: *Algorithmic Learning Theory, 20th International Conference, ALT 2009*. Vol. 5809. Lecture Notes in Computer Science. Springer, pp. 186–200. DOI: 10.1007/978-3-642-04414-4_18.

Tenenbaum, Joshua B., Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman (2011). "How to Grow a Mind: Statistics, Structure, and Abstraction". In: *Science (New York, N.Y.)* 331.6022, pp. 1279–1285.

Thomas, Philip S. and Emma Brunskill (2016). "Data-Efficient off-Policy Policy Evaluation for Reinforcement Learning". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 2139–2148.

Tiapkin, Daniil and Alexander V. Gasnikov (2022). "Primal-Dual Stochastic Mirror Descent for MDPs". In: *International Conference on Artificial Intelligence and Statistics, AISTATS 2022*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 9723–9740.

Uehara, Masatoshi, Chengchun Shi, and Nathan Kallus (2022). "A Review of Off-Policy Evaluation in Reinforcement Learning". In: *CoRR* abs/2212.06355. DOI: 10.48550/ARXIV.2212.06355. arXiv: 2212.06355.

Uehara, Masatoshi and Wen Sun (2022). "Pessimistic Model-Based Offline Reinforcement Learning under Partial Coverage". In: *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Uehara, Masatoshi, Xuezhou Zhang, and Wen Sun (2022). “Representation Learning for Online and Offline RL in Low-Rank MDPs”. In: *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.net.

Van Hasselt, Hado, Arthur Guez, Matteo Hessel, Volodymyr Mnih, and David Silver (2016). “Learning Values across Many Orders of Magnitude”. In: *NIPS*, pp. 4287–4295.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, et al. (2017). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc.

Veness, Joel, Kee Siong Ng, Marcus Hutter, William T. B. Uther, and David Silver (2011). “A Monte-Carlo AIXI Approximation”. In: *Journal of Artificial Intelligence Research* 40, pp. 95–142. DOI: 10.1613/JAIR.3125.

Verdú, Sergio (2014). “Total Variation Distance and the Distribution of Relative Information”. In: *2014 Information Theory and Applications Workshop, ITA 2014*. IEEE, pp. 1–3. DOI: 10.1109/ITA.2014.6804281.

Wang, Ziyu, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Frcitas (2016). “Dueling Network Architectures for Deep Reinforcement Learning”. In: *33rd International Conference on Machine Learning, ICML 2016* 4.9, pp. 2939–2947.

Watkins, Christopher J. C. H. and Peter Dayan (1992). “Technical Note Q-learning”. In: *Machine Learning* 8, pp. 279–292.

Wen, Zheng, Doina Precup, Morteza Ibrahimi, André Barreto, Benjamin Van Roy, and Satinder Singh (2020). “On Efficiency in Hierarchical Reinforcement Learning”. In: *Advances in Neural Information Processing Systems 33, NeurIPS 2020*.

Wierstra, Daan, Alexander Förster, Jan Peters, and Jürgen Schmidhuber (2007). “Solving Deep Memory POMDPs with Recurrent Policy Gradients”. In: *Artificial Neural Networks, ICANN 2007*. Vol. 4668. Lecture Notes in Computer Science. Springer, pp. 697–706. DOI: 10.1007/978-3-540-74690-4_71.

Wiewiora, E. (2003). “Potential-Based Shaping and Q-Value Initialization Are Equivalent”. In: *Journal of Artificial Intelligence Research* 19, pp. 205–208. ISSN: 1076-9757.

Wiewiora, Eric, Garrison W. Cottrell, and Charles Elkan (2003). “Principled Methods for Advising Reinforcement Learning Agents”. In: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*. AAAI Press, pp. 792–799.

Xie, Tengyang, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai (2021). “Policy Finetuning: Bridging Sample-Efficient Offline and Online Reinforcement Learning". In: *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 27395–27407.

Xu, Zhe, Ivan Gavran, Yousef Ahmad, Rupak Majumdar, et al. (2020). "Joint Inference of Reward Machines and Policies for Reinforcement Learning". In: *Proceedings of the International Conference on Automated Planning and Scheduling.* Vol. 30, pp. 590–598.

Yang, Lin and Mengdi Wang (2019). "Sample-Optimal Parametric Q-learning Using Linearly Additive Features". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019.* Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 6995–7004.

Ye, Yinyu (2011). "The Simplex and Policy-Iteration Methods Are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate". In: *Mathematics of Operations Research* 36.4, pp. 593–603.

Yin, Ming and Yu-Xiang Wang (2021). "Towards Instance-Optimal Offline Reinforcement Learning with Pessimism". In: *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 4065–4078.

Yu, Huizhen and Dimitri P. Bertsekas (2008). "On near Optimality of the Set of Finite-State Controllers for Average Cost POMDP". In: *Mathematics of Operations Research* 33.1, pp. 1–11. DOI: `10.1287/moor.1070.0279`.

Zhan, Wenhao, Baihe Huang, Audrey Huang, Nan Jiang, and Jason D. Lee (2022). "Offline Reinforcement Learning with Realizability and Single-Policy Concentrability". In: *Conference on Learning Theory.* Vol. 178. Proceedings of Machine Learning Research. PMLR, pp. 2730–2775.

Zhan, Wenhao, Masatoshi Uehara, Wen Sun, and Jason D. Lee (2023). "PAC Reinforcement Learning for Predictive State Representations". In: *The Eleventh International Conference on Learning Representations, ICLR 2023.* OpenReview.net.

Zhang, Nevin Lianwen, Stephen S. Lee, and Weihong Zhang (1999). "A Method for Speeding up Value Iteration in Partially Observable Markov Decision Processes". In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI '99.* Morgan Kaufmann, pp. 696–703.

Zhang, Yiming, Quan Vuong, and Keith W. Ross (2020). "First Order Constrained Optimization in Policy Space". In: *Advances in Neural Information Processing Systems 33, NeurIPS 2020.*

Zhang, Zongzhang, Michael L. Littman, and Xiaoping Chen (2012). "Covering Number as a Complexity Measure for POMDP Planning and Learning". In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2012.* Ed. by Jörg Hoffmann and Bart Selman. AAAI Press.

Nomenclature

AI	Artificial Intelligence, page 3
DFA	Deterministic Finite Automaton, page 17
HRL	Hierarchical Reinforcement Learning, page 27
k-MDP	k-Markov Decision Process, page 18
LP	Linear Programming, page 74
MDP	Markov Decision Process, page 18
ML	Machine Learning, page 4
NMDP	Non-Markov Decision Process, page 18
NN	Neural Networks, page 24
PAC	Probably Approximately Correct, page 16
PDFA	Probabilistic Deterministic Finite Automaton, page 131
PI	Policy Iteration, page 21
POMDP	Partially Observable Markov Decision Process, page 19
PSR	Predictive State Representation, page 133
RDP	Regular Decision Process, page 19
RL	Reinforcement Learning, page 15
RM	Reward Machine, page 101
RNN	Recurrent Neural Network, page 134
RS	Reward Shaping, page 37
SDM	Sequential Decision-Making, page 3
VI	Value Iteration, page 21