Numéro d ordre : 4672

THÈSE

PÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

Par **Youssouf OUALHADJ**

POUR L'OBTENTION DU GRADE DE

DOCTEUR

SPÉCIALITÉ : **INFORMATIQUE**

Le problème de la valeur dans les jeux stochastiques

Soutenue publiquement le : 11 décembre 2012

Après avis des rapporteurs :

MME Christel Baier	Professeur, Technische Universität Dresden
M. Jean-François Raskin	Professeur, Université Libre de Bruxelles

Devant la commission d examen composée de :

MME Christel Baier	Professeur, Technische Universität Dresden	Rapporteur
MME Nathalie Bertrand	CR INRIA, INRIA Rennes Bretagne Atlantique	Examinateur
M. Hugo Gimbert	CR CNRS, LaBRI, Université Bordeaux 1	Directeur de thèse
M. Jean Mairesse	DR CNRS, LIAFA, Université Paris 7	Président du jury
MME Anca Muscholl	Professeur, LaBRI, Université Bordeaux 1	Directrice de thèse
M. Jean-François Raskin	Professeur, Université Libre de Bruxelles	Rapporteur
M. Luigi Santocanale	Professeur, LIF, Aix Marseille Université	Examinateur
M. Marc Zeitoun	Professeur, LaBRI, Université Bordeaux 1	Examinateur

à ma famille …

Remerciements

Je tiens à remercier mes directeurs Hugo Gimbert et Anca Muscholl pour leur soutient tout au long de cette thèse et bien avant au moment où j'étais étudiant en master. Anca m'a transmis son gout pour la synthèse de contrôleur alors qu'Hugo m'a transmis sa passion pour les probabilités. Ils m'ont patiemment guidé à travers les sujets de recherches qui m'intéressaient en me faisant confiance. Avec eux j'ai redécouvert la notion de preuve, le pouvoir d'un exemple bien choisi et la nécessité de faire au plus simple. J'espère avoir acquis un peu de leur rigueur scientifique, ainsi que leurs enthousiasme pour la recherche. Bien sur je n'oublie pas de les remercier pour leur soutient logistique et matériels.

Je voudrais aussi remercier mes rapporteurs Christel Baier et Jean-François Raskin, ils ont accepté de prendre sur leur temps – bien qu'étant très occupés – pour relire et commenter mes travaux. Je voudrais aussi les remercier pour leurs ponctualité au moment de rendre les rapports.

Jean Mairesse a accepté de présider mon jury de thèse et cela me fait très plaisir. Je n'oublie pas non plus les autre membres qui ont accepté d'examiner cette thèse; Nathalie Bertrand, Luigi Santocanale et Marc Zeitoun que je remercie doublement car il m'a donné l'envie de faire de l'informatique théorique au moment ou j'ai suivi le cours calculabilité.

Cette thèse a était réalisée au seins de l'équipe Méthodes Formelles au LaBRI dont j'ai eu le plaisir de faire partie et je remercie les membre de cette équipe pour leur accueil amical et sientifiquement enrichissant. Je remercie aussi ces thésards Srivathsan, Alexender, Marc avec qui les discussion n'étaient pas toujours scientifiques mais n'en n'étaient pas moins intéressantes.

Au moment où j'ai soumis ce manuscrit j'étais déjà en postdoc à Marseille au seins de l'équipe MoVe du LIF, cette équipe m'a chaleureusement accueilli. Grâce à Pierre-Alain Reynier et au projet ECSPER, j'ai pu vivre ma première expérience "postdoctorale" bien qu'étant officiellement en vie "predoctorale". Je remercie aussi Jean-Marc Talbot pour les discussions amusantes et sujets de recherche stimulants, Arnaud Labourel ainsi que la petite équipe de thésards Florent et Mathieu.

Avant d'en finir avec le volé scientifique, je voudrais enfin remercié Krishnendu Chatterjee pour son accueil de 3 semaines à l'IST période durant laquelle les résultats du Chapitre 4 ont été finalisés. Je remercie Nathanaël Fijalkow pour les discussions et les directions de recherches, ainsi que Laurent Doyen et Soumya Paul.

Je remercie chaleureusement l'équipe administrative du LaBRI et particulièrement, Lebna, Brigitte et Philippe pour leur gentillesse, disponibilité et surtout leur réactivité. Sans leur aide je ne pense pas que j'aurais réussi à me sortir des rouages de l'administration.

Bien sur, une thèse ne peut être réalisé sans le support et les encouragement des amis et des proches. Je remercie Natalia pour ces conseils et son calme, Gaël pour les parties de tennis qui ont duré des heures et des heures et qui m'ont permis d'évacuer la frustration des théorèmes non prouvés sans oublier Florent, Petru, Sri pour les parties de pingpong. Enfin cela va sans dire que sans l'aide, la confiance, les encouragements de mes parents, frères et belles sœurs, jamais je n'aurais même envisagé de faire une thèse, je voudrais aussi les remercier pour avoir organisé le pot qui fut réussite.

Résumé

La théorie des jeux est un outils standard quand il s agit de l étude des systèmes réactifs. Ceci est une conséquence de la variété des modèle de jeux tant au niveau de l interaction des joueurs qu au niveau de l information que chaque joueur possède.

Dans cette thèse, on étudie le problème de la valeur pour des jeux où les joueurs possèdent une information parfaite, information partiel et aucune information.

Dans le cas où les joueurs possèdent une information parfaite sur l état du jeu, on étudie le problème de la valeur pour des jeux dont les objectifs sont des combinaisons booléennes d objectifs qualitatifs et quantitatifs.

Pour les jeux stochastiques à un joueur, on montre que les valeurs sont calculables en temps polynomiale et on montre que les stratégies optimales peuvent être implementées avec une mémoire finie.

On montre aussi que notre construction pour la conjonction de parité et de la moyenne positive peut être étendue au cadre des jeux stochastiques à deux joueurs.

Dans le cas où les joueurs ont une information partielle, on étudie le problème de la valeur pour la condition d accessibilité.

On montre que le calcul de l ensemble des états à valeur 1 est un problème indécidable, on introduit une sous classe pour laquelle ce problème est décidable. Le problème de la valeur 1 pour cette sous classe est PSPACE-complet dans le cas de joueur aveugle et dans EXPTIME dans le cas de joueur avec observations partielles.

Mots clés: Théorie des jeux, jeux stochastiques, automates, synthèses de contrôleur, vérification quantitative.

Abstract

Game theory proved to be very useful in the field of verification of open reactive systems. This is due to the wide variety of games model that differ in the way players interact and the amount of information players have.

In this thesis, we study the value problem for games where players have full knowledge on their current configuration of the game, partial knowledge, and no knowledge.

In the case where players have perfect information, we study the value problem for objectives that consist in combination of qualitative and quantitative conditions.

In the case of one player stochastic games, we show that the values are computable in polynomial time and show that the optimal strategies exist and can be implemented with finite memory.

We also showed that our construction for parity and positive-average Markov decision processes extends to the case of two-player stochastic games.

In the case where the players have partial information, we study the value problem for reachability objectives.

We show that computing the set of states with value 1 is an undecidable problem and introduce a decidable subclass for the value 1 problem. This sub class is PSPACE-complete in the case of blind controllers and EXPTIME is the setting of games with partial observations.

Key words: Games theory, stochastic games, automata theory, controller synthesis, quantitative verification.

Contents

Chapter 1

Introduction

Contents

1.1 Background

Game Theory Is the formal tool to study decision making. Quoting the Nobel prize laureate *Roger Myerson*, game theory is:

> *the study of mathematical models of conflict and cooperation between intelligent rational decision-makers* [Mye91].

Although the most notable motivation for game theory is economics [NM44, Nas50], games, from a theoretical point of view, can model a wide variety of behaviors which makes them one of the most versatile tool. Game theoretic models were successfully used in fields ranging from biology [Smi82] to political science [Mor94]. In result of these various applications, numerous models or type of games were introduced.

Games Model One of the first model of games studied is the so called *matrix games*, these are games played between two players with opposite objectives. Back in the 18th century, *James Waldegrave* discussed in a letter [OW96] the notion of what is now known as *minmax* solution to the two players version of the game *Le Her*[1]. In the beginning of the 20th century, the french mathematician *Émile Borel* [Bor21] defined the *normal form* of a game: a matrix representation of a game where each entry of the matrix specifies the amount of money the second player (called Min in this work) has to give to the first player (called Max). Borel introduced the notion of *mixed strategies* these are strategies that the players adopt where they use probabilities to make their decisions. He also conjectured the existence of mixed strategies that ensure a reward for Max no matter what strategy Min is using. Later on, *John von Neumann* proved the conjecture in his famous minmax theorem using Brouwer s fixed-point theorem [NM44].

Meanwhile, in 1913 Ernst Zermelo [Zer13] formalized the game of chess using games played on graphs[2] where the players play in turn and each one of them wants to reach a specific set of states, theses game are known as *reachability games*. Zermelo, formalized the notion of winning positions and attractor sets.

[1] Card game

[2] Each state corresponds to configuration is the chess board.

Another fundamental model is the one of *stochastic game with states* introduced by *Lloyd Shapley* [Sha53]. Shapley introduced a model where the rules are merging the previous models. In this new model, the players face each other in a sequence of matrix games. The current matrix game and the actions chosen by both players decide what is the next matrix game they will have to play.

Overview of the model In the present work, we study variations of the model introduced by Shapley where plays are either infinite or finite as in Zermelo s setting. Our goal is to design algorithms that compute the value of states for such games.

1.2 Context

Games and Algorithms Game theory in its early days was addressed by mathematicians and economists. Both communities were interested in problems such as the existence of different equilibria and strategies that achieve those equilibria. No work was fully dedicated to the algorithmic side of the topic. For instance, when Nash proved that every finite game has an equilibrium in mixed strategies, the result did not mention how hard is it to compute those strategies. The same can be said about the minmax theorem of von Neumann. The first algorithmic result on game theory was published by Lemke [LJ64] where an algorithm for computing Nash equilibria was designed. One of the reason that helped the development of the algorithmic side of game theory is its tight link with computer science, e.g. *automata theory* and *verification of open reactive systems.*

Automata Theory One of the most notorious result linking automata theory to games is the one of Rabin where he proves that the emptiness of automata over infinite tree is decidable using game theoretic techniques. A corollary of this result is the decidability of the monadic second order theory over infinite trees [Rab69]. Many other results use the fact that parity games are positionally determined[3] [EJ91, GH82, MS85, MS95, Zie98]. An other example is the one of Ehrenfeucht-Fraïssé games that allow to establish strict hierarchies between fragments of logics.

In general we often reduce automata theoretic problems to the problem of deciding the winner in a particular game. For instance solving the Mostowski hierarchy for parity automata over trees amounts to proving finite memory determinacy[4] for parity distance automata over trees [CL08].

In this thesis we inspire ourself from automata theoretic techniques and solve problem on games. In particular, we use the idea of iteration that appeared in [Sim90] to solve the limitedness problem for distance automata. These techniques are used in Chapter 6 and Chapter 7 to solve the Value 1 problem.

Verification and Control Synthesis In computer science we are often facing the problem of verifying whether a system, which is interacting with its environment, has the desired behavior or not. The following problem known as Church synthesis problem is the mathematical interpretation of the previous problem: *given an* Input/Output *relation, decide whether there exists a circuit that implements the relation.* In general, the system we want to verify or control is modeled by a game played over a finite graph between two players. The first player called Max is the player that represents the controller and the second player is called Min and represents the environment. The behavior we want the system to ensure, usually called specification, is given by a Borel set of plays. A strategy is a policy that tells player how to play. If Max has a strategy such that

[3] One of the players wins the game and both players can forget the past.

[4] One of the players wins the game and both can play using finite memory strategy.

against every strategy for Min the play belongs to the winning set, we say that Max has a winning strategy. A winning strategy models the policy our system should follow to ensure the specification. Moreover, if the winning strategy remembers only a finite information about the past, this strategy can be implemented by a finite automaton. Büchi Landweber theorem [BL69] provides a solution to the Church synthesis problems under the condition that the specification is regular. Later on this problem was proved to be decidable under different setting, always using game techniques.

Quantitative model checking The problem of model checking [Cla08, QS82] is the following: *given a model $\mathcal{M}$ and a specification φ, answer whether $\mathcal{M}$ satisfy φ.* This problem is decidable for many models and specification logics, it gives a qualitative answer. But when it comes to real life systems, we cannot always be in the ideal situation where a system is either fully correct or fully wrong, hence an alternative approach was developed and is now known as the quantitative model checking. The goal of this alternative approach is to quantify how good a system is with respect to some specification. For instance, consider the case of system that consists of requests and replies. A natural specification for such is system is: *for each request the system should send a reply.* A Quantitative formulation of this specification could be: *Maximize the probability that for each request the system should send a reply.* To study the quantitative model checking problem, we usually use the setting of stochastic models. These models can be seen as games where Max plays against a random player that generates his plays using a coin toss, or against both the random player and Min. In this setting, the goal of Max is to maximize the probability that the specification is achieved and the goal of Min is to minimize this same probability.

Qualitative against quantitative analysis When studying stochastic models one can distinguish between two approaches. The quantitative analysis refers to the computation of values of each state. As opposed to the qualitative approach which consists in partitioning the set of states into the set of states from where Max wins with probability 1 and the set of states from where Max wins with strictly positive probability.

Perfect or partial information? Another advantage of stochastic models, is that they can model situations where Max does not have a full knowledge of its environment which is even closer to a real life situation. In this thesis we study both settings. For the perfect information setting, we study specifications that consist of both regular and reward objectives. And for the partial information setting, we restrict our research to the simple reachability objective.

1.3 Outline and Contributions

1.3.1 Outline

This manuscript is divided into three parts. Part I consist of two chapters. In these chapters we review the basic concepts and results on probability theory, Markov chains, and Markov decision processes. Chapter 2 is where we introduce notations and theorems related to Markov chains. In Chapter 3 we enhance the model of Markov chains with a control power to obtain Markov decision processes. In the same chapter we review also the main tools we will use in the subsequent analysis.

Part II is dedicated to the setting of perfect information models. We study two models. The first one addressed in Chapter 4 is Markov decision processes where the controller has to ensure different objectives with same strategy in order to win. The second model is stochastic games. This model is essentially a Markov decision processes where the controller has to face a random player and Min. In Chapter 5, we study the algorithmic complexity of stochastic games with objectives that consist of conjunction of winning conditions. In Part III we return to the study of Markov decision process but this time, the controller has restricted knowledge about his environment. In Chapter 6 we study probabilistic automata. A probabilistic automaton is a Markov decision process where the controller cannot differentiate between states, hence the controller has to choose her moves in function of time elapsed. In Chapter 7 we study a slight variation that consists in enabling the controller to differentiate between subsets of states. In this last part we focus on reachability objectives. Figure 1.1 illustrates the evolution and relationships between chapters.

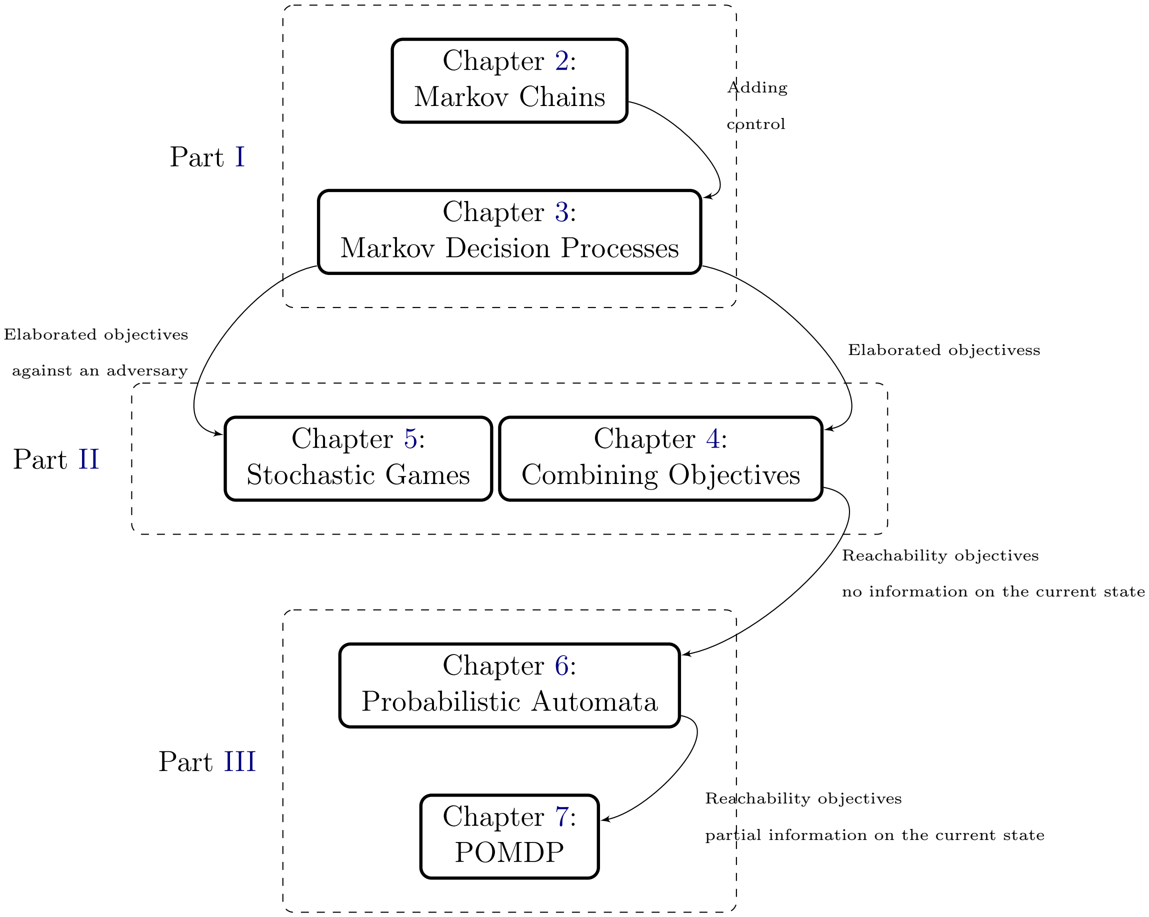


Figure 1.1: Map of the thesis

1.3.2 Contributions

In this thesis we study the value problem for stochastic models of both perfect information and partial information. In particular we will focus on algorithms that design strategies that ensure value 1. The reason we focus on this problem is because of in the setting of perfect information games with tail objectives, in order to compute the value of states, it is enough to compute the set of states with value 1.

In the case of partial information games, not only we focus on the value 1 problem but we consider even simpler objectives. We focus on reachability objectives, the reason we concentrate on such basic objective is that we believe that partial information games are not fully understood from an algorithmic point view and the undecidability barrier is easily reached.

Perfect information setting In Chapter 4 we study Markov decision processes where the goal of the controller is to satisfy combination of objectives. The first results obtained concern parity and positive average objectives. The positive-average objective asks the long term average payoff of a run to be strictly positive. The positive-average objective can be defined using to semantics that we shall call $\limsup$ and $\liminf$ semantics. We study combination of parity and positive-average with both semantics. When combining the parity and the positive-average objectives, one asks Max to maximize the probability that the two objectives are satisfied. The results obtained concerning this objective are as follows:

– A characterization of the almost-sure regions for Markov decision processes equipped with parity and positive-average objectives with $\limsup$(c.f. Lemmata 4.1 and 4.2).

– We give a polynomial time algorithm that computes this region, the correctness is established in Theorem 4.3.

– We study the complexity of optimal strategies and show that an exponential size memory is sufficient and necessary to implement optimal strategies with $\limsup$. (c.f. Theorem 4.5).

– The result on the size of the memory allows us to show that the objective parity and positive-average with $\liminf$ semantics can be solved using same algorithm and the memory requirements remain unchanged (c.f results of Section 4.4).

In order to solve boolean combinations of parity and positive-average objectives, we also studied objectives that consists of combination of different positive-average winning conditions (c.f. Section 4.5). Such a winning condition can be seen as a conjunction of positive-averages objectives. The result obtained are the following:

– We show in Theorem 4.16 how to solve a conjunction when all the objectives have the $\limsup$ semantics.

– In Theorem 4.18 we solve the conjunction when all the objectives have the $\liminf$ semantics.

– In Proposition 4.19 we solve the conjunction when the objectives are mixing $\liminf$ and $\limsup$ semantics.

The results of Chapter 4 is a joint work with Soumya Paul and were obtained independently from the work of Krishnendu Chatterjee et al [CD11, BBC+11]. In particular a different algorithm is presented in order to obtain Theorem 4.3. The advantage of our algorithm is that it can be easily extended to stochastic games, whereas the approach of [CD11] breaks in that setting.

This extension is presented in Chapter 5 where we study stochastic games with parity and positive-average objectives. We obtain the following results:

– In Section 5.3 we show that the problem of deciding whether Max has an almost-sure winning state is in NP (c.f. Theorem 5.26).

- We extend the algorithm presented in Chapter 4 to the case of stochastic games, the correctness is established in Theorem 5.27.

The result of Chapter 5 were obtained in join work with Krishnendu Chatterjee and Laurent Doyen and are prepared for submission.

Partial information setting In Chapter 6 we turn our attention to a yet another generalization of Markov decision processes that consists in hiding the information about the current states, hence the controller knows the description of the system and its initial configuration, but once the execution starts, the only knowledge of the controller is the time elapsing. This model correspond to the one introduced by Rabin in 1963 and called *Probabilistic automata*. The main motivation for studying probabilistic automata is to identify families of partial information games with computable values. The value of an automaton corresponds to the supremum probability that a word is accepted. Actually it is undecidable to decide whether a probabilistic automaton accepts some words with probability greater than $\frac{1}{2}$ or not [Paz71, MHC03a]. In Section 6.3 we focus on the emptiness problem for probabilistic automata and obtain the following results:

- we give an alternative version of the undecidability proof of the emptiness problem for probabilistic automata. The key point of this proof is the reduction to the equality problem 6.8 using the construction presented in Proposition 6.11.

- We show that the emptiness problem remains undecidable for automata with two probabilistic transitions (c.f. Proposition 6.15).

The other problem we tackle is the value problem. Although this problem is known to be undecidable [Ber74, BMT77] we focus on a special case, namely the value 1 problem where one is asked to decide whether a given automaton accepts words with probability arbitrarily close to 1. This problem remained open since the constructions presented by Bertoni in 1977 for the value problem excluded the value 1. The results obtained concerning this problem are as follows:

- We solve the value 1 problem and show that it is undecidable (c.f. Proposition 6.24).

- We show that the value 1 problem is undecidable even for automata with 1 probabilistic transition (c.f. Proposition 6.25).

- In Section 6.5, we identify a family of probabilistic automata (i.e $\sharp$-acyclic automata) for which this problem turns out to be decidable. In order to define this class, we first introduce an operation called iteration that abstract the behavior of the automaton when repeating an action an arbitrarily large number of time. Second, we define a graph called the *support graph* that abstracts the behavior of the automaton. The decidability result follows from the fact that the automaton has value 1 if and only a subset of the set of accepting states is reachable in the support graph.

The result of Chapter 6 were published in [GO10], and extended in a join work with Nathanaël Fijalkow [FGO12].

In Chapter 7, we use the knowledge we acquired about probabilistic automata and we tackle the value 1 problem for partially observable Markov decision processes. In this model, the information about the current state is still hidden but we assign to each state a color and the controller can only see the color of a state. Hence, if two states are colored the same way, the controller cannot differentiate between them. Our contribution consists in defining a family of partially observable

Markov decision processes for which the value 1 is decidable. In order to define this family we first generalize the operation of iteration introduced in Chapter 6 to the case of POMDP. We also generalized the support graph and call this new abstraction the *knowledge graph*. To get the decidability result, we construct for each POMDP a perfect information game played on the knowledge graph called the *knowledge game* and we show that the first player wins the knowledge game if and only if the POMDP has value 1.

The results of Chapter 7 were obtained recently and are now prepared for submission.

Part I

Prequel

Chapter 2

Markov Chains

Contents

In this chapter we recall basic notions and results of probability theory and Markov chain that we will use throughout this manuscript.

2.1 Basic Concepts

2.1.1 Events

Probability theory provides a mathematical framework for the study of random phenomena. When such a phenomenon occurs, we are interested in its outcome denoted ω. The collection of all possible outcomes is called the sample space Ω. For technical reasons, one considers only the collections of subsets of Ω that are closed under complementation, countable union and contain Ω. Such a collection is called a σ-field. Let $\mathcal{F}$ be a σ-field, the elements of $\mathcal{F}$ are called events, and the couple $(\Omega, \mathcal{F})$ is called a measurable space.

For example, tossing a die is a random phenomenon, the possible outcomes are $\omega = 1, 2, \cdots, 6$, the space sample is then $\Omega = \ \ 1, 2, \cdots, 6\ \ $, and $A = \ \ 1, 3, 5\ \ $ is an event. Note that Ω and   are also events, the former called the *certain event* and the latter called the *impossible event*. A probability measure assigns to each event a number called its probability.

2.1.2 Random Variables

Definition 2.1 (Random variables)**.** *A random variable is an application* $X : \Omega \quad E$ *such that:*

E is countable and in this case X is called discrete,

$E = \mathbb{R}$ *and* $\ r \quad \mathbb{R}, \quad \omega \quad X(\omega) \leq r\ \ $ *is an event.*

In the previous example of die tossing the identity function $X(\omega) = \omega$ defined from Ω to $\ \ 1, 2, \cdots, 6\ \ $ can be taken as random variable.

For a random variable X over a measurable space $(\Omega, \mathcal{F})$, we denote $\mathcal{F}_X$ the smallest sub σ-field of $\mathcal{F}$ where X is a random variable.

2.1.3 Probability

The probability $\mathbb{P}(A)$ of an event $A \quad \mathcal{F}$ measures the likelihood of its occurrence. Formally,

Definition 2.2 (Probability measure)**.** *A probability measure is a mapping* $\mathbb{P} : \mathcal{F} \quad \mathbb{R}$ *such that for every* $A \quad \mathcal{F}$

1. $0 \leq \mathbb{P}(A) \leq 1$

2. $\mathbb{P}(\Omega) = 1$

3. $\quad (A_i)_{i \quad \mathbb{N}} \quad \mathcal{F}^{\mathbb{N}}, \ \mathbb{P}\left(\bigcup_{i=0} A_i\right) = \lim_{n} \quad \mathbb{P}(\bigcup_{i=0}^{n} A_i)$ *(i.e.* $\mathbb{P}$ *is sigma additive).*

Definition 2.3 (Independence of events)**.** *Two events* A *and* B *are independent if*

$$\mathbb{P}(A \setminus B) = \mathbb{P}(A)\mathbb{P}(B) \ .$$

Definition 2.4 (Independence of random variables)**.** *Let* $X : \Omega \quad A$ *and* $Y : \Omega \quad B$ *be two random variables.*

If A *and* B *are countable, then* X *and* Y *are independent if*

$$(a, b) \quad A \times B, \ \mathbb{P}(\ X = a \ \setminus \ Y = b\ ) = \mathbb{P}(\ X = a\ )\mathbb{P}(\ Y = b\ ) \ .$$

If $A = B = \mathbb{R}$*, then* X *and* Y *are independent if*

$$(a, b) \quad A \times B, \ \mathbb{P}(\ X \leq a \ \setminus \ Y \leq b\ ) = \mathbb{P}(\ X \leq a\ )\mathbb{P}(\ Y \leq b\ ) \ .$$

2.1.4 Conditional probability

Let A and B be two events, the probability that A occurs given that B has occurred is called the conditional probability and denoted $\mathbb{P}(A \ B)$. Formally, $\mathbb{P}(A \ B)$ is the probability of A according to a new probability measure on the sample space Ω such that all the outcomes not in B have probability 0. Mathematically, $\mathbb{P}(A \ B)$ is defined for $\mathbb{P}(B) = 0$:

$$\mathbb{P}(A \quad B) = \frac{\mathbb{P}(A \setminus B)}{\mathbb{P}(B)} \ .$$

Expectation

The expected value of a real valued random variable is the weighted average of all possible values that this random variable can take on. The weights used in computing this average correspond to the probabilities in case of a discrete random variable

Definition 2.5 (Expectation)**.** *Let* X *be a random variable over a measurable space* $(\Omega, \mathcal{F})$*, the expected value of* X *denoted* $\mathbb{E}[X]$ *is the following Lebesgue integral (when it exists):*

$$\mathbb{E}[X] = \int_{\Omega} X(\omega)\mathbb{P}(d\omega) \ .$$

Proposition 2.6. *Let* X *and* Y *be two independent random variables, then*

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] \ .$$

Lemma 2.7 (Fatou s lemma)**.** *Let* $(X_n)_{n \quad \mathbb{N}}$ *be a sequence of random variables, then*

$$\mathbb{E}[\liminf_{n} X_n] \leq \liminf_{n} \mathbb{E}\left[X_n\right] \ .$$

Conditional Expectation

For a random variable X over a measurable space $(\Omega, \mathcal{F})$ and let $\mathcal{E}$ a sub σ-field of $\mathcal{F}$. The expectation of X conditioned by $\mathcal{E}$ represents the expected value of X when the information available is $\mathcal{E}$. Formally,

Definition 2.8 (Conditional expectation)**.** *Let X be a random variable over a measurable space $(\Omega, \mathcal{F})$ and let $\mathcal{E}$ be a sub σ-field of $\mathcal{F}$. The conditional expectation of X given $\mathcal{E}$ denoted* $\mathbb{E}[X\ \mathcal{E}]$ *is the only $\mathcal{F}_{\mathcal{E}}$-measurable function that satisfies:*

$$E \quad \mathcal{E},\ \mathbb{E}\left[\mathbb{E}[X \quad \mathcal{E}]\mathbb{1}_E\right] = \mathbb{E}[X\mathbb{1}_E] \ .$$

Proposition 2.9. *Let X be a random variable over a measurable space $(\Omega, \mathcal{F})$ and let $\mathcal{E}$ be a sub σ-field of $\mathcal{F}$, then*

$$\mathbb{E}[\mathbb{E}[X \quad \mathcal{E}]] = \mathbb{E}[X] \ .$$

2.2 Asymptotic Behavior

We turn our attention now to the way events and random variables behave in the long term, more precisely, we consider sequences of events or random variables and study their limit.

For a given sequence of events $\ A_n \ _{n\geq 1}$, one is interested in the probability that A_n occurs infinitely often in the sense that A_n is realized for infinitely many indices n. Borel-cantelli lemma answers the former question, but before stating the lemma let us formalize the sentence *"occurs infinitely often"*.

Definition 2.10. *Let $(A_n)_{n\geq 1}$ be a sequence of events,*

$$A_n \ i.o. \ = A_n occurs\ in\ finitely\ often = \bigcap_{n\geq 1} \bigcup_{k\geq n} A_k \ .$$

Lemma 2.11 (Borel-Cantelli)**.** *Let $(A_n)_{n\geq 1}$ be a sequence of events such that*

$$\sum_{n=1} \mathbb{P}(A_n) < \quad .$$

Then

$$\mathbb{P}\left(A_n \ i.o.\right) = 0 \ .$$

The following theorem draws a relation between the expectation of a random variable and its mean value.

Theorem 2.12 (Strong law of large numbers)**.** *Let $(X_n)_{n\geq 1}$ be an* i.i.d[1] *sequence of random variables such that*

$$\mathbb{E}[\ X_1\ ] < \quad .$$

Then

$$\mathbb{P}\left(\lim_{n} \ \frac{1}{n}\sum_{i=0}^{n} X_i = \mathbb{E}[X_1]\right) = 1 \ .$$

[1] Independent and identically distributed.

2.3 Markov Chains

Definition 2.13 (Distribution)**.** *Let S a finite set. Denote $\Delta(S)$ the set of probability distributions over S,*

$$\Delta(S) = \left\{ \delta \in [0,1]^S \;\middle|\; \sum_{q \in S} \delta(q) = 1 \right\} .$$

Remark 2.14. *From now on, for a set S we denote δ_S the uniform distribution over the states of S.*

Definition 2.15 (Support)**.** *Let $\delta \in \Delta(S)$ be a distribution the support of δ denoted* $\mathrm{Supp}(\delta)$ *is the set*

$$\mathrm{Supp}(\delta) = \{ s \in S \mid \delta(s) > 0 \} .$$

2.3.1 Homogeneous Markov Chains

Definition 2.16 (Markov chain)**.** *A Markov chain is a tuple $\mathcal{M} = (S, p)$, where*

- *S is a finite set of states,*
- *$p \in [0,1]^{S \times S}$ is transition a matrix with the property that the elements of each line sum up to 1.*

$p(s, r)$ denotes the probability to reach state r from state s.

Intuitively, a Markov chain models the temporal evolution of a random variable. There exists general model of Markov chains with continuous state space, continuous time, and such that transition probabilities depends on time. In this manuscript we consider homogeneous discrete time Markov chains. Therefore when referring to a Markov chain $\mathcal{M}$, it is implicit that $\mathcal{M}$ is homogenous discrete time Markov chain. We usually represent Markov chains by their transition graph as depicted in Fig 2.1.

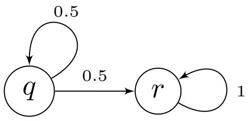


Figure 2.1: Markov chain

Example 2.17. *Consider the Markov chain depicted in Fig 2.1.*

- *the set of state $S = \{s, r\}$,*
- *the transition matrix $p = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix}$.*

Given a Markov chain $\mathcal{M}$, denote by S^ω any infinite sequence of states in S. A cone is any sequence of the form $s_0 s_1 \cdots s_n S^\omega$. Given an initial state $s_0 \in S$, we associate with $\mathcal{M}$ the probability measure $\mathbb{P}_{s_0}$ over the measurable space $(S^\omega, \mathcal{F})$ with $\mathcal{F}$ the smallest σ-field generated by cones. The measure $\mathbb{P}_{s_0}$ satisfies all the axioms of probability measure plus the axiom of cones that is

$$\mathbb{P}_{s_0}(r_0 r_1 \cdots r_n S^\omega) = \begin{cases} 0 \text{ if } s_0 = r_0, \\ p(r_0, r_1) p(r_1, r_2) \cdots p(r_{n-1}, r_n) \text{ otherwise} \end{cases} .$$

In the sequel, we denote S_i the random variable with values in S that gives the state of a Markov chain at time i, i.e. $S_i(s_0 s_1 s_2 \cdots) = s_i$.

Definition 2.18 (Markov properties). *Let $\mathcal{M}$ be a Markov chain and s_0 an initial state, then*

$$\mathbb{P}_{s_0}(S_n = s_n \quad S_0 = s_0 \quad \cdots \quad S_{n-1} = s_{n-1}) = \mathbb{P}_{s_0}(S_n = s_n \quad S_{n-1} = s_{n-1}) .$$

Definition 2.19 (Stopping time). *A stopping time T with respect to a sequence of random variable $(S_n)_{n \leq 0}$ is a random variable with values in $\mathbb{N}$ such that the event $T = m$ is $S_0, \cdots, S_m$ $-$measurable.*

In the sequel, for a state s we denote T_s the stopping time in state s defined by

$$T_s = \min \quad n \quad \mathbb{N} \quad S_n = s \quad .$$

Definition 2.20 (Strong Markov). *Let $\mathcal{M}$ be a Markov chain, T be a stopping time, and s_0 be an initial state, then*

$$\mathbb{P}_{s_0}(S_{T+n} = s_n \quad S_{T+n-1} = s_{n-1} \quad T < \quad) = \mathbb{P}_{s_0}(S_n = s_n \quad S_{n-1} = s_{n-1} \quad T < \quad) .$$

Definition 2.21 (Recurrent states). *Let $\mathcal{M}$ be a Markov chain and let $s \quad S$ be a state of $\mathcal{M}$, s is recurrent if for every state $t \quad S$ we have:*

$$\mathbb{P}_s(\quad n \quad \mathbb{N}, S_n = t) > 0 = \quad \mathbb{P}_t(\quad m \quad \mathbb{N}, S_m = s) > 0 .$$

A state of a Markov chain is either transient or recurrent.

Definition 2.22 (Closed class). *Let $C \subseteq S$ be a subset of states. C is a closed class if C is a strongly connected and contains only recurrent states.*

A Markov chain is irreducible if it is strongly connected. An alternative way to define irreducibility is to say that all states of the Markov chain belongs to the same closed class.

Example 2.23. *Back to the example of Fig 2.1, the only recurrent state is state r and q is the only transient state.*

The underlying idea beyond this decomposition of states of a Markov chain $\mathcal{M}$ is that, every run of $\mathcal{M}$ will eventually reach a closed class and never visit transient states. This decomposition raises two natural questions:

1. what is the mean time before reaching a given closed class C?

2. What is the upper bound of the absorption mean time?

Absorption s mean time

The mean time absorption is the mean time for a Markov chain $\mathcal{M}$ to reach the different closed classes of $\mathcal{M}$.

Definition 2.24 (Absorption s mean time)**.** *Let $\mathcal{M}$ be a Markov chain with initial state s_0. The absorption's mean time is the random variable with values in $\mathbb{N}$ defined as follows:*

$$\mathbb{E}_{s_0}[\min \quad n \quad \mathbb{N} \quad S_n \textit{ is recurrent }] \ .$$

In order to compute this time, we give a canonical representation of p the transition matrix. For any transition matrix p let P the matrix where states of $\mathcal{M}$ are reordered so the transient states come first. Hence, if T is the set of transient states and R the set of recurrent states, p is rewritten in the following shape:

$$P = \left(\begin{array}{c|c} P & P \\ \hline 0 & P \end{array}\right)$$

Where $P \quad [0,1]^{T\times T}, P \quad [0,1]^{T\times R}$, and $P \quad [0,1]^{R\times R}$. Write N the matrix $(I - P\ )^{-1}$ (N exists since the kernel of $(I - P\ )$ is equal to 0). One can prove that $N = I + P\ + P\ ^2 \cdots$. It follows that the time to absorption is Nc where c is the column vector whose all entries are 1.

Example 2.25. *Consider the Markov chain whose transition matrix is given by*

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0.5 & 0 & 0 \\ 2 & 0 & 0.5 & 0 & 0.5 & 0 \\ 3 & 0 & 0 & 0.5 & 0 & 0.5 \\ 4 & 0 & 0 & 0 & 0 & 1 \end{array}$$

The canonical form is then

$$\begin{array}{c|ccc|cc} & 1 & 2 & 3 & 0 & 4 \\ 1 & 0 & 0.5 & 0 & 0.5 & 0 \\ 2 & 0.5 & 0 & 0.5 & 0 & 0 \\ 3 & 0 & 0.5 & 0 & 0 & 0.5 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 0 & 1 \end{array}$$

A simple computation shows that the matrix N is

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ 1 & 1.5 & 1 & 0.5 \\ 2 & 1 & 2 & 1 \\ 3 & 0.5 & 1 & 1.5 \end{array}$$

It follows that the mean time of absorption is given by the following vector

$$\begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

This means that from state $1, 2,$ *and* 3 *the mean time to absorption is* $3, 4,$ *and* 3.

Proposition 2.26. *Let $\mathcal{M}$ be a Markov chain and s_0 be an initial state, then*

$$\mathbb{E}_{s_0}[\min \ \ n \quad \mathbb{N} \quad S_n \textit{ is recurrent } \ ] \leq 2^{Q(\ \mathcal{M} \ )} \ ,$$

where Q is a polynomial and $\ \mathcal{M}\ $ is the description of the Markov chain $\mathcal{M}$.

Proof. Using previous argument we know that the time to absorption is given by $(I - P \ )^{-1}(s_0)$. Hence

$$\mathbb{E}_{s_0}[\min \ \ n \quad \mathbb{N} \quad S_n \text{ is recurrent } \ ] \leq \max_{s \quad S} \ (I - P \ )^{-1}(s) \quad .$$

The right hand side of this equation is a rational fraction in $\ \mathcal{M}\ $, thus there exists a polynomial Q with degree at most $\ S\ $ and whose coefficient are polynomials expression with degree at most $\ S\ $ in the coefficient of p such that

$$\mathbb{E}_{s_0}[\min \ \ n \quad \mathbb{N} \quad S_n \text{ is recurrent } \ ] \leq 2^{Q(\ \mathcal{M} \ )} \ ,$$

where $\ \mathcal{M}\ $ is the description of the Markov chain $\mathcal{M}$. □

Steady distribution

Definition 2.27. *(Steady distribution) Steady distribution is a distribution $\pi \quad \Delta(S)$ such that:*

$$\pi = \pi p \ \ .$$

This distribution always exists for homogenous Markov chains.

2.3.2 Markov Chains with Reward

Lemma 2.28 (see e.g. Theorem 1.10.2 [Nor97])**.** *Let $\mathcal{M}$ be a Markov chain and $r : S \qquad \mathbb{R}$ be a reward function. The following equality holds for almost all runs.*

$$\liminf_{n} \sum_{i=0}^{n-1} \frac{r(S_i)}{n} = \limsup_{n} \sum_{i=0}^{n-1} \frac{r(S_i)}{n} \quad .$$

Lemma 2.29. *Let $\mathcal{M}$ be an irreducible Markov chain with reward. Let s be a state of $\mathcal{M}$. Assume that*

$$\mathbb{P}_s \left(\liminf_{n} \sum_{i=0}^{n} r(S_i) = \qquad \right) \quad .$$

Then there exists an $\eta > 0$ such that:

$$\mathbb{E}_s \left[\frac{1}{T_s} \sum_{i=0}^{T_s - 1} r(S_i) \right] \geq \eta \ \ .$$

Moreover the bit complexity of η is polynomial in the size of $\mathcal{M}$.

Proof. Let $\mathcal{M}$ be a finite irreducible Markov chain with reward. Suppose

$$s \quad S, \ \mathbb{P}_s \left(\liminf_{n} \sum_{i=0}^{n} r(S_i) = \qquad \right) = 1$$

According to [BBE10a]

$$s \quad S,\ \mathbb{P}_s\left(\liminf_n \sum_{i=0}^{n} r(S_i) = \qquad \right) = 1 \qquad \mathbb{E}_s\left[\frac{1}{T_s}\sum_{i=0}^{T_s-1} r(S_i)\right] > 0\ .$$

This proves the first part of the lemma.

We use a discounted approximation to prove the second part. Let $0 < \lambda < 1$ and $V_\lambda \quad \mathbb{R}^S$ the vector defined by,

$$V_\lambda(s) = \mathbb{E}_s\left[\sum_{i\geq 0} \lambda^i r(S_i)\right]\ .$$

We first show that

$$\lim_{\lambda \quad 1}(1-\lambda)V_\lambda(s) = \mathbb{E}_s\left[\lim_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right]\ . \tag{2.1}$$

By [Put94] (Corollary 8.2.4) we have

$$(1-\lambda)^{-1}\mathbb{E}_s\left[\lim_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] = V_\lambda(s) - h(s) - f_s(\lambda)\ , \tag{2.2}$$

where $f_s(\lambda)$ is a function which converges to 0 as λ converges to 1 from below and $h(s)$ is the vector that gives the reward of the Markov chain at the steady distribution. Multiplying both sides of (2.2) by $(1-\lambda)$ and passing to the limit when λ converges to 1 leads (2.1).

Second, we have

$$\begin{aligned} V_\lambda(s) &= \mathbb{E}_s\left[\sum_{i\geq 0} \lambda^i r(v_i)\right] \\ &= r(s) + \sum_{t \quad S} \mathbb{P}_s(S_1 = t)\mathbb{E}_s\left[\sum_{i\geq 1} \lambda^i r(S_i) \,\middle|\, S_1 = t\right] \\ &= r(s) + \lambda \sum_{t \quad S} \mathbb{P}_s(S_1 = t)V_\lambda(t)\ , \end{aligned}$$

where R is the reward vector and P is the transition matrix of $\mathcal{M}$. Hence

$$\begin{aligned} V_\lambda &= R + \lambda P V_\lambda \\ &= (I - \lambda P)^{-1} R\ , \end{aligned} \tag{2.3}$$

$(I - \lambda p)^{-1}$ exists because the kernel of $(I - \lambda p)$ is equal to 0 (consequence of the fact that $0 \leq \quad \lambda p \quad < 1$). For every state s of $\mathcal{M}$, (2.3) can be written as

$$(1-\lambda)V_\lambda(s) = (1-\lambda)((I - \lambda P)^{-1}R)(s)\ . \tag{2.4}$$

The right hand side of (2.4) is a rational fraction of λ, therefore there exists two polynomials P and Q with degree at most $\quad S \quad$ and whose coefficients are polynomial expression with degree at most $\quad S$ in the coefficients of p. such that

$$(1-\lambda)V_\lambda(s) = \frac{P(\lambda)}{Q(\lambda)}\ . \tag{2.5}$$

By (2.1) we get that

$$\mathbb{E}_s\left[\lim_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] = \frac{P(1)}{Q(1)} \quad . \tag{2.6}$$

The right hand side of (2.6) is a polynomial expression of degree at most S in the coefficients of p Thus there exists a polynomial T such that

$$\mathbb{E}_s\left[\lim_n \frac{1}{n}\sum_{i=0}^{n-1} r(Si)\right] \geq 2^{-T(\ \mathcal{M}\ )} \quad ,$$

where $\mathcal{M}$ of the description of $\mathcal{M}$.
Using the strong Markov property we have

$$\mathbb{E}_s\left[\lim_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] = \mathbb{E}_s\left[\frac{1}{T_s}\sum_{i=0}^{T_s-1} r(S_i)\right] \quad .$$

Which terminates the proof of the lemma. □

Chapter 3

Markov Decision Processes

Contents

3.1 Introduction

In systems where hardware failures and other random events occur, the behavior of the environment is typically represented as a stochastic process [KNP07, TAHW09]. Markov decision processes have proven to be a powerful [KEY07, BCG05] yet algorithmically tractable [CY90] tool. In Markov decision processes, the environments moves are chosen randomly according to fixed transition probabilities that depend on the current state of the system.

An optimal controller of the system maximizes the probability that the system behaves correctly in its stochastic environment. Synthesizing such a controller amounts to computing an optimal strategy σ for Max in the Markov decision process.

Outline of the chapter

- In Section 3.2 we discuss the model.

- In Section 3.3 we introduce strategies and the probability measure associated with a strategy.

- In Section 3.4 we introduce the notions objectives and values.

- In Section 3.5 we introduce the main tools for studying reachability objectives.

- In Section 3.6 we introduce the main tools for studying tail objectives.

- In Section 3.7 we introduce known result on parity games.

- In Section 3.8 we study tail games with quantitative pay-offs.

3.2 Markov Decision Processes and Plays

A Markov decision process is a transition system such that at each step Max chooses an action a to play from the current state s then the successor is chosen at random from the set of reachable states from s by playing the action a. Formally,

Definition 3.1 (Markov decision process)**.** *A Markov decision process is a tuple $\mathcal{M} = (S, A, p)$ such that*

S is a finite set of states,

A is a finite set of actions,

p is function defined by $p : S \times A \quad \Delta(S)$.

Example 3.2. *Fig 3.1, represents a Markov decision process where:*

the set of states is $\ q, r\ $,

the set of actions is $\ a, b\ $,

the function p is described in the transition graph.

Remark 3.3. *Note that a Markov decision process where the set A is a singleton is nothing but a Markov chain.*

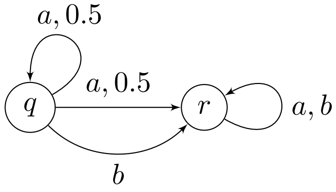


Figure 3.1: A Markov decision process.

For a given Markov decision process $\mathcal{M}$ and a state $s_0 \quad S$, a play from s_0 is an infinite sequence $s_0 a_0 s_1 a_1 \cdots \quad S(AS)^\omega$ such that for every $i \geq 0$ we have

$$p(s_i, a_i)(s_{i+1}) > 0 \ .$$

A finite prefix of a play is called history. We denote by $hs \quad S(AS)^*$ a history of a play up to state s. By S_i we denote the random variable with values in S that gives the current state after i steps i.e.

$$S_i(s_0 a_0 s_1 a_1 \cdots) = s_i \ ,$$

and by A_i we denote the random variable with values in A that gives the action played after i steps i.e.

$$A_i(s_0 a_0 s_1 a_1 \cdots) = a_i \ .$$

A useful notion is the one of sub Markov decision process. Intuitively, a sub Markov decision process $\mathcal{M}$ is a subgraph such that a play can always continue in $\mathcal{M}$. Formally,

Definition 3.4 (Sub Markov decision processes)**.** *Let $\mathcal{M}$ be a Markov decision process with state space S and actions A. $\mathcal{M}[S\ ]$ is a sub Markov decision process induced by the subset $S \subseteq S$ if*

$$(\ s \quad S\ ),\ (\ a \quad A),\ p(s, a)(S\ ) = 1 \ .$$

3.3 Strategies and Measures

While playing, Max chooses her moves according to a strategy. A strategy for Max associates to each history $hs \quad S(AS)^*$ a distribution over A. Formally,

Definition 3.5 (Strategy)**.** *A strategy σ for* Max *is an application:*

$$\sigma : S(AS)^* \quad \Delta(A) \ .$$

A strategy σ is:

– pure, if for every history $hs \quad S(AS)^*$, the set $\mathrm{Supp}(\sigma(hs))$ is a singleton.

– stationary, if for every history $hs \quad S(AS)^*$ the outcome of $\sigma(hs)$ depends only on the state s.

– positional, if it is a pure and stationary.

In the case where a strategy is not stationary, a natural question is: how much information should the player remember in order to make the next move. This is formalized by the notion of strategies with memory.

Definition 3.6 (Strategies with memory)**.** *A strategy with memory is a set M, a memory state $m_0 \quad M$ called the initial memory state, and two functions σ_m, σ_u such that:*

$$\sigma_m : S \times M \quad \Delta(S),$$

$$\sigma_u : S \times M \quad M.$$

The function σ_u is usually called the update function, it gives the next memory state. In the case of stationary strategies, the set M is a singleton.

In a Markov decision process $\mathcal{M}$, once we have fixed a strategy σ for Max and an initial state s, this defines naturally a probability measure $\mathbb{P}_s^\sigma$ over $s(AS)^\omega$ the set of all plays starting from s. This probability measure is defined by induction as follows

$$r \quad S,\ \mathbb{P}_s^\sigma(r) = \begin{cases} 1 \text{ if } s = r \ , \\ 0 \text{ otherwise.} \end{cases}$$

Let $h \quad S(AS)^*$ a finite history such that h starts in $s \quad S$ and ends in $t \quad S$, then:

$$r \quad S,\ \mathbb{P}_s^\sigma(har(AS)^\omega) = \mathbb{P}_s^\sigma(h) \cdot \sigma(h)(a) \cdot p(t,a)(r) \ .$$

Thanks to Tulcea s theorem [BS78], there is a unique extension of $\mathbb{P}_s^\sigma$ to $s(AS)^\omega$.

3.4 Objectives and Values

Definition 3.7 (Objective)**.** *A winning condition Φ is a subset of S^ω. We say that a play is winning for* Max *if it belongs to Φ.*

An objective Φ is a Borel objective if Φ is a Borel set.

While playing, Max is trying to maximize the probability that some objective is achieved.

Let $\Phi \subseteq S^\omega$ be an objective, the value of a state s S with respect to strategy σ is denoted:

$$\mathrm{Val}_\sigma(s) = \mathbb{P}_s^\sigma(\Phi) \ ,$$

intuitively this is the probability that Max wins if the play starts in s and is consistent with the strategy σ.

Definition 3.8 (Values and optimal strategies)**.** *The value of a state is defined as:*

$$\mathrm{Val}(s) = \sup_\sigma \mathrm{Val}_\sigma(s) \ .$$

Obviously, Max wants to apply the best possible strategy so she can ensure the best possible value. The best possible strategies are called optimal. Formally,

Definition 3.9 (Optimal strategy)**.** *A strategy σ is optimal if:*

$$\mathrm{Val}_\sigma(s) = \mathrm{Val}(s) \ .$$

Optimal strategies do not always exist, hence a relaxed notion of optimality has been defined. It is the so-called ε-optimal strategies.

Definition 3.10 (ε-Optimal strategy)**.** *Let $\varepsilon > 0$, a strategy σ is ε-optimal if:*

$$\mathrm{Val}_\sigma(s) \geq \mathrm{Val}(s) - \varepsilon \ .$$

One can also study Markov decision processes from a rather qualitative point of view. This alternative approach was introduced by De Alfaro in [dAH00] and it informs wether Max has a strategy that ensures him to satisfy the objective with probability 1. In which case we say that Max wins almost-surely. Dually, we say that Max wins positively, if she has a strategy that ensures the satisfaction of the objective with probability strictly positive.

Definition 3.11 (Almost-sure and positive winning strategies)**.** *We say that* Max *wins almost-surely (resp. positively) from a state s if she has a strategy σ such that $\mathbb{P}_s^\sigma(\Phi) = 1$ (resp. $\mathbb{P}_s^\sigma(\Phi) > 0$).*

Remark 3.12. *In the sequel we use the following notations.*

A strategy which allows Max *to win almost-surely (resp. positively) is called an almost-sure (resp. positive) strategy.*

A state s is said to be almost-sure (resp. positive) for Max*, if there exists an almost-sure (resp. positive) strategy from s for* Max.

The set of almost-sure (resp. positive) winning states for Max *is denoted $W_{=1}$ (resp. $W_{>0}$) and called the almost-sure (resp. positive) winning region of* Max.

3.5 Reachability Objectives

The simplest class of objectives is the class of *reachability* objectives. In a reachability game, the goal of the player is to reach a set of target states $T \subseteq S$, in other words the winning condition is set of plays:

$$\Phi = S^* T S^\omega \ .$$

In reachability games, the sets of positive and almost-sure winning states are easy to compute, using elementary fixpoint algorithms.

The positive attractor for Max to a subset T of S, denoted $\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$, is the set of states in S from which Max can reach T with positive probability. Formally we define it as follows.

Definition 3.13 (Positive attractor). *Let $W \subseteq S$ be a subset and $f : 2^W \quad 2^W$ be the operator such that for any $U \subseteq W$,*

$$f(U) = \quad s \quad W \quad (s \quad T) \quad (\ a \quad A, \ (p(s,a)(U) > 0) \quad (p(s,a)(W) = 1)) \quad .$$

Then $\overline{\mathrm{R}_{\mathrm{Max}}}(T, W)$ is the least fixed point of f.

For a Markov decision process $\mathcal{M}$ with state space S and a target states T. The set $S \quad \overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ induces a sub Markov decision process. Actually, it induces a specific sub Markov decision process called *trap*. Formally, a trap is:

Definition 3.14 (Trap). *Let $\mathcal{M}$ be a Markov decision process. $\mathcal{M}[S \ ]$ is a trap induced by a subset $S \ \subseteq S$ if*

$$(\ s \quad S \ ), \ (\ a \quad A), \ p(s,a)(S \ ) = 1 \ .$$

Proposition 3.15. *Let $\mathcal{M}$ be a Markov decision process and $T \subseteq S$ a target set. The complement of $\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ in S is a trap for* Max.

Proof. We show that $S \quad \overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ is a trap for Max. Assume toward a contradiction that $S \quad \overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ is not a trap for Max, it would mean that there exists a state $s \quad S \quad \overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ and an action $a \quad A$ such that

$$p(s,a)(\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)) > 0 \ ,$$

which contradicts the fixpoint definition of $\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$. □

The almost-sure attractor for Max to a subset T of S, denoted $\overline{\mathrm{R}_{\mathrm{Max=1}}}(T, S)$, is the set of states from which Max can reach T with probability 1.

Definition 3.16 (Almost-sure attractor). *The almost-sure attractor is the set $\overline{\mathrm{R}_{\mathrm{Max=1}}}(T, S) = \backslash_i R_i$ where R_i is obtained by the following induction:*

$$R_0 = \overline{\mathrm{R}_{\mathrm{Max}}}(T, S) \ ,$$
$$R_{i+1} = \overline{\mathrm{R}_{\mathrm{Max}}}(T, R_i) \ .$$

In Proposition 3.17 we show that there is a positional strategy for Max to *attract* the play from any state in $\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ (resp. $\overline{\mathrm{R}_{\mathrm{Max=1}}}(T, S)$) to T with positive probability (resp. probability 1).

Proposition 3.17. *Let $W \subseteq S$ a subset of states such that W induces a sub Markov decision process $\mathcal{M}[W]$. The positive (resp. almost-sure) attractor of* Max *to T $\overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$ (resp. $\overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)$) is exactly the set of positive (resp. almost-sure) states of* Max *in the reachability objective to W^*TW^ω played on the Markov decision process $\mathcal{M}[W]$.*

Proof. We show that Max has a positive strategy for the reachability objective. From any state $s \quad \overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$. By Definition 3.13 we know that for every state $s \quad \overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$ there exists $n > 0$ such that $s \quad f^n(T)$, denote $n(s) = \min \quad n \quad \mathbb{N} \quad s \quad f^n(T)$. Hence a positive strategy for Max from s consists in choosing an action $a \quad A$ such that $p(s,a)(f^{n(s)-1}(T)) > 0$. The existence of a is established by Definition 3.13, since from each $s \quad \overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$ there is a non zero probability to eventually reach T we get that σ is positively winning. We show that the positive region is subsumed by the positive attractor. According to Proposition 3.15 the set $W \quad \overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$ is a trap for Max thus for any strategy σ and any state $s \quad W \quad \overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$ we have:

$$\mathbb{P}_s^\sigma(\quad n \geq 0,\ S_n \quad T) = 0 \ ,$$

which shows that the positive region is subsumed by $\overline{\mathrm{R}}_{\mathrm{Max}}(T,W)$.

The second part of Proposition 3.17 is a consequence of the following facts: *a*) the almost-sure attractor is subsumed by the positive attractor and *b*) for any state $s \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)$ either $s \quad T$ or there exists an action a such that $p(s,a)(\overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)) = 1$. Hence from fact *a*) Max has a positive strategy to reach T and by fact *b*) we get that this happens almost-surely. We show that the almost-sure region is subsumed by the almost-sure attractor. Let $s \quad W \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)$ be a state and σ be a strategy, then either

$$\mathbb{P}_s^\sigma(\quad n \geq 0,\ S_n \quad T \quad n \geq 0,\ S_n \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)) = 0 \ , \tag{3.1}$$

or

$$\mathbb{P}_s^\sigma(\quad n \geq 0,\ S_n \quad T \quad n \geq 0,\ S_n \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)) > 0 \ . \tag{3.2}$$

If Equation (3.1) holds then it is clear that s is not almost-sure for the reachability objective. If Equation (3.2) holds, then if

$$\mathbb{P}_s^\sigma(\quad n \geq 0,\ S_n \quad T \quad n \geq 0,\ S_n \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)) = 1 \ ,$$

it implies that $s \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}$ which contradicts the assumption, hence

$$\mathbb{P}_s^\sigma(\quad n \geq 0,\ S_n \quad T \quad n \geq 0,\ S_n \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(T,W)) < 1 \ ,$$

which shows that s is not almost-sure. □

We define also the safe set for Max with respect to a subset $B \subseteq S$ as the largest sub Markov decision process in which Max has a strategy to avoid reaching B for sure. Formally,

Definition 3.18 (Safe set)**.** *Let $B \subseteq S$ a set of bad states, the safe set for* Max *with respect to B is denoted* Safe(B,S) *and obtained as follows:*

$$\mathrm{Safe}(B,S) = S \quad \overline{\mathrm{R}}_{\mathrm{Max}}(B,S)$$

We conclude this section by given examples of computation of positive attractor, and almost-sure attractor.

Example 3.19. *Consider the Markov decision process of Fig 3.2, the positive attractor to the set $T = \quad r,t \quad$ is $\overline{\mathrm{R}}_{\mathrm{Max}}(T,S) = \quad p,q,r,s,t \quad$ and the almost-sure attractor to the set $T = \quad r,t \quad$ is $\overline{\mathrm{R}}_{\mathrm{Max}=1}(T,S) = \quad p,q,r,t$*

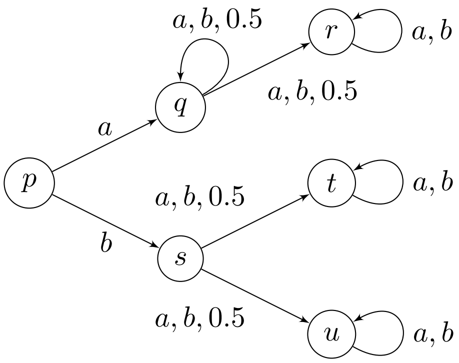


Figure 3.2: Reachability game with target set $T = \ r, t\ $.

3.6 Tail Objectives

An important class of objectives that is widely used in verification, is the class of tail objectives.

Definition 3.20 (Tail objective)**.** *Let* $\Phi \quad S^\omega$ *a winning condition.* Φ *is tail if* $\ u \quad S^*$ *and* $\ w \quad S^\omega$ *we have:*

$$uw \quad \Phi \qquad w \quad \Phi\ .$$

We say that a Markov decision process is tail if the objective associated to it is tail.

For Markov decision processes equipped with tail objective, the notions of values and qualitative solutions are tightly linked.

Theorem 3.21 (Positive-almost property [Cha07, GH10])**.** *In any Markov decision process equipped with a tail objective, either there exists a state with value 1, or all states have value 0. Moreover the states with value 1 are almost-sure.*

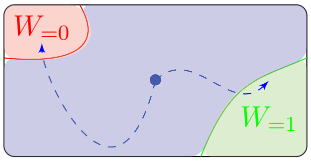


Figure 3.3: Positive-almost property

In Fig 3.3 we get the intuition of how one can use Corollary 3.23. Indeed, since the set of states with value 1 is exactly the set of almost-sure states, it follows that to increase her chances to win, Max needs only to increase her chances to reach the set with almost-sure states. It follows that computing the value of states for some tail condition, it suffices to compute the set of almost-sure states then compute the value of a reachability objective. Hence one can focus only on the computation of almost-sure regions when considering tail objectives on Markov decision processes.

Moreover, if the computation of the almost-sure region takes polynomial time it follows that the computation of the value of each state takes polynomial time as well. We give a formal proof in Corollary 3.22.

Corollary 3.22. *Let Φ be a tail winning condition. Assume that for every Markov decision process $\mathcal{M}$, $W_{=1}[\Phi]$ can be computed in polynomial time, then there exists a polynomial time algorithm to compute the value of each state of $\mathcal{M}$.*

Proof. Denote $\mathrm{Val}_{W_{=1}[\Phi]}$ the value of the reachability objective where the target set is $W_{=1}[\Phi]$. We first show that

$$s \quad S,\ \mathrm{Val}_{W_{=1}[\Phi]}(s) = \mathrm{Val}_{\Phi}(s)\ .$$

Theorem 3.21 shows that $W_{=1}[\Phi]$ is exactly the set of states with value 1. Hence

$$s \quad S,\ \mathrm{Val}_{W_{=1}[\Phi]}(s) \leq \mathrm{Val}_{\Phi}(s)\ . \tag{3.3}$$

Let us show the converse inequality. Let $s \quad S$ be a state and σ be a strategy, then

$$\mathbb{P}^{\sigma}_{s}(\Phi) = \mathbb{P}^{\sigma}_{s}(\Phi \setminus \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) + \mathbb{P}^{\sigma}_{s}(\Phi \setminus \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi])\ .$$

We show that

$$s \quad S,\ \quad \sigma,\ \mathbb{P}^{\sigma}_{s}(\Phi \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) = 0\ . \tag{3.4}$$

Assume toward a contradiction that there exists a strategy σ and a state s such that

$$\mathbb{P}^{\sigma}_{s}(\Phi \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) > 0\ .$$

Rewriting the above expression leads

$$\mathbb{P}^{\sigma}_{s}(\Phi \setminus \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) > 0\ ,$$

which shows that there exists a positively winning play in the largest trap $\mathcal{M}[S\ ]$ such that $S \ \subseteq S \quad W_{=1}[\Phi]$ which contradicts Theorem 3.21. Thus Equation (3.4) holds. It follows that:

$$\begin{aligned}\mathrm{Val}_{\Phi}(s) = \sup_{\sigma} \mathbb{P}^{\sigma}_{s}(\Phi) &= \sup_{\sigma} \mathbb{P}^{\sigma}_{s}(\Phi \setminus \quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) \\ &\leq \sup_{\sigma} \mathbb{P}^{\sigma}_{s}(\quad n \quad \mathbb{N},\ S_n \quad W_{=1}[\Phi]) = \mathrm{Val}_{W_{=1}[\Phi]}(s)\ .\end{aligned}$$

This shows Equation (3.3).

Second, since in Markov decision processes, the value of reachability games can be computed using linear programming in polynomial time [Con92]. Assuming that $W_{=1}[\Phi]$ can be computed in polynomial time terminates the proof. □

as consequence we get the following corollary

Corollary 3.23. *In any Markov decision process equipped with a tail objective, if there exists an almost-sure strategy with memory M, then there exists an optimal strategy with the same memory.*

Proof. This is consequence of the fact that reachability objectives are positional. □

Theorem 3.24 shows yet another nice property enjoyed by tail games. To our knowledge this is the first time an algorithm is provided to solve disjunction of tail objectives on Markov decision processes.

Theorem 3.24. *Let $\Phi_1, \cdots, \Phi_n$ be n tail objectives and $\mathcal{M}$ a Markov decision process. The almost-sure region for the game $\bigcup_{i=0}^n \Phi_i$ is given by the set*

$$\overline{\mathrm{R}}_{\mathrm{Max}=1}\left(\bigcup_{i=0}^{n} W_{=1}[\Phi_i], S\right) \ .$$

Proof. Denote $W = \overline{\mathrm{R}}_{\mathrm{Max}=1}(\bigcup_{i=0}^n W_{=1}[\Phi_i], s)$ and let us prove that Max has an almost-sure strategy from the set W. Max plays as follows. First she applies her attractor strategy until she reaches one of the $W_{=1}[\Phi_i]$ then she applies her almost-sure winning strategy σ_i associated with the objective Φ_i. This strategy is clearly almost-sure winning since Max reaches one of the $W_{=1}[\Phi_i]$ with probability 1.

To see that Max cannot win almost-surely outside W, consider the largest trap $\mathcal{M}[S\ ]$ such that $S \ \subseteq S \quad W$. For every i, Max has no almost-sure state in $(\mathcal{M}[S\ ], \Phi_i)$. According to the positive-almost property (c.f. Theorem 3.21), we get

$$s \quad S\,, \quad \sigma, \quad 0 \le i \le n,\ \mathbb{P}_s^\sigma(\Phi_i \quad k \ge 0,\ S_k \quad S\,) = 0 \ .$$

This implies

$$s \quad S\,, \quad \sigma,\ \mathbb{P}_s^\sigma\left(\bigcup_{i=0}^{n} \Phi_i \,\middle|\, \quad k \ge 0,\ S_k \quad S\right) = 0 \ ,$$

which shows that every state in S has value 0. For any other state s not in W and not in S , the probability for a given strategy σ that the play reaches S is strictly less than 1 otherwise it would imply that there exists a strategy σ such that

$$\mathbb{P}_s^\sigma(\ n \quad \mathbb{N},\ S_n \quad W) = 1 \ .$$

which implies that $s \quad W$, which terminates the proof. □

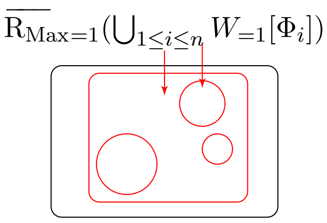

$\overline{\mathrm{R}}_{\mathrm{Max}=1}(\bigcup_{1\le i\le n} W_{=1}[\Phi_i])$

Figure 3.4: Solving disjunction of tail objectives

Remark 3.25. *Note that if M_i is the memory of the almost-sure strategy for the objective Φ_i then the memory of an almost-sure strategy for the objective $\bigcup_{i=0}^n \Phi_i$ is* $\max_i \quad M_i \quad$.

3.7 Parity Objectives

In this section we study a more specific objective, the so called *parity* objective. These objectives are very important in verification of reactive system [GTW02], indeed parity objectives subsume all the ω-regular objectives. These games were also looked at by Rabin earlier [Rab63], Rabin used the parity objective in the proof of complementation of tree automata. In parity objectives, we assign to each state a priority. The objective is achieved according to the set of priorities visited infinitely often during the play.

Definition 3.26 (Parity objective)**.** *Let $C \subsetneq \mathbb{N}$ be a finite subset, called the set of priorities and $\chi : S \quad C$ a priority function. The parity objective is:*

$$\mathrm{Par} = \quad s_0 s_1 \cdots \quad S^\omega \quad \limsup_n \chi(s_n) \textit{ is even} \quad .$$

For any priority $d \quad C$, we denote S_d the following set

$$S_d = \quad s \quad S \quad \chi(v) = d \quad .$$

A special case of parity objectives are *Büchi* objectives. The Büchi condition is formally defined as follows:

Definition 3.27 (Büchi games)**.** *Let $B \subseteq S$ be a subset of states, called the set of Büchi states. The Büchi objective is:*

$$\textit{Büchi} = (S^* B)^\omega \quad .$$

Theorem 3.28 ([CY95, CJH04])**.** *In Markov decision processes,* Max *has a positional optimal strategy. Moreover, the value of each state is computable in polynomial time.*

We give an other version of the proof of the theorem above. The reason we give this new version of the proof is to allow the reader to get a better insight and intuition regarding tools and notions presented earlier.

Proof. Let $\mathcal{M}$ be a Markov decision process, To prove the polynomial upper bound, notice that the parity condition can be written as a disjoint union of winning condition where in each one, Max wins if she satisfies the parity condition played in a parity Markov decision process with three priorities. Formally, for each priority c, we define $\mathcal{M}_c = (S, A, p, \chi_c)$ as the following Markov decision process:

– the set of states is the same as in $\mathcal{M}$,

– the set of action is the same as in $\mathcal{M}$,

– the transition function is the same as in $\mathcal{M}$,

– the coloring function is defined as follow:

$$\chi_c(s) = \begin{cases} 1 \text{ if } \chi(s) < c \ , \\ 2 \text{ if } \chi(s) = c \ , \\ 3 \text{ if } \chi(s) > c \ . \end{cases}$$

For each c we write Φ_c the parity objective associated with the coloring function χ_c, one can write:

$$\mathrm{Par} = \bigcup_{c \quad C} \Phi_c \quad .$$

According to Theorem 3.24 the almost-sure region is given by:

$$\overline{\mathrm{R}}_{\mathrm{Max}=1}\left(\bigcup_{c\in C} W_{=1}[\Phi_c]\right) .$$

Let us show now that solving parity objectives on a Markov decision process with exactly 3 priorities can be done in polynomial time. This is consequence of the fact that solving such a parity game amounts to solving a Büchi objective on the sub Markov decision process induced by the set $\mathrm{Safe}(S_3, S)$ and computing an almost-sure attractor (Details of the correctness are given in the proof of Theorem 4.3). Since Büchi objectives on Markov decision processes can be solved in polynomial time [dAH00, CJH03] it follows that that each $W_{=1}[\Phi_c]$ can be computed in polynomial time. Using result of Corollary 3.22 shows that the original objective can be solved in polynomial time.

Let us show that the strategy described is positional. This follows from the fact the strategy used for Büchi objectives are positional [dAH00, CJH03], and thanks to Remark 3.25, it follows that the almost-sure strategy for the objective $\bigcup_{c\in C}\Phi_c$ is positional as well. □

3.8 Mean-payoff and Positive-average Objectives

We turn our attention to another type of objective, we study in this section objectives with rewards.

3.8.1 Mean-payoff Objectives

Definition 3.29 (Mean-payoff objective)**.** *A Markov decision process with mean-payoff objective is a Markov decision process such that the set of states S is labelled with a reward mapping $r : S \to \mathbb{R}$ that assigns to each state a rational number called the reward. The value of a state $s \in S$ in $\mathcal{M}$ is*

$$\mathrm{Val}(s) = \sup_{\sigma} \mathbb{E}_s^{\sigma}\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] .$$

The value of a state in mean-payoff objective is not a probability but it is the maximal expected average reward Max can ensure.

To compute the value of a state in a Markov decision process equipped with a mean-payoff objective one uses linear programming. It is also well known that positional strategies are sufficient to play optimally [Put94].

Theorem 3.30. *Mean-payoff games can be solved in polynomial time. Moreover optimal strategies exist and can be chosen positional.*

3.8.2 Positive average objectives

In positive average objectives, Max wants to maximize the probability that the average value of the accumulated rewards is strictly positive.

Definition 3.31 (Positive-average objectives)**.** *Let $\mathcal{M}$ be a Markov decision process equipped with a reward function $r : S \to \mathbb{R}$. The positive average objective is:*

$$\mathrm{Avg}_{>0} = \left\{ s_0a_0s_1a_1\cdots \in S(AS)^{\omega} \mid \limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(s_i) > 0 \right\} .$$

At a first glance, mean-payoff games and positive-average games seem to be similar, Fig 3.5 exhibit an arena where the optimal strategies are different for mean-payoff objective and positive-average objective. Indeed an optimal strategy for the mean-payoff objective would go to the state with reward 4 while an optimal strategy for the positive-average objective would stay in the state with reward 1.

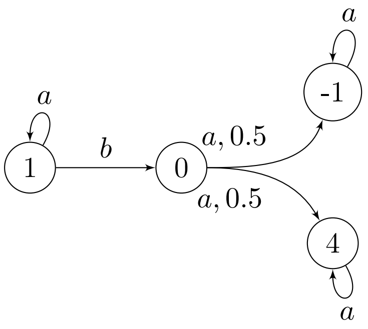


Figure 3.5: Markov decision process where the mean-payoff and positive-average objectives have different optimal strategies

There is another natural definition of positive average objective, which is very similar except the lim sup is replaced by lim inf. We denote this condition $\underline{\mathrm{Avg}}_{>0}$.

We shall show later that the choice of either definition does not impact our results for Markov decision processes.

Theorem 3.32 ([BBE+10b]). *In a Markov decision process equipped with positive-average condition,* Max *has a positional optimal strategy. Moreover, the value of each state is computable in polynomial time.*

We give our own proof of this theorem, to illustrate the use of closed components and recurrent states.

Proof of Theorem 3.32. Since the set of almost-sure states is exactly the set of states with value 1 (c.f. Theorem 3.21), computing the values amounts to compute the almost-sure region.

We show that the almost-sure region for the objective $\mathrm{Avg}_{>0}$ is the largest sub Markov decision process $\mathcal{M}[W]$ such that the value of each state in W is strictly greater than 0 for the mean-payoff objective in $\mathcal{M}[W]$. Such a sub Markov decision is not empty because otherwise, it would imply that every nonempty sub Markov decision process $\mathcal{M}[S\ ]$ is such that for every $s \quad S$ the value of s is less or equal to 0 for the mean-payoff objective. In particular since $\mathcal{M}$ is a sub Markov decision process of $\mathcal{M}$ it follows that for every strategy σ we get

$$s \quad S,\ \mathbb{E}_s^\sigma\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] \leq 0 \ .$$

Applying the strong law of large numbers (c.f. Theorem 2.12) we obtain

$$s \quad S,\ \mathbb{P}_s^\sigma\left(\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i) \leq 0\right) = 1 \ .$$

Which implies

$$s \quad S,\ \mathbb{P}_s^{\sigma}(\mathrm{Avg}_{>0}) = 0 \ .$$

Thus according to the positive-almost property, there is no almost-sure state for the objective $\mathrm{Avg}_{>0}$ in $\mathcal{M}$.

We show that W is a subset of the almost-sure region. Let s be state s W and let τ be an optimal strategy for the mean-payoff game, by [LL69, Gil57] we know that this strategy can be chosen positional, let also $\mathcal{M}[\tau]$ the Markov chain induced by τ. Since $\mathcal{M}[W]$ is a sub Markov decision process, we know that any play consistent with τ will almost-surely reach a closed component C in $\mathcal{M}[\tau]$ we also know that for every state c C we have:

$$\mathbb{E}_c^{\tau}\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i)\right] > 0 \ ,$$

since c is recurrent using the strong law of large numbers (c.f. Theorem 2.12) we get that:

$$\mathbb{P}_c^{\tau}\left(\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i) > 0\right) = 1 \ ,$$

which shows that $\mathcal{M}[W]$ is almost-sure.

We show that any state s W is not almost-sure. Let s be a state not in W and let σ be a positional strategy. We show that

$$\mathbb{P}_s^{\sigma}(\mathrm{Avg}_{>0}) < 1 \tag{3.5}$$

Let $\mathcal{C}$ be the set of closed classes reachable from s in the Markov chain $\mathcal{M}[\sigma]$. Let C $\mathcal{C}$ be a closed class and c C be a state. Assume that

$$\mathbb{E}_c^{\sigma}\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i) > 0\right] > 0 \ ,$$

it follows that for every state c C we have

$$\mathbb{E}_{c'}^{\sigma}\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i) > 0\right] > 0 \ ,$$

which implies that $C \subseteq W$, thus there exists C $\mathcal{C}$ and c C such that

$$\mathbb{E}_c^{\sigma}\left[\limsup_n \frac{1}{n}\sum_{i=0}^{n-1} r(S_i) > 0\right]] \leq 0 \ ,$$

hence

$$\mathbb{P}_c^{\sigma}(\mathrm{Avg}_{>0}) = 0 \ ,$$

and because c is accessible from s we obtain

$$\mathbb{P}_s^{\sigma}(\mathrm{Avg}_{>0}) < 1 \ ,$$

which terminates the proof.

The polynomial running time complexity is a consequence of the fact the value of state for mean-payoff objectives can be computed in polynomial time [Put94]. □

Corollary 3.33. *In any Markov decision process $\mathcal{M}$ we have:*

$$s \quad S,\ \mathrm{Val}_{\mathrm{Avg}_{>0}}(s) = \mathrm{Val}_{\underline{\mathrm{Avg}}_{>0}}(s)\ .$$

The proof of this corollary is postponed to Chapter 4 where it is proved for a larger class of objectives (c.f. Proposition 4.13).

Part II

Perfect Information Setting

Chapter 4

Multi Objectives Markov Decision Processes

Contents

Abstract We study Markov decision processes equipped with parity and positive-average conditions. In these setting, the goal of the controller is to maximize the probability that both the parity and the positive-average conditions are fulfilled. We show that the values of these games are computable in polynomial time. We also show that optimal strategies exist, require only finite memory and can be effectively computed.

4.1 Introduction

To perform at the same time both qualitative and quantitative verification of reactive systems, it is necessary to consider combinations of parity and mean-payoff conditions. This has been done in several papers about *non-stochastic* games. In [CHJ05], mean-payoff parity games were considered and solved.

Lately, there has been also several papers about *energy games* [CDHR10, CD10]. Max is declared to be the winner in an energy game if her payoff never goes below 0. The relationship between this winning condition and mean-payoff objective is straightforward in the case of non-stochastic games but breaks in the case of stochastic games.

Yet another class of games called *priority mean-payoff games*, which generalize both mean-payoff and parity games were introduced and solved in [GZ06, GZ07b, GZ07a].

Our initial motivation is to generalize the the result of [CHJ05] where the mean-payoff parity objective was introduced. In [CHJ05], the value of a state s with respect to mean-payoff parity condition is defined by supremum payoff that Max can ensure along a run which satisfies the parity condition and equal to $-$ if the parity objective cannot be achieved from s. The purpose of such objective is the fact that mixing these two objectives is useful when one wants to perform at the same time verification of qualitative properties such as fairness and quantitative properties such as energy resources.

In order to extend this result to a stochastic setting, one needs to slightly modify the winning condition and call it *parity and positive-average*. The value of a state s with respect to parity and positive-average objective is the supremum probability that a run that starts in state s achieves the parity objective and the long term average payoff along this run is strictly positive. We study the value problem and the memory requirements for optimal strategies.

Contribution and results Our main result concerns the construction of almost-sure strategies for parity and positive-average objectives. We show that the set of almost-sure states can be computed in polynomial time and that an exponential size memory is sufficient and necessary to win almost-surely. Our algorithm for the of computation of the almost-sure region is based on an inductive characterization of the almost-sure region inspired from Zielonka s construction for parity games as opposed to the one that appeared in [CD11] where the computation of the almost-sure region relies on a fine end-component analysis. The advantage of our approach is that a very small modification of the characterization allows us to extend our result to the case of stochastic games.

Outline of the chapter

- In Section 4.2 we explicit the construction of the almost-sure regions and give a polynomial time algorithm to compute the value of each state.
- In Section 4.3 we show that exponential size memory is sufficient and necessary to implement optimal strategies.
- In Section 4.5 we show how to solve objectives that consists of boolean combination of quantitative objectives.

4.2 Computing the values

In this section we consider Markov decision processes equipped with $\mathrm{Par}\quad \mathrm{Avg}_{>0}$ winning condition. We give a polynomial time algorithm that computes the value of each state.

4.2.1 Characterizing the Almost-sure Regions

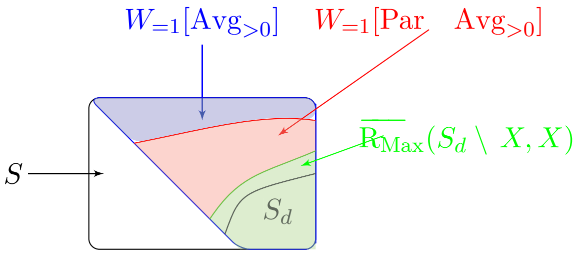
We characterize the winning regions by induction on the priorities available in the arena. The two following lemmata characterize the almost-sure regions when the highest priority is even (Lemma 4.1) and when the highest priority is odd (Lemma 4.2). This construction is inspired from Zielonka s construction [Zie04] for solving parity games.

Lemma 4.1. *Let $\mathcal{M}$ be a Markov decision process, $r : S \quad \mathbb{R}$ be a reward function, and $\chi : S \quad C$ be a priority function. Assume that the highest priority d is even, then the almost-sure region for the objective $\mathrm{Par}\quad \mathrm{Avg}_{>0}$ is the largest set $W \subseteq S$ such that:*

1. *$\mathcal{M}[W]$ is a sub Markov decision process of $\mathcal{M}$,*

2. Max *wins almost surely the* $\mathrm{Avg}_{>0}$ *objective played in* $\mathcal{M}[W]$*,*

3. Max *wins almost surely the* $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ *objective played in* $\mathcal{M}[W \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus W, W)]$.


$W_{=1}[\mathrm{Avg}_{>0}]$
$W_{=1}[\mathrm{Par} \quad \mathrm{Avg}_{>0}]$
$\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus X, X)$
S
S_d

We want the largest sub Markov decision process $\mathcal{M}[U]$ such that

$$W_{=1}\left[\mathrm{Avg}_{>0}\right] = U \ ,$$

and

$$W_{=1}\left[\mathrm{Par} \quad \mathrm{Avg}_{>0}\right] = U \ \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus X, X) \ .$$

Figure 4.1: Construction of the almost-sure region when the highest priority is even.

Proof. To prove this lemma we show the following:

i) Any set $X \subseteq S$ satisfying 1, 2 and 3 is almost-sure.

ii) The almost-sure region satisfies 1, 2 and 3.

Let τ be an almost-sure strategy for the $\mathrm{Avg}_{>0}$ objective played in $\mathcal{M}$.

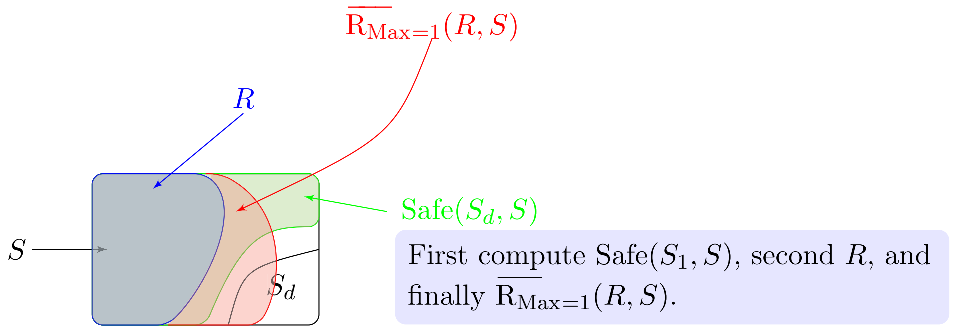
We start by proving i). Let X be a subset of S and assume that X satisfies 1, 2 and 3. We exhibit an almost-sure strategy σ for Max from any state in X. Roughly speaking, an almost-sure strategy is defined as follows: if the play is in $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus X, X)$, Max applies a strategies to attempt a visit to a priority-d state(attractor strategy) for X steps, then switches to an almost-sure strategy for the positive-average objective until her accumulated average reward goes above some well chosen threshold. Then she either starts these two steps again or in case the play is in $\mathcal{M}[X \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus X, X)]$, Max applies an almost-sure strategy in this sub Markov decision process. The proof of *i)* is postponed to the next chapter where we show that the same strategy is almost-sure for the same objective in the setting of two players stochastic games (c.f. Lemma 5.18).

Let us show (ii). Denote W the almost-sure region for $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ objective played in $\mathcal{M}$. We prove that W satisfies 1, 2 and 3. 1 holds because $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ is a tail objective. That W satisfies 2 follows from the fact that Max can win almost-surely $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ in $\mathcal{M}[W]$. To see that 3 holds, note that $\mathcal{M}[W \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus W, W)]$ is a trap for Max. So if she plays her almost-sure strategy σ defined on W, she wins almost-surely the $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ objective which shows (ii) and concludes the proof of the lemma. □

Lemma 4.2. *Let $\mathcal{M}$ be a Markov decision process, $r : S \quad \mathbb{R}$ be a reward function, and $\chi : S \quad C$ be a priority function. Assume that the highest priority d is odd, then the almost-sure region for the objective* $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ *is*

$$\overline{\mathrm{R}}_{\mathrm{Max}=1}(R, S) \ ,$$

where R is the almost-sure winning region for the $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ *game played in the sub Markov decision process* $\mathcal{M}[\mathrm{Safe}(S_1, S)]$.



First compute $\mathrm{Safe}(S_1,S)$, second R, and finally $\overline{\mathrm{R}}_{\mathrm{Max}=1}(R,S)$.

Figure 4.2: Construction of the almost-sure region when the highest priority is odd.

Proof. Let $\mathcal{M}$ be a Markov decision process and let R be the almost-sure region for the Par Avg$_{>0}$ game played in the sub Markov decision process $\mathcal{M}[\mathrm{Safe}(S_d,S)]$. We show that from any state in $W = \overline{\mathrm{R}}_{\mathrm{Max}=1}(R,S)$ Max has an almost-sure strategy for the Par Avg$_{>0}$ objective. Max applies the following strategy. As long as a play has not reached R, Max plays her attractor strategy π induced by $\overline{\mathrm{R}}_{\mathrm{Max}=1}(R,S)$. If the play is in R, she uses her almost-sure strategy, τ, in R. Formally,

$$\sigma : S - \quad S$$

$$\sigma(s) = \begin{cases} \pi(s) \text{ if } s \quad S \\ \tau(s) \text{ if } s \quad S \end{cases}$$

This strategy is almost-sure since any play consistent with it eventually reaches the set R and stays there forever.

We now prove that the almost-sure region is exactly the set W, i.e. we show that Max cannot win almost-surely in $S \quad W$. Let σ be a strategy and $s \quad S \quad W$ be a state, then either

$$\mathbb{P}_s^{\sigma'}(\quad n \geq 0,\ S_n \quad S \quad \mathrm{Safe}(S_d,S)) > 0 \ , \tag{4.1}$$

or

$$\mathbb{P}_s^{\sigma'}(\ N \geq 0, \quad n \geq N,\ S_n \quad \mathrm{Safe}(S_d,S)) = 1 \ . \tag{4.2}$$

If Equation (4.1) holds, using the Borel-Cantelli Lemma we get that a state of priority 1 is visited infinitely many times, thus σ cannot be almost-sure.
If Equation (4.2) holds, it follows that ultimately the play stays in $\mathcal{M}[\mathrm{Safe}(S_d,S)]$. Denote $\mathcal{M}[S\ ]$ the largest trap induced by $S \ \subseteq \mathrm{Safe}(S_d,S) \quad R$, according to the almost-sure property (c.f. Theorem 3.21), it follows that

$$\mathbb{P}_s^{\sigma'}(\mathrm{Par} \quad \mathrm{Avg}_{>0} \quad n \geq 0,\ S_n \quad S\ ) = 0 \ ,$$

and since $s \quad \overline{\mathrm{R}}_{\mathrm{Max}=1}(R,S)$, according to Proposition 3.17 we have:

$$\mathbb{P}_s^{\sigma'}(\ n \geq 0,\ S_n \quad R) < 1 \ ,$$

thus

$$\mathbb{P}_s^{\sigma'}(\mathrm{Par} \quad \mathrm{Avg}_{>0} \quad n \geq 0,\ S_n \quad \mathrm{Safe}(S_d,S)) < 1 \ ,$$

which shows that W is the largest set from where Max wins almost-surely the Par Avg$_{>0}$ objective, which concludes the proof of the lemma. □

4.2.2 Algorithm

We are now ready to state the main theorem of this section.

Theorem 4.3. *Let $\mathcal{M}$ be a Markov decision process, the almost-sure region for the objective* Par $\mathrm{Avg}_{>0}$ *is computable in* polynomial *time.*

To prove the theorem, we give an algorithm that computes the almost-sure region. Intuitively, our algorithm, starts by reducing the objective Par $\mathrm{Avg}_{>0}$ to a disjunction of other objectives say $\Phi_1, \cdots, \Phi_n$, and solve each one of them, then outputs the almost-sure region for the original objective as the almost-sure region for the objective $\Phi_1 \quad \cdots \quad \Phi_n$.

Algorithm 1 Computes the almost-sure region for the objective Φ_d.

```
$Q \quad$ Safe$(S_3, S)$
In the Markov decision process $\mathcal{M}[Q]$ compute $R$ the almost-sure region for the objective $\mathrm{Avg}_{>0}$

repeat
    In the Markov decision process $\mathcal{M}[R]$ compute $R$ the almost-sure region for the objective Büchi$(S_2)$.
    $R \quad R \quad R$
    $R \quad$ Safe$(R \ , R)$
until $R \ =$
return $\overline{\mathrm{R}}_{\mathrm{Max}=1}(R \ , S)$
```

Proof. To compute the values in polynomial time we use similar technics as in the proof of Theorem 3.28. For each even priority $d \quad C$, we create a new coloring function and a new Markov decision process $(\mathcal{M}, \Phi_d, r, \chi \ )$ where:

- $\mathcal{M}$ is the original Markov decision process,
- Φ_d is the new Par $\mathrm{Avg}_{>0}$ objective obtained accordingly to $\chi \ $,
- $r : S \quad \mathbb{R}$ is the original reward function,
- $\chi \ : S \quad 1, 2, 3$ is the new coloring function obtained the following way:

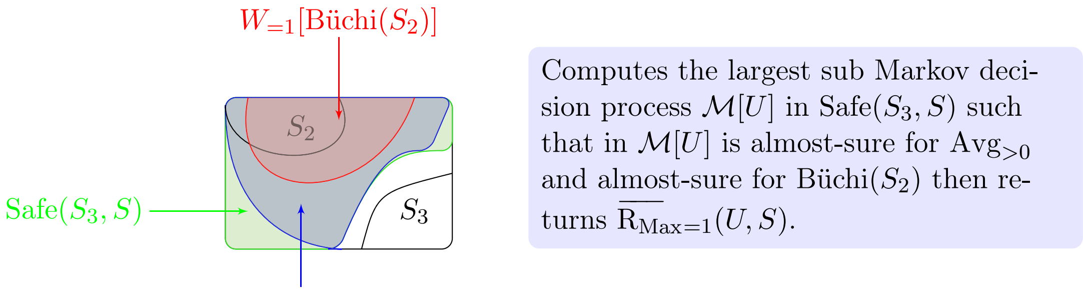
$$s \quad S, \ \chi(s) = \begin{cases} 1 \text{ if } \chi(s) < d \ , \\ 2 \text{ if } \chi(s) = d \ , \\ 3 \text{ if } \chi(s) > d \ . \end{cases}$$

To solve each of these objectives we use the procedure described in Algorithm 1, in Fig 4.3 we depict a rough idea of how Algorithm 1 proceeds. .

Let us show that Algorithm 1 is correct. First it considers the largest sub Markov decision process which is almost-sure for the objectives Büchi(S_2) and $\mathrm{Avg}_{>0}$ namely $\mathcal{M}[R \ ]$, note that $\mathcal{M}[R \ ]$ satisfies the conditions of Lemma 4.1. Finally it uses Lemma 4.2 to compute the almost-sure region.

Using the fact that the original objective can be rewritten as the disjunction of all the Φ_d, Theorem 3.24 shows that the almost-sure region is given by $\overline{\mathrm{R}_{\mathrm{Max}=1}}(\bigcup_{d \quad D} W_{=1}[\Phi_d])$, where D is the set of even priorities.

We now argue on the running time complexity. Each Φ_d can be solved in polynomial time since Büchi objectives can be solved in polynomial time [dAH00, CJH03] as well as computing the set of attractors and the almost-sure region for $\text{Avg}_{>0}$(c.f. Theorem 3.31). It follows that our procedure runs in $O(D \cdot L)$ where L is the time one needs to solve each Φ_d. □



Computes the largest sub Markov decision process $\mathcal{M}[U]$ in $\text{Safe}(S_3, S)$ such that in $\mathcal{M}[U]$ is almost-sure for $\text{Avg}_{>0}$ and almost-sure for $\text{Büchi}(S_2)$ then returns $\overline{\text{R}_{\text{Max}=1}}(U, S)$.

Figure 4.3: Construction of the almost-sure region $W_{=1}[\Phi_d]$.

From Theorem 4.3 and by Corollary 3.22, we get the following corollary.

Corollary 4.4. *In any Markov decision process $\mathcal{A}$ where the winning condition is* $\text{Par} \quad \text{Avg}_{>0}$*, the values are computable in* polynomial *time.*

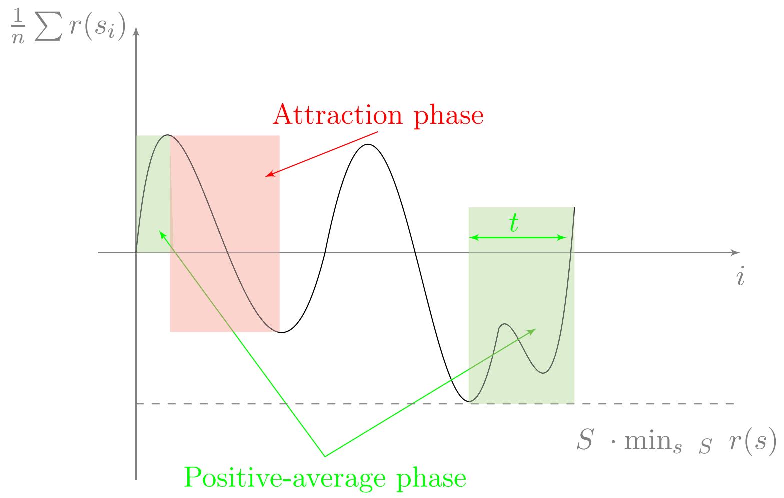


Figure 4.4: The average payoff along a play consistent with σ.

Memory for almost-sure winning We conclude this section by a discussion on the memory that an almost-sure strategy may require. The graphic in Fig 4.4 depicts the average reward accumulated along a play consistent with the strategy σ described in Lemma 4.1 We recall that σ applies in turn the positive average strategy and the attractor strategy.

In order to know when to switch between these two strategies, σ has to keep track of average payoff accumulated along the play. As shown in Fig 4.4, let t be the time that σ spends in the positive average phase. Since the attraction phase requires a bounded memory, it follows that in order to get a finite memory strategy one needs to bound t from above. But as shown in Fig 4.4, the average payoff along a play can fluctuate considerably before reaching the value that allows the switch, for instance the Markov decision process depicted in Fig 4.6 show that in order to switch from the attraction strategy to the positive-average strategy, Max has to win n successive coin tosses. Thus this time t cannot be bounded from above which makes the memory needs infinite.

4.3 Implementing optimal strategies with finite memory

In this section, we take a closer look at the memory needed by Max to win almost-surely. Our goal is to slightly modify the strategy described in Section 4.2 in order to implement almost-sure strategies with finite memory. The main idea, is to consider the expected average reward rather than the actual average reward. The advantage of this approach is that one can estimate the value of the expected average reward at time t along a play consistent with a given strategy. Fig 4.5 is slight modification of Fig 4.4, where σ is the almost-sure strategy described in Section 4.2.

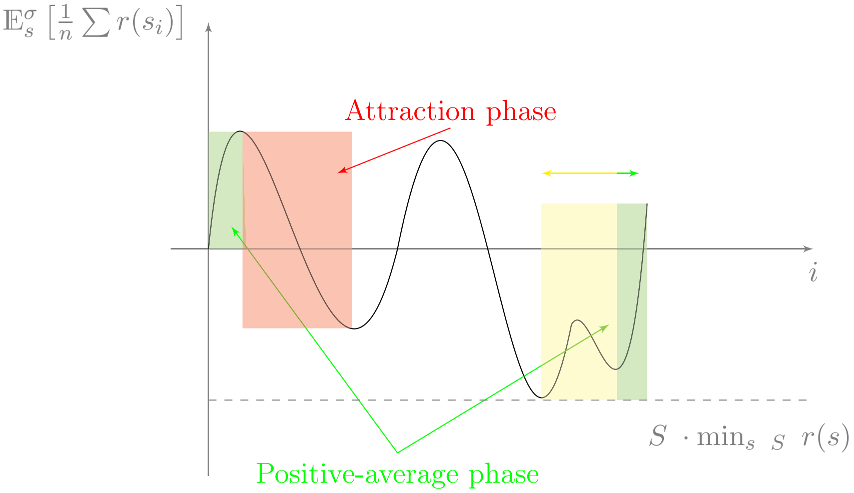


Figure 4.5: The expected average payoff along a play consistent with σ.

The difference in this figure is that instead of keeping track of the accumulated average reward, Max keeps track of the *expected* accumulated average reward. The advantage of this approach over the previous one is that focusing on the expected value one can bound the time Max applies her positive-average strategy before switching and thus the memory obtained for such a strategy is finite. This follows from the fact that since the positive-average strategy applied is positional, any play consistent with it is similar to an execution of a finite state Markov chain. Thus the play will first visit transient states, then eventually will reach a closed class. Now, the finiteness of the memory follows from the following facts:

a) Since the play is consistent with an almost-sure strategy it is not possible for the expected average reward in this closed class to be decreasing,

b) in each closed component, the expected average reward between two consecutive visits of the same state can be bounded from bellow.

We state formally the main result of this chapter.

Theorem 4.5. *Let $\mathcal{M}$ be a Markov decision process with state space S. A memory of size $O(2^{\ S\ })$ is sufficient and necessary to implement an almost-sure strategy for the objective* $\mathrm{Par} \quad \mathrm{Avg}_{>0}$.

The proof of Theorem 4.5 goes through 3 steps:

1. We show that if one has a bound on the time the expected average reward needs to go above a certain value, then one can implement a finite memory almost-sure strategy.

2. We show that a memory of size exponential is sufficient.

3. We show that a memory of size exponential is necessary.

4.3.1 Existence of Finite Memory Optimal Strategies

To establish the existence of finite memory strategies, we define the notion of total-reward objective and prove the following lemma.

Definition 4.6 (Total-reward objective)**.** *Let $\mathcal{M}$ be a Markov decision process with state space S and reward function $r : S \quad R$, the total-reward objective is defined by the following set of plays:*

$$\mathrm{Rwd}_{=} \quad = \left\{ s_0 a_0 s_1 a_0 \cdots \quad S(AS)^{\omega} \quad \liminf_n \sum_{i=0}^{n-1} r(S_i) = \quad \right\} .$$

The next lemma shows the relationship between the total-reward objective defined above and the positive-average objective (see Definition 3.31).

Lemma 4.7. *Let $\mathcal{M}$ be Markov decision process, then* Max *has a positional strategy σ such that*

$$s \quad W_{=1}[\mathrm{Avg}_{>0}],\ \mathbb{P}_s^{\sigma}\left(\mathrm{Rwd}_{=} \quad\right) = 1 \ . \tag{4.3}$$

Proof. The winning condition $\mathrm{Avg}_{>0}$ is submixing and tail, hence there exists a positional optimal strategy [Gim07]. Therefore, there exists a positional almost-sure strategy. Thus by Corollary 3.33, σ is almost-sure for $\underline{\mathrm{Avg}}_{>0}$ as well. Hence the following equation holds,

$$s \quad W_{=1}[\mathrm{Avg}_{>0}],\ \mathbb{P}_s^{\sigma}\left(\liminf_n \frac{1}{n+1}\sum_{i=0}^{n} r(S_i) > 0\right) = 1 \ .$$

The same strategy σ yields (4.3). □

Now that the lemma is proved, we prove item 1. This is done thanks to Proposition 4.8.

Proposition 4.8. *Let $\mathcal{M}$ be a Markov decision process,* Max *has a finite memory almost-sure strategy.*

Proof. Let $\mathcal{M}$ be a Markov decision process with a reward function $r : S \quad \mathbb{R}$ and a coloring function $\chi : S \quad C$. We prove by induction on the number of priorities that Max has an almost-sure strategy with finite memory. Suppose that $C \; = 1$ and let c be the only priority of C. If c is even then Max plays a positive-average objective, according to Theorem 3.32, there exists a positional optimal strategy for Max. If c is odd then Max has no winning strategy.

Suppose that Max can win almost-surely using finite memory in any Markov decision process which contains less than d priority. Let $\mathcal{M}$ be a Markov decision process with d priorities.

If the highest priority d is *odd*. According to Lemma 4.2, to win Max applies her attractor strategy until she reaches the almost-sure region for the game Par $\quad$ Avg$_{>0}$ played in the sub Markov decision process $\mathcal{M}[\mathrm{Safe}(S_d, S)]$. Note that in $\mathcal{M}[\mathrm{Safe}(S_d, S)]$ the number of priorities is strictly less than d and thus she has a finite memory strategy. Since the attraction strategy is positional, Max has a finite memory strategy to win almost-surely if the highest d priority is odd.

If the highest priority d is *even*. According to Lemma 4.1, either Max is playing in the almost-sure region for Par $\quad$ Avg$_{>0}$ in the sub Markov decision process $\mathcal{M}[S \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_d, S)]$ or the play visits $\overline{\mathrm{R}_{\mathrm{Max}}}(S_d, S)$. In the former case, by induction, Max has a finite memory strategy to win and the proof is done. In the latter case she applies her attractor strategy π for a specified time, then she switches to her positive-average strategy τ. In the remaining of this proof, we are going to show that the time Max should apply τ can be bounded.

Let us describe how Max decides the time she applies τ.

– Apply τ until a recurrent state r in the Markov chain $\mathcal{M}[\tau]$ is visited.

– Whenever r is reached increment a counter.

– When the counter reaches a well chosen value switch.

The almost-sure strategy σ, the memory and the update function are formalized as follows; Let R be the set of all recurrent states in the Markov chain $\mathcal{M}[\sigma]$ and let T be the random variable with value in $\mathbb{N}$ $\quad$ that gives the time needed to reach a state $r \quad R$ plus the time r should be visited before switching. For each even priority d we need the following memory $M_d = S \times \; 0, 1, 2 \; \times \; 0, \cdots, \; S \; - 1 \; \times \; 0, \cdots, T$. Let Update : $S \times M_d \quad M_d$ be the update function defined as follows:

$$\mathrm{Update}(s, (r, b, i, j)) = \begin{cases} (r, 0, i, j+1) \text{ if } (b = 0) \quad (j < \; S \; - 1) \quad (\chi(s) = d) \; . \\ (r, 1, i, j) \text{ if } (b = 0) \quad [(j = \; S \; - 1) \quad (\chi(s) = d)] \; . \\ (r, 1, i, j) \text{ if } (b = 1) \quad (s \quad R) \; . \\ (s, 2, 0, j) \text{ if } (b = 1) \quad (s \quad R) \; . \\ (r, 2, i+1, j) \text{ if } (b = 2) \quad (s = r) \quad (i < T) \; . \\ (r, 2, i, j) \text{ if } (b = 2) \quad (s = r) \quad (i < T) \; . \\ (r, 0, i, 0) \text{ if } (i = T_n) \; . \end{cases}$$

The strategy $\sigma : S \times M_d \quad S$ consists in applying π the attractor strategy whenever $b = 0$ and applying τ the Avg$_{>0}$ strategy whenever $b = 0$.

We show that $T <$ Since $\mathcal{M}[\tau]$ is a finite state Markov chain it follows that

$$\mathbb{P}_s^\tau(\quad 0 \leq n < \quad , S_n \quad R) = 1 \ , \tag{4.4}$$

and according to Lemma 4.7 we know that

$$\mathbb{P}_s^\tau(\mathrm{Rwd}_{=} \quad) = 1 \ , \tag{4.5}$$

thus Lemma 2.29 applies and we have for every recurrent state r, there exists $\eta > 0$ such that

$$\mathbb{E}_r^\tau \left[\frac{1}{n+1} \sum_{i=0}^{n} r(S_i) \,\middle|\, T_r = n \right] \geq \eta \ , \tag{4.6}$$

where

$$T_r = \min \quad n \geq 1 \quad S_n = r \quad .$$

It follows that there exists $0 \leq m <$ such that

$$\mathbb{E}_s^\tau \left[\frac{1}{m+1} \sum_{i=0}^{m} r(S_i) \,\middle|\, T = m \right] > \eta \ , \tag{4.7}$$

which shows that Max can use a finite memory to decide when to switch.

We show that σ is almost-sure for the objective Par $\quad \mathrm{Avg}_{>0}$. First notice that a very similar argument as the one used in the proof of Lemma 5.18 shows that the parity objective is achieved almost-surely.

We show now that the $\mathrm{Avg}_{>0}$ objective is achieved almost-surely as well.

Since Equation 4.7 holds. Repeating this argument each time Max switches from strategy τ to the attraction strategy, we build a sequence of random variable $T^{(1)}, T^{(2)}, \cdots$ with values in $\mathbb{N}$ and a rational $\eta > 0$ such that:

$$i \geq 1, \ \mathbb{E}_s^\tau \left[\frac{1}{n_i+1} \sum_{k=0}^{n_i} r(S_k) \,\middle|\, T^{(i)} = n_i \right] \geq \eta \ ,$$

$$= \quad \limsup_n \mathbb{E}_s^\tau \left[\frac{1}{n} \sum_{k=0}^{n-1} r(S_k) \right] \geq \eta \ , \tag{4.8}$$

$$= \quad \mathbb{E}_s^\tau \left[\limsup_n \frac{1}{n} \sum_{k=0}^{n-1} r(S_k) \right] \geq \eta \ . \tag{4.9}$$

Where the transformation from (4.8) to (4.9) is by Fatou s lemma 2.7. Using the strong law of large numbers (c.f Theorem 2.12) we get that the positive-average objective is ensured almost-surely. This shows that Max has finite memory almost-sure strategy for the objective Par $\quad \mathrm{Avg}_{>0}$, Proposition 4.9 gives an upper bound on the size of this memory. □

4.3.2 Sufficiency of Exponential Size Memory

Next step toward the proof of Theorem 4.5 is to show that an exponential size memory is sufficient, this would prove item 2.

Proposition 4.9. *Let $\mathcal{M}$ be a Markov decision process, memory of size exponential in the size of $\mathcal{M}$ is sufficient to implement an almost-sure strategy for the objective* Par $\quad \mathrm{Avg}_{>0}$.

Proof. Let $\mathcal{M}$ be a Markov decision process, denote $\mathcal{M}[\tau]$, $\mathcal{M}[\pi]$ the Markov chains induced by τ the almost-sure strategy for the objective $\text{Avg}_{>0}$ and π the attractor strategy respectively. We define the following random variables,

- T_R: with values in $\mathbb{N}$, is the absorption time in recurrent state of $\mathcal{M}[\tau]$ (c.f. Section 2.3).
- T_n: with values in $\mathbb{N}$, is the time needed to reach a state $r \quad R$ plus the time r is visited n times.
$$T_n = \min \quad n \geq 0 \qquad (i_0, \cdots, i_n),\ (S_{i_0} \quad R) \quad (S_{i_0} = \cdots = S_{i_n}) \quad .$$

Note that if all the rewards in $\mathcal{M}$ are strictly positive, Max plays only for the parity objective. Hence no memory is required (c.f. Theorem 3.28).

Assume that there exist negative rewards in $\mathcal{M}$. We want to compute an upper bound for T_n such that the objective $\text{Avg}_{>0}$ is achieved.

$$\frac{1}{T_n}\sum_{i=0}^{T_n-1} r(S_i) = \frac{1}{T_n}\left[\sum_{i=0}^{\ S\ -1} r(S_i) + \sum_{i=\ S}^{T_0-1} r(S_i) + \sum_{j=0}^{n-1}\sum_{i=T_j}^{T_{j+1}-1} r(S_i)\right] \quad .$$

Let

- $A = \sum_{i=0}^{\ S\ -1} r(S_i)$.
- $B = \sum_{i=\ S}^{T_0-1} r(S_i)$.
- $C_j = \sum_{i=T_j}^{T_{j+1}-1} r(S_i)$.

Hence for every $s \quad S$

$$\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=0}^{T_n-1} r(S_i)\right] = \mathbb{E}_s^\sigma\left[\frac{A}{T_n} + \frac{B}{T_n} + \frac{\sum_{j=0}^{n-1} C_j}{T_n}\right] \quad .$$

We first compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{A}{T_n}\right]$.

$$\frac{1}{T_n}\sum_{i=0}^{\ S\ -1} r(S_i) = \frac{\ S\ }{T_n}\frac{1}{\ S\ }\sum_{i=0}^{\ S\ -1} r(S_i) \geq \frac{\ S\ }{n}\min_{s\ \ S}\ r(s) \quad .$$

Where the inequality holds because $T_n \geq n$ and $\min_{s\ \ S}\ r(s)$ is negative. Hence

$$\mathbb{E}_s^\sigma\left[\frac{A}{T_n}\right] \geq \frac{\ S\ }{n}\min_{s\ \ S}\ r(s) \quad . \tag{4.10}$$

Next, we compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{B}{T_n}\right]$

$$\begin{aligned}\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=\ S}^{T_0-1} r(S_i)\right] &= \mathbb{E}_s^\sigma\left[\frac{T_0 - \ S\ }{T_n}\frac{1}{T_0 - \ S\ }\sum_{i=\ S}^{T_0-1} r(S_i)\right] \\ &\geq \mathbb{E}_s^\sigma\left[\frac{T_0 - \ S\ }{T_n}\min_{s\ \ S}\ r(s)\ \right] \\ &\geq \mathbb{E}_s^\sigma\left[\frac{T_0 - \ S\ }{n}\min_{v\ \ V}\ r(v)\ \right] = \frac{\mathbb{E}_s^\sigma\left[T_0 - \ S\ \right]}{n}\min_{s\ \ S}\ r(s)\end{aligned}$$

Where the first inequality holds because $T_n \geq n$ and $\min_{s \quad S} \; r(s)$ is negative. Hence

$$\mathbb{E}_s^\sigma\left[\frac{B}{T_n}\right] \geq \frac{\mathbb{E}_s^\sigma\left[T_0 - \quad S\;\right]}{n} \min_{s \quad S} \; r(s) \quad . \tag{4.11}$$

Finally, we compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{\sum_{j=0}^{n-1} C_j}{T_n}\right]$.

$$\begin{aligned}
\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=T_0}^{T_n-1} r(S_i)\right] &= \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{1}{T_n}\sum_{i=T_j}^{T_{j+1}-1} r(S_i)\right] \\
&= \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{T_{j+1}-T_j}{T_n}\frac{1}{T_{j+1}-T_j}\sum_{i=T_j}^{T_{j+1}-1} r(S_i)\right] \\
&= \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{T_{j+1}-T_j}{T_n}\mathbb{E}_s^\sigma\left[\frac{1}{T_{j+1}-T_j}\sum_{i=T_j}^{T_{j+1}-1} r(S_i) \,\middle|\, \mathcal{F}_{T_j}\right]\right] \\
&= \mathbb{E}_s^\sigma\left[\frac{T_n-T_0}{T_n}\mathbb{E}_s^\sigma\left[\frac{1}{T_1-T_0}\sum_{i=T_0}^{T_1-1} r(S_i) \,\middle|\, \mathcal{F}_{T_0}\right]\right] \qquad (4.12) \\
&\geq \eta\,\mathbb{E}_s^\sigma\left[1-\frac{T_0}{T_n}\right] \qquad (4.13) \\
&\geq \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_0\right]}{n}\right) \qquad (4.14) \\
&\geq \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_0 - \quad S\;\right] + \quad S}{n}\right)
\end{aligned}$$

Where the transformation from (4.12) to (4.13) holds because according to Lemma 2.29:

$$\eta > 0,\ \mathbb{E}_s^\sigma\left[\frac{1}{T_1-T_0}\sum_{i=T_0}^{T_1-1} r(S_i) \,\middle|\, \mathcal{F}_{T_0}\right] \geq \eta \quad ,$$

and from (4.13) to (4.14) because $T_n \geq n$. Hence,

$$\mathbb{E}_s^\sigma\left[\frac{\sum_{j=0}^{n-1} C_j}{T_n}\right] \geq \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_0 - \quad S\;\right] + \quad S}{n}\right) \quad . \tag{4.15}$$

From (4.10), (4.11) and (4.15) we get

$$\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=0}^{T_n-1} r(s_i)\right] \geq \frac{\quad S}{n}m + \frac{\mathbb{E}_s^\sigma\left[T_R\right]}{n}m + \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_R\right] + \quad S}{n}\right) \quad .$$

Let us find a value for n such that

$$\frac{m}{n}\left(\quad S \quad + \mathbb{E}_s^\sigma\left[T_R\right]\right) + \frac{\eta}{n}\left(n - \mathbb{E}_s^\sigma\left[T_R\right] + \quad S\;\right) > 0 \quad .$$

We find

$$n > \mathbb{E}_s^\sigma[T_R] + \ S \ - \frac{m}{\eta} (\ V \ + \mathbb{E}_s^\sigma[T_R]) \ \ .$$

According to Lemma 2.29, we know that there exists a polynomial Q such that $\eta \geq 2^{-Q(\ \mathcal{M}[\tau] \ )}$ where $\mathcal{M}[\tau]$ is the description of the Markov chain $\mathcal{M}[\tau]$, hence

$$n \geq \mathbb{E}_s^\sigma[T_R] + \ S \ - m 2^{Q(\ \mathcal{M}[\tau] \ )} (\ S \ + \mathbb{E}_s^\sigma[T_R]) \ \ .$$

We compute an upper bound for $\mathbb{E}_s^\sigma[T_R]$. Using Lemma 2.26, we get that this quantity is at most exponential in the description of the Markov decision process $\mathcal{M}$. It follows that exponential size memory is sufficient. □

4.3.3 Exponential Size Memory is Necessary

Last step in the proof of Theorem 4.5 is to show item 3. This is done in Proposition 4.10

Proposition 4.10. *Let $\mathcal{M}$ be a Markov decision process, a memory of size exponential in the size of $\mathcal{M}$ is necessary to implement an almost-sure strategy.*

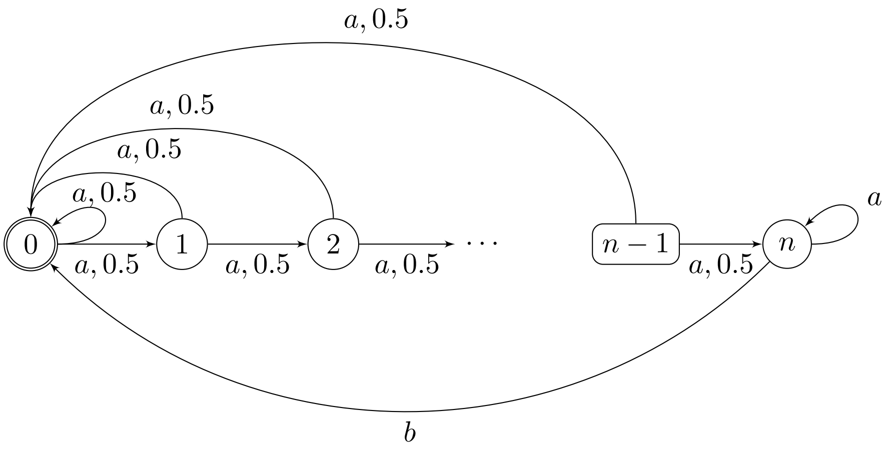


Figure 4.6: Max needs a memory of size exponential to achieve almost-surely the objective Par Avg$_{>0}$.

Proof. Consider the Markov decision process $\mathcal{M}$ depicted in Fig 4.6 where:

- The set of states is $S = \ \ 0, 1, \cdots, n \ \ ,$
- the set of action is $A = \ \ a, b \ \ \ ,$
- the transition function is defined in Fig 4.6,
- the reward function is defined as follow:

$$i \quad\ 0, \cdots, n \quad r(i) = \begin{cases} -1 \text{ if } i = n \\ 1 \text{ if } i = n \end{cases}$$

Assume that a play is winning if the state 0 is visited infinitely often and the positive-average objective is achieved.

We show that memory of size exponential in the size of $\mathcal{M}$ is necessary to achieve the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ almost-surely. Notice that in the Markov decision process of Fig 4.6 Max wins almost-surely from any state $s \quad S$. Let σ be an almost-sure strategy with finite-memory of size k. Denote T_R the absorption time in state n in the Markov chain obtained by removing from $\mathcal{M}$ the action b. $\mathbb{E}_s^\sigma[T_R]$ gives the expected time to reach n. Thus the expected average reward for Max on the path from state 0 to state n is

$$\mathbb{E}_0^\sigma\left[\frac{1}{l+1}\sum_{i=0}^{l} r(S_i) \;\middle|\; S_k = n\right] = -\mathbb{E}_0^\sigma[T_R] \ .$$

We show by contradiction that $k \geq \mathbb{E}_v^\sigma[T_R]$. Since σ is almost-sure, any play consistent with σ cannot stay forever in state n almost-surely, thus it leaves state n after at most k loops. Thus the expected accumulated reward on a play from state 0 to 0 is

$$\mathbb{E}_0^\sigma\left[\frac{1}{l+1}\sum_{i=0}^{l} r(S_i) \;\middle|\; S_k = 0\right] \leq k - \mathbb{E}_s^\sigma[T_R] \ .$$

If this value is negative then according to the law of large numbers, the expected average reward will almost-surely be negative as well, a contradiction, hence

$$k > \mathbb{E}_s^\sigma[T_R] \ .$$

Let us show that $\mathbb{E}_0^\sigma[T_R]$ is exponential in $\ S\ $. We know that for every state $0 \leq i \leq n-1$

$$\mathbb{E}_i^\sigma[T_R] = 1 + \frac{1}{2}\mathbb{E}_0^\sigma[T_R] + \frac{1}{2}\mathbb{E}_{i+1}^\sigma[T_R] \ .$$

and for $i = n$

$$\mathbb{E}_n^\sigma[T_R] = 0 \ .$$

Thus we get

$$\mathbb{E}_0^\sigma[T_R] = 2^n \sum_{i=0}^{n-1} \frac{1}{2^i} = 2^{n+1}\left(1 - \frac{1}{2^n}\right) \ .$$

Thus σ has a memory at least exponential in the size of the Markov decision process. □

We now conclude this section by putting things together and proving Theorem 4.5.

Proof of Theorem 4.5. Proposition 4.8 describes the shape of the finite memory almost-sure strategy, Proposition 4.9 shows that exponential memory in the size of the arena is sufficient and Proposition 4.10 shows that it is necessary. □

To conclude this section we use Theorem 3.21 that leads the following corollary:

Corollary 4.11. *Let $\mathcal{M}$ be a Markov decision process with state space S. Optimal strategies with memory of size $O(2^{\ S\ })$ for the objective* $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ *are sufficient and necessary.*

4.4 Solving Parity and Positive-average Objectives with lim inf semantics

In the previous section we studied parity and positive-average objectives with lim sup semantics. An alternative definition of these objectives is to replace lim sup by lim inf in the definition of the $\mathrm{Avg}_{>0}$ winning condition. We show that all results of the previous section hold for this alternative definition.

Definition 4.12. *Let $\mathcal{M}$ be a Markov decision process equipped with a reward function $r : S \to \mathbb{R}$. The objective $\underline{\mathrm{Avg}}_{>0}$ is:*

$$\underline{\mathrm{Avg}}_{>0} = \left\{ s_0 s_1 s_2 \cdots \in s^\omega \mid \liminf_{n} \frac{1}{n} \sum_{i=0}^{n-1} r(S_i) > 0 \right\} .$$

To compute the value of state for a Markov decision process equipped with $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$ objective, we use the previous result on optimality using finite memory and known results on Markov chains theory. Actually we show that the value of a state s for the objective $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$ is equal to the value of s for the objectives $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$.

Proposition 4.13. *In any Markov decision process $\mathcal{M}$ we have:*

$$\forall s \in S,\ \mathrm{Val}_{\mathrm{Par} \wedge \mathrm{Avg}_{>0}}(s) = \mathrm{Val}_{\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}}(s) .$$

Proof. We show that the following inequalities hold:

$$\forall s \in S,\ \mathrm{Val}_{\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}}(s) \leq \mathrm{Val}_{\mathrm{Par} \wedge \mathrm{Avg}_{>0}}(s) . \tag{4.16}$$

$$\forall s \in S,\ \mathrm{Val}_{\mathrm{Par} \wedge \mathrm{Avg}_{>0}}(s) \leq \mathrm{Val}_{\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}}(s) . \tag{4.17}$$

That (4.16) holds is trivial. It is a consequence of the fact that every winning strategy for $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$ is also winning for $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$.

To prove (4.17), notice that according to Corollary 4.4 Max can play optimally using finite memory in the $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ game, thus there exists a strategy $\sigma^\sharp$ which is optimal and with finite memory. Hence:

$$\begin{aligned} \mathrm{Val}_{\mathrm{Par} \wedge \mathrm{Avg}_{>0}}(s) &= \mathbb{P}_s^{\sigma^\sharp}(\mathrm{Par} \wedge \mathrm{Avg}_{>0}) \\ &= \mathbb{P}_s^{\sigma^\sharp}(\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}) \\ &\leq \sup_\sigma \mathbb{P}_v^{\sigma}(\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}) = \mathrm{Val}_{\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}}(s) , \end{aligned}$$

where the first equality is by definition of the value and the second is by Lemma 2.28. Therefore (4.17) holds and Proposition 4.13 is proved. □

Proposition 4.13 leads the following theorem.

Theorem 4.14. *In any Markov decision process $\mathcal{M}$ equipped with the objective $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$, the values are computable in* polynomial *time. Moreover memory of size exponential in the size of the arena is sufficient and necessary to implement optimal strategies.*

4.5 Towards Boolean Formulae of Objectives

In this section we tackle the problem of solving Markov decision process equipped with objective that consist of boolean combination of the objectives seen previously.

First of all notice that in the case of disjunctive formulae, the result follows from Theorem 3.24.

To study the case of conjunctive formulae we start by solving boolean combination of objectives consisting of positive-average only. We first study objectives consisting of conjunction of $\mathrm{Avg}_{>0}$ conditions. Second we study objectives consisting of $\underline{\mathrm{Avg}}_{>0}$ conditions. Finally, we mix the two previous conditions. Theorem 4.18 was obtained separately of the work published in [BBC+11], the technics we use are inspired by [Vel11].

Definition 4.15 (Generalized positive average objectives)**.** *Let $\mathcal{M}$ be a Markov decision process equipped with k reward function $r_i : S \quad \mathbb{R}$ for $1 \le i \le k$. The generalized positive-average winning condition is:*

$$\mathrm{Avg}^k_{>0} = \bigwedge_{i=1}^{k} \mathrm{Avg}^{(i)}_{>0} \ ,$$

where $\mathrm{Avg}^{(i)}_{>0}$ is the positive-average reward associated with the reward function r_i.

4.5.1 Solving conjunction of $\mathrm{Avg}_{>0}$

Theorem 4.16. *The almost-sure region of* Max *for the objective $\mathrm{Avg}^k_{>0}$ is given by the largest sub Markov decision process $\mathcal{M}[W]$ where $W \subseteq \bigcap_{i=1}^{k} W_{=1}[\mathrm{Avg}^i_{>0}]$.*

Proof. let $U \subseteq S$ such that $\mathcal{M}[U]$ is a sub Markov decision process where Max can almost-surely win the positive average objective induced by every reward function r_i for $1 \le i \le k$. We show that Max has an almost-sure strategy to win the objective $\mathrm{Avg}^k_{>0}$ in $\mathcal{M}[U]$. Let σ_i be the almost-sure strategy for $\mathrm{Avg}^i_{>0}$. Max applies the following strategy σ, play consistently with σ_1 until the accumulated average reward with respect to r_1 goes above a threshold $\eta_1 > 0$, then switches to σ_2 until a threshold η_2 is reached and so on, when the play is consistent with σ_k and the threshold η_k is reached Max restart from scratch.

We show that the strategy σ is almost-sure. The fact that $\mathcal{M}[U]$ is a sub Markov decision process ensures the fact that the play will never go outside of U and hence Max will always have the possibility to switch from one strategy to an other. To see that σ is almost-sure, notice that

$$s \quad U, \quad 1 \le i \le k,\ \mathbb{P}_s^{\sigma}\left(\quad n \ge 0,\ \frac{1}{n+1}\sum_{j=0}^{n} r_i(S_n) > 0\right) = 1 \ ,$$

hence

$$s \quad U, \quad 1 \le i \le k,\ \mathbb{P}_s^{\sigma}\left(\limsup_{n},\ \frac{1}{n+1}\sum_{j=0}^{n} r_i(S_n) > 0\right) = 1 \ ,$$

thus

$$s \quad U,\ \mathbb{P}_s^{\sigma}\left(\mathrm{Avg}^k_{>0}\right) = 1 \ .$$

We show that any state not in U is not almost-sure. Let $s \quad U$ be a state and let σ be a strategy, then either

$$\mathbb{P}_s^{\sigma'}(\quad n \ge 0, \quad a \quad A,\ p(S_n, a)(U) > 0) > 0 \ , \tag{4.18}$$

or

$$\mathbb{P}_s^{\sigma'}(\quad n \geq 0, \quad a \quad A,\ p(S_n,a)(U) = 0) = 1 \quad . \tag{4.19}$$

If Equation (4.18) holds, then assume that

$$\mathbb{P}_s^{\sigma'}(\quad n \geq 0, \quad a \quad A,\ p(S_n,a)(U) = 1) = 1 \quad ,$$

then $\mathcal{M}[S_n \quad U]$ is a sub Markov decision process in $\bigcap_{i=1}^{k} W_{=1}[\mathrm{Avg}^i_{>0}]$ which contradicts the fact that $\mathcal{M}[U]$ is the largest sub Markov decision process in $\bigcap_{i=1}^{k} W_{=1}[\mathrm{Avg}^i_{>0}]$, thus

$$\mathbb{P}_s^{\sigma'}(\quad n \geq 0, \quad a \quad A,\ p(S_n,a)(U) = 1) < 1 \quad .$$

If Equation (4.19) holds, then a play consistent with σ eventually reaches the largest trap contained in $S \quad U$ and since

$$s \quad S \quad U, \quad \tau, \quad 1 \leq i \leq k,\ \mathbb{P}_s^{\tau}(\mathrm{Avg}^i_{>0}) < 1 \quad ,$$

if follows that

$$s \quad S \quad U, \quad \tau,\ \mathbb{P}_s^{\tau}(\mathrm{Avg}^k_{>0}) < 1 \quad ,$$

which terminates the proof. □

4.5.2 Solving conjunction of lim inf

While solving conjunction of $\mathrm{Avg}_{>0}$ was straightforward, solving the conjunction of $\underline{\mathrm{Avg}}_{>0}$ requires a little more work. Indeed, we define a set of equations and show that a strategy that ensures the objective $\underline{\mathrm{Avg}}^k_{>0}$ exists if and only if the set of equations has a solution. In order to define this system of equations in a more convenient way, we assume without loss of generalities that a state s is one of the following cases:

- there exists an action $a \quad A$ such that $0 < p(s,a)(t) < 1$ for every $t \quad S$ and for every $b = a$ we have $p(s,b)(s) = 1$; we say that $s \quad S_R$.
- for every action in $a \quad A$ we have $p(s,a)(t) \quad\ 0, 1\ $ for every state $t \quad S$; we say that $s \quad S_M$.

This assumption does not restrict the model since

Proposition 4.17. *For every Markov decision process $\mathcal{M}$, one can compute in polynomial time a Markov decision process $\mathcal{M}$ such that the set of states of $\mathcal{M}$ is partitioned into S_R and S_M and for any tail objective Φ:*

$$s \quad S, \quad \sigma,\ \mathbb{P}_s^{\sigma}(\Phi) = 1 \qquad\quad s \quad S_M, \quad \sigma\ ,\ \mathbb{P}_s^{\sigma'}(\Phi) = 1 \quad .$$

Proof. Let $\mathcal{M}\ = (S_M, S_R, p\ : S_R \quad \Delta(S))$ be the Markov decision process constructed the following way:

- $S_M = S$,
- $S_R = \quad s_a \quad t \quad S,\ p(s,a)(t) > 0 \quad ,$
- $\quad s_a \quad S_R,\ p\ (s_a)(t) = p(s,a)(t).$

S_M is the set of original states, S_R the set of fresh states, and E the set of edges obtained. Let us show the direct implication. Let σ be a strategy and $s \quad S$ be a state such that:

$$\mathbb{P}^{\sigma}_{s}(\Phi) = 1 \ ,$$

we define $\sigma \ : S^* S_M \qquad S$ as follows, for every history $hs \quad S^* S_M$,

$$\sigma \ (hs) = s_a \text{ if } \sigma(s_0 \cdots s_n)(a) = 1 \ ,$$

We show that $\mathbb{P}^{\sigma'}_{s}(\Phi) = 1$. This is consequence of the fact that

$$t \quad S,\ p(s, \sigma(hs))(t) = \sigma \ (hs) \cdot p \ (s_{\sigma(hs)})(t) \ .$$

We show the converse implication, let $s \quad S_M$ be a state and $\sigma \ $ be a strategy for $\mathcal{M} \ $ such that

$$\mathbb{P}^{\sigma'}_{s}(\Phi) = 1 \ ,$$

Let $\sigma : S^* S : \quad A$ be the strategy obtained as follows, for every history $hs \quad (SA)^* S$ we have

$$\sigma(hs) = a \text{ if } \sigma \ (h \ s) = s_a \ ,$$

where $h \ $ is obtained as follows, if $h = s_0 a_0 \cdots s_n a_n$ then $h \ = s_0 s_{a_0} \ldots s_n s_{a_n}$. Similar argument as in the first part of the proof yields the result. □

In the sequel we assume that $S = S_M \quad S_R$ and for a state $s \quad S_M \quad S_R$ we denote sE the set of states q such that the couple (s, q) is an edge and Es the set of states q such that the couple (q, s) is an edge. The reward function is also transformed in such way that its labels the edges of the new transition graph. Hence we obtain the new reward function $r \ : E \quad \mathbb{R}^n$ such that $r \ (s, s \ ) = r(s)$.

For each edge we define a variable x_e and we denote p_e the transition probability of e. Consider the following system of equations:

$$s \quad S_M, \sum_{e \ sE} x_e = \sum_{e \ Es} x_e \ . \tag{4.20}$$

$$s \quad S_R, \quad e \quad sE,\ x_e = p_e \sum_{e' \ Es} x_{e'} \ . \tag{4.21}$$

$$i \quad 1, \cdots, k \ , \sum_{e \ E} x_e r_i(e) > 0 \ . \tag{4.22}$$

$$e \quad E, x_e \geq 0 \ . \tag{4.23}$$

$$\sum_{e \ E} x_e \geq 1 \ . \tag{4.24}$$

Theorem 4.18. *The system (4.20), (4.21), (4.22), (4.23) and (4.24) has a solution if and only if the positive region is nonempty.*

Proof. We first show that if there exists a solution that satisfies (4.20), (4.21), (4.22), (4.23) and (4.24) then there exists a positive strategy for Max and hence the positive strategy is not empty. Let $(n_1, \ldots, n_k)$ where $k = \ E \ $ a solution to the above system of equations. We define the stationary strategy σ defined as follows,

$$v \quad V, \quad e \quad vE,\ \mathbb{P}^{\sigma}_{v}(e) = \frac{x_e}{\sum_{e \ Ev} x_e} \ .$$

The strategy σ induces a Markov chain $\mathcal{M}[\sigma]$. According to (4.22) we know that there exists a closed class C such that:

$$c \quad C, \quad i \leq k, \ \mathbb{E}_c^{\sigma}\left[\sum_{j=0}^{n} r_i(S_j) \,\middle|\, T_c = n\right] > 0 \ ,$$

Using similar argument as in proof of Lemma 2.29 we obtain that

$$c \quad C, \quad i \leq k, \ \mathbb{P}_c^{\sigma}\left(\liminf_n \sum_{j=0}^{n} r_i(S_j) = \qquad\right) = 1 \ = \qquad \mathbb{P}_c^{\sigma}\left(\underline{\mathrm{Avg}}_{>0}^{k}\right) = 1 \ ,$$

and since

$$s \quad S, \ \mathbb{P}_s^{\sigma}(\ \ n \geq 0, \ S_n \quad C) > 0 \ ,$$

the implication follows.

We now show the converse implication. Assume that the system has no solution and let σ a strategy for Max. For every $e \quad E$ we define the quantities

$$\bar{F}_e = \limsup_n \mathbb{E}^{\sigma}\left[\frac{\sum_{i=0}^{n} \mathbb{1}_{e_i = e}}{n+1}\right] \ ,$$

and

$$\bar{x}_e = \frac{\bar{F}_e}{\sum_{i \quad E} \bar{F}_i} \ ,$$

We show that

$$i, \ \sum_{e \quad E} \bar{x}_e r_i(e) \leq 0 \ , \tag{4.25}$$

By definition $\bar{x}_e$ satisfies (4.20), (4.21), (4.23) and (4.24) and since by supposition the system does not have a solution, (4.25) follows.

We now show that σ is not almost-sure.

$$\begin{aligned}
\sum_{e\ \ E} \bar{x}_e r_i(e) \leq 0 \;=\; & \sum_{e\ \ E} \bar{F}_e r_i(e) \leq 0 \\
= \; & \sum_{e\ \ E} \limsup_n \mathbb{E}^\sigma \left[\frac{\sum_{j=0}^n \mathbb{1}_{e_j = e}}{n+1} \right] r_i(e) \leq 0 \\
= \; & \limsup_n \sum_{e\ \ E} \mathbb{E}^\sigma \left[\frac{\sum_{j=0}^n \mathbb{1}_{e_j = e}}{n+1} \right] r_i(e) \leq 0 \\
= \; & \limsup_n \mathbb{E}^\sigma \left[\sum_{e\ \ E} \frac{\sum_{j=0}^n \mathbb{1}_{e_j = e}}{n+1} r_i(e) \right] \leq 0 \\
= \; & \limsup_n \mathbb{E}^\sigma \left[\sum_{j=0}^n \frac{r_i(e_j)}{n+1} \right] \leq 0 \\
= \; & \liminf_n \mathbb{E}^\sigma \left[\sum_{j=0}^n \frac{r_i(e_j)}{n+1} \right] \leq 0 \\
= \; & \mathbb{E}^\sigma \left[\liminf_n \sum_{j=0}^n \frac{r_i(e_j)}{n+1} \right] \leq 0 \\
= \; & \mathbb{P}^\sigma \left(\liminf_n \sum_{j=0}^n \frac{r_i(e_j)}{n+1} \leq 0 \right) > 0 \\
= \; & \mathbb{P}^\sigma \left(\liminf_n \sum_{j=0}^n \frac{r_i(e_j)}{n+1} > 0 \right) < 1 \ ,
\end{aligned}$$

and hence if there is no solution then there is no almost-sure strategy thus no positive and hence the result. □

4.5.3 Comparison between objectives

In this part, we give an example where solving a conjunction of $\mathrm{Avg}_{>0}$ is possible but no strategy can ensure the conjunction of $\underline{\mathrm{Avg}}_{>0}$. Consider the Markov decision process depicted in Fig 4.7. The reward vector in state p is $(1, -1)$ and in state q is $(-1, 1)$.

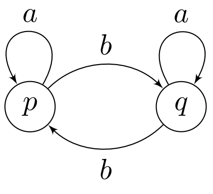


Figure 4.7: Markov decision process where Max can ensure conjunction of $\mathrm{Avg}_{>0}$ but no conjunction of $\underline{\mathrm{Avg}}_{>0}$

A strategy that achieves the objective $\mathrm{Avg}_{>0}$ would visit state p once then state q twice then again state p three times and so on, such strategy is winning with probability 1 since the accumulated average on two dimensions behaves as it is shown in Fig 4.8. On the other hand, no strategy can ensure $\underline{\mathrm{Avg}}_{>0}$ since one can easily verify that the equations given in the previous section cannot be satisfied.

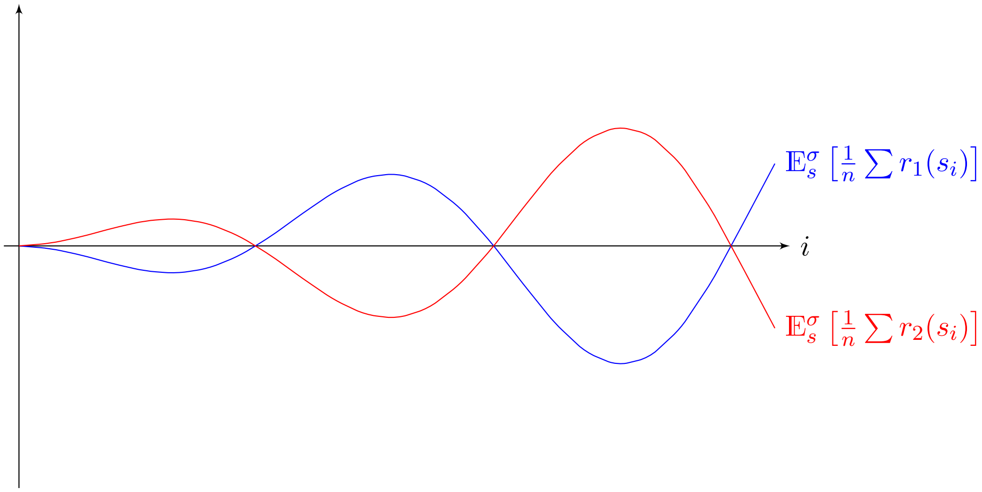


Figure 4.8: Behavior of the accumulated average when being consistent with σ.

4.5.4 Mixing lim inf and lim sup

In order to solve conjunctions of $\mathrm{Avg}_{>0}$ and $\underline{\mathrm{Avg}}_{>0}$ we prove that the value for the objective $\underline{\mathrm{Avg}}_{>0}$ $\mathrm{Avg}_{>0}$ are equal to the value of the objective $\underline{\mathrm{Avg}}_{>0} \quad \underline{\mathrm{Avg}}_{>0}$ played on the same Markov decision process.

Proposition 4.19 ([BBE10a]). *Let $\mathcal{M}$ be a Markov decision process with reward functions $r_i : S \quad \mathbb{R}$ for $i \quad 1, 2$ and let s be an almost-sure state for the objective $\underline{\mathrm{Avg}}_{>0} \quad \mathrm{Avg}_{>0}$ if and only if s is almost-sure for $\underline{\mathrm{Avg}}_{>0} \quad \underline{\mathrm{Avg}}_{>0}$.*

Proof. Let s be a state and assume that s is almost-sure for the objective $\underline{\mathrm{Avg}}_{>0} \quad \underline{\mathrm{Avg}}_{>0}$, then it is straightforward that s is almost-sure for the objective $\underline{\mathrm{Avg}}_{>0} \quad \mathrm{Avg}_{>0}$.

Let us proof the converse implication. Let τ be an almost-sure strategy for $\underline{\mathrm{Avg}}_{>0} \quad \mathrm{Avg}_{>0}$ from a state s. We turn the strategy τ into a finite memory strategy to ensure the same objective, then one can easily conclude that the new strategy achieves the objective $\underline{\mathrm{Avg}}_{>0} \quad \underline{\mathrm{Avg}}_{>0}$. Since τ is almost-sure, there exists $m > 0$ and a measurable set of runs A_m such that:

$$k \geq 0,\ A_m = \left\{ S^\omega \;\middle|\; \sum_{i=0}^{k} r_1(S_k) \geq -m \right\} ,$$

and

$$\mathbb{P}_s^\tau(A_m) \geq \frac{1}{2} .$$

Again since τ is almost-surely winning there exists a measurable set of runs B_n, $n > 0$, and $m \; > 0$ such that:

$$B_n = \left\{ S^\omega \;\middle|\; \left(\sum_{i=0}^{n} r_1(S_k) \geq 4m \right) \quad \left(\frac{1}{n+1} \sum_{k=0}^{n} r_2(S_k) \geq m \right) \right\} \;\; ,$$

and

$$\mathbb{P}_s^\tau(B_n \quad A_m) \geq \frac{1}{2} \;\; .$$

Let T_u be the stoping time associated with the state u defined as follow:

$$T_u = \min \left\{ 0 \leq k \leq n \;\middle|\; \left(4m \leq \sum_{i=0}^{k} r_1(S_i) \leq -m \right) \quad \left(\frac{1}{k+1} \sum_{i=0}^{k} r_2(S_i) \geq m \right) \right\} \;\; .$$

We define a new strategy σ as follows: from any almost-sure state u, σ simulates τ for T_u steps then restart simulating τ from the current state say v for T_v steps and so on. σ is clearly using only finite memory (counters of bounded size), let us show that σ is also almost-sure.

We show that σ ensures the objective $\underline{\mathrm{Avg}}_{>0}$. For each almost-surely winning state u, the expected accumulated reward at the stopping time T_u is $m - \frac{3}{4}m > 0$ hence in the long term, the accumulated reward on the first dimension diverge to   , thus the objective $\underline{\mathrm{Avg}}_{>0}$ is satisfied.

We show that σ satisfies the objective $\mathrm{Avg}_{>0}$. Let u be an almost-sure state, at the stopping time T_u we have

$$\mathbb{P}_u^\sigma \left(\frac{1}{k+1} \sum_{i=0}^{k} r_2(S_i) \geq m \;\middle|\; T_u = k \right) \geq \frac{1}{4} \;\; ,$$

According to Borel-Cantelli, the accumulated average reward goes above 0 infinitely often with probability 1, thus the objective $\mathrm{Avg}_{>0}$ is satisfied. □

As a consequence we obtain the following proposition:

Proposition 4.20. *The almost-sure region of* Max *for the objective* $\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^k_{>0}$ *is given by the largest sub Markov decision process* $\mathcal{M}[W]$ *where* $W \subseteq W_{=1}[\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^{(i)}_{>0}]$ *for* $0 \leq i \leq k$.

Proof. Let $\mathcal{M}[U]$ be a sub Markov decision process such that $U \subseteq W_{=1}[\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^{(i)}_{>0}]$, and let σ_i be an almost-sure strategy for the objective $\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^{(i)}_{>0}$ for $0 \leq i \leq k$.

We show that $\mathcal{M}[U]$ is almost-sure for Max for the objective $\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^k_{>0}$. Max alternates between strategies σ_i for $0 \leq i \leq k$ in a similar fashion as in the proof of Theorem 4.16.

We show that $W_{=1}[\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^k_{>0}]$ contains U. This follows from the fact that if a strategy is almost-sure for $\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^k_{>0}$ then it is almost-sure for any $\underline{\mathrm{Avg}}^k_{>0} \quad \mathrm{Avg}^{(i)}_{>0}$ played on the same Markov decision process □

Finally we obtain the following theorem:

Theorem 4.21. *Let* $\mathcal{M}$ *be a Markov decision process and let* $\Phi \subseteq S^\omega$ *be a winning condition that consists of boolean combination of positive-average conditions, the set* $W_{=1}[\Phi]$ *is computable in exponential time.*

Proof. The result follows from the fact that Φ can be rewritten as a new formula $\Phi \;$ such that Φ is in disjunctive normal form, the result then follows from Theorem 3.24 and the fact that we can solve any conjunctive formula of positive-average. □

4.6 Conclusion

In this chapter our main result is that the values of states for parity and positive-average objective for Markov decision processes are computable in polynomial times and that optimal strategies with finite memory exists. This result makes the synthesis of controller effective.

The other result is an algorithm for the computation of the almost-sure region for boolean combination of positive-averages combination.

The result obtained regarding the memory requirement and the computation time of the almost-sure region are presented in Table 4.1.

	Par $\mathrm{Avg}_{>0}$	Par $\underline{\mathrm{Avg}}_{>0}$	$\mathrm{Avg}^k_{>0}$	$\underline{\mathrm{Avg}}^k_{>0}$	$\underline{\mathrm{Avg}}^k_{>0}$ $\mathrm{Avg}^k_{>0}$	B.C.
A.S. region	Polynomial	Polynomial	Polynomial	Polynomial	Polynomial	Exponential
A.S. strategy	Pure	Pure	Pure	Stationary	Pure	Pure
Memory	Exponential	Exponential	Infinite	Memoryless	Infinite	Infinite

Table 4.1: Memory requirement for the different objectives; A.S. refers to almost-sure region and B.C. refers to boolean combination of positive-average objectives.

An interesting research direction is the boolean combination of parity and positive-average objectives. We seem to solve this problem in the restricted case of parity and positive-average objectives with lim sup semantics.

Another research direction is to solve parity and positive-average objectives in the setting of stochastic games. In the next chapter we give give an NP algorithm that solves parity and positive-average games with lim sup semantics.

Chapter 5

Two-player Par Avg$_{>0}$ Games

Contents

Abstract In this chapter, we generalize the construction of the previous chapter to stochastic games. We show that a slightly different construction for the almost-sure region allows us to compute value of two-player games with perfect information equipped with the Par Avg$_{>0}$ objectives. Moreover we show that even though the optimal strategies may require infinite memory, there exists an NP algorithm that computes the almost-sure region.

5.1 Introduction

Stochastic games with perfect information generalize Markov decision processes in the sense that the model is equipped with an second controller usually called Min whose objective is to minimize the probability that max satisfies her objective. In this model the two-player play in turn and the state space is partionned into Max s states and Min s states as opposed to concurrent games where the players choose there actions simultaneously.

These games are very useful in modeling problems and providing solutions for verification of open reactive systems even though they are less tractable than Markov decision processes. For instance computing the value of a reachability games is a problem that lies in NP $\setminus$ CoNP [Con92] as opposed to the polynomial time algorithm for Markov decision processes.

Our main goal in this chapter is to study stochastic games equipped with combination of parity and positive-average objectives. This objective were first studied in the case of non-stochastic games [CHJ05]. In the previous chapter we solved this problem for Markov decision processes, in the present chapter we show how to extend our result to the case of stochastic games.

Contribution and result In this chapter we give characterization of the almost-sure region for Max when the objective is Par Avg$_{>0}$, we also give an NP algorithm that computes this region together with an almost-sure strategy even though our almost-sure strategy may require infinite memory.

Outline of the chapter

- In Section 5.2 we introduce stochastic games and basic tools to study this model.
- In Section 5.3 we show that deciding whether Max has an almost-sure state for the objective Par Avg$_{>0}$ in a stochastic game is in NP.
- In Section 5.4 we give an algorithm that computes the value of each state for the objective Par Avg$_{>0}$ in a stochastic game.

5.2 Two-player Stochastic Games with Perfect Information

Two-player Stochastic Games with Perfect Information are similar to Markov decision processes except there are two kinds of states: states controlled by player Max whose goal is to maximize the probability that some objective is achieved, and states controlled by player Min who has the opposite goal and tries to minimize this probability.

Definition 5.1 (Stochastic game with perfect information)**.** *A stochastic game with perfect information is a tuple* $\mathcal{A} = (S, (S_1, S_2), A, p)$ *where:*

S *is a finite set of states,*

(S_1, S_2) *is a partition of* S*,*

A *is a set of actions,*

p *is a transition function.*

In the sequel we refer to two-player stochastic game with perfect information by stochastic game unless it is not clear by the context.

As opposed to Markov decision processes, the adversary can interfere in the play, and the notions of strategy and value have to be defined accordingly. First, the notion of strategy:

Definition 5.2 (Strategies)**.** *A strategy for* Max *is a function* $\sigma : (SA)^*S_1 \quad \Delta(A)$ *and a strategy for* Min *is a function* $\tau : (SA)^*S_2 \quad \Delta(A)$*.*

Once a couple of strategies chosen (σ, τ) and an initial state s fixed, we associate the probability measure $\mathbb{P}_s^{\sigma,\tau}$ over $s(AS)^\omega$ as the only measure over S^ω such that:

$$\begin{aligned}
&\mathbb{P}_s^{\sigma,\tau}(S_0 = s) = 1 \ , \\
&\mathbb{P}_s^{\sigma,\tau}(S_{n+1} = s \quad S_n = s_n \quad A_{n+1} = a_{n+1}) = p(s_n, a_{n+1})(s) \ , \\
&\mathbb{P}_s^{\sigma,\tau}(A_{n+1} = a \quad S_0 A_1 S_1 \cdots S_n = s_0 a_1 \cdots s_n) = \begin{cases} \sigma(s_0 a_1 \cdots s_n) \text{ if } s_n \quad S_1 \\ \tau(s_0 a_1 \cdots s_n) \text{ if } s_n \quad S_2 \end{cases}
\end{aligned}$$

Second, the notion of value of a state has to change as well. We define the value associated with a couple of strategies as follows:

Definition 5.3. *Let* s *be a state,* (σ, τ) *a couple of strategies, and* Φ *and objective. The value of* s *with respect to* (σ, τ) *for* Φ *is:*

$$\mathrm{Val}(s)_{\sigma,\tau} = \mathbb{P}_s^{\sigma,\tau}(\Phi) \ .$$

Also, since player Max and Min play in turns, it make sense to differentiate between two definition of the value of state. The first one is the so called *superior value*. Intuitively, this is the best possible value for a state when Min chooses his strategy first and Max decides the best possible answer. Formally,

Definition 5.4 (Superior value)**.** *Let s be a state and Φ be an objective, the superior value of s for Φ is:*

$$\overline{\mathrm{Val}}(s) = \inf_{\tau} \sup_{\sigma} \mathbb{P}_s^{\sigma,\tau}(\Phi) \ .$$

Dually, one defines also the so called *inferior value* of a state with the intuition that now Max chooses her strategy first and let Min defines the best possible answer.

Definition 5.5 (Inferior value)**.** *Let s be a state and Φ be an objective, the superior value of s for Φ is:*

$$\underline{\mathrm{Val}}(s) = \sup_{\sigma} \inf_{\tau} \mathbb{P}_s^{\sigma,\tau}(\Phi) \ .$$

The following equation always holds.

$$s \quad S, \ \underline{\mathrm{Val}}(s) \leq \overline{\mathrm{Val}}(s) \ . \tag{5.1}$$

Equation (5.1) follows the natural intuition; it is easier to win if one knows the strategy of his opponent. A legitimate question raises. When does these two quantities coincide? The answer follows from Martin s second determinacy theorem [Mar98] extended to stochastic games by Maitra and Sudderth [MS], which shows that for any Borel objective both values coincide.

Definition 5.6 (Determinacy)**.** *Let s be a state and Φ be a objective, then Φ is determined (for finite stochastic games with perfect information) if and only if for every stochastic game with perfect information and finitely many states and actions objective Φ:*

$$\underline{\mathrm{Val}}(s) = \overline{\mathrm{Val}}(s) \ .$$

In this case we denote the value of a state $\mathrm{Val}(s)$*.*

Theorem 5.7 (Borel Determinacy [Mar75, MS])**.** *Every Borel objective is determined for finite stochastic games with perfect information.*

This determinacy result shows that for Borel objectives, there always exist ε-optimal strategies for both players.

Definition 5.8 (ε-optimal strategies)**.** *Let $\varepsilon > 0$. A strategy $\sigma^\sharp$ for player 1 is ε-optimal if*

$$s \quad S, \quad \tau, \ \mathbb{P}_s^{\sigma^\sharp,\tau}(\Phi) \geq \mathrm{Val}(s) \ .$$

For player 2 the definition is symmetric. A 0-optimal strategy is simply called optimal.

While ε-strategies are guaranteed to exist in determined games, this is not the case for optimal strategies. However, provided the objective is tail, this existence is guaranteed:

Theorem 5.9 (Existence of optimal strategies [GH10])**.** *In every stochastic game with perfect information equipped with a tail Borel objective, both players have optimal strategies.*

Note that optimal strategies can be characterized as follows:

Definition 5.10 (Optimal strategies)**.** *Let $(\sigma^\sharp, \tau^\sharp)$ be a couple of strategies and Φ be an objective, $(\sigma^\sharp, \tau^\sharp)$ is an pair of optimal strategies if for every pair of strategies (σ, τ)*

$$s \quad S,\ \mathbb{P}_s^{\sigma,\tau^\sharp}(\Phi) \leq \mathbb{P}_s^{\sigma^\sharp,\tau^\sharp}(\Phi) \leq \mathbb{P}_s^{\sigma^\sharp,\tau}(\Phi)\ .$$

If this property holds in a game then the game is determined and:

$$s \quad S,\ \mathrm{Val}(s) = \mathbb{P}_s^{\sigma^\sharp,\tau^\sharp}(\Phi)\ .$$

In a similar fashion as the one for Markov decision processes, the notion of value is not the only interesting solution concept, we are also interested in qualitative solution concepts.

Definition 5.11 (Almost-sure and positive winning strategies)**.** *We say that* Max *wins almost-surely (resp. positively) from a state s if she has a strategy σ such that for every strategy τ $\mathbb{P}_s^{\sigma,\tau}(\Phi) = 1$ (resp. $\mathbb{P}_s^{\sigma,\tau}(\Phi) > 0$).*

We will use the following result about qualitative determinacy.

Theorem 5.12 (Qualitative determinacy [GH10])**.** *In any stochastic game equipped with a tail objective, each state is either almost-sure for* Max *or positive for* Max *and* Min *or almost-sure for* Min*.*

As a consequence,

Corollary 5.13. *In any stochastic game equipped with a tail objective, the following assertions hold.*

1. *If there exists an almost-sure strategy with memory M, then there exists an optimal strategy with same memory.*
2. *The states with value 1 are exactly the almost-sure states.*

Remark 5.14. *In the sequel, we say that a game $\mathcal{A}$ is almost-sure (resp positive), if every state in the game is almost-sure (resp positive).*

Finally, the notions of positive attractor and subgame will be basic tool notions to build our proofs upon.

Definition 5.15 (Positive attractor)**.** *Let $f : 2^S \quad 2^S$ be the operator such that for any $U \subseteq S$,*

$$f(U) = T \quad s \quad S_1 \quad a \quad A,\ p(s,a)(U) > 0 \qquad s \quad S_2 \quad a \quad A,\ p(s,a)(U) > 0 \quad .$$

Then $\overline{\mathrm{R}_{\mathrm{Max}}}(T, S)$ is the least fixed point of f.

We define also $\overline{\mathrm{R}_{\mathrm{Min}}}$ as the positive attractor for Min in a dual way.

Definition 5.16 (Subgame)**.** *Let $\mathcal{A}$ be a stochastic game with state space S. $\mathcal{A}[S\ ]$ is a subgame induced by $S\ \subseteq S$ if*

$$(\ s \quad S\ ),\ (\ a \quad A),\ p(s,a)(S\ ) = 1\ .$$

5.3 A Polynomial Certificate

Parity and Positive-average Stochastic Games In this section we study stochastic games where Max wants to maximize the probability to achieve the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$. Again we focus on the computation of the almost-sure region. We show that deciding whether Max wins almost-surely lies in NP and we give an algorithm to compute the value of each state. The challenging part is to provide a polynomial certificate even though the almost-sure strategies may require infinite memory, hence the usual trick of guessing a strategy for max and checking whether it is almost-sure or no will not work since there are infinitely many possible strategies.

Our goal is to provide a polynomial certificate for the almost-sure winning. We want to solve the following problem

Problem 5.17. *For a given stochastic game $\mathcal{A}$ with perfect information and a state s, decide whether s is almost-sure for* Max *for the* $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ *objective.*

Our approach consists in providing a polynomial size certificate for a subgame $\mathcal{A}[U]$ of $\mathcal{A}$ that contains s. This notion of certificate is defined by induction on the number of priorities in the arena, and the recursive definition depends on the parity of the highest priority in the subgame $\mathcal{A}[U]$.

A precise definition of the certificates is given in Definitions 5.22 and 5.23 for a start we give a first rough description a d-certificate (where d is the number of priorities) and why they are sufficient to prove that the subgame $\mathcal{A}[U]$ is almost-sure:

(a) If the highest priority d in the subgame $\mathcal{A}[U]$ is even, then denote S_d the set of vertices with priority d, a d-certificate is a decomposition of $\mathcal{A}[U]$ into $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)$ and $U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_f \setminus U, U)$, a $(d-1)$-certificate for the subgame $\mathcal{A}[U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)]$ and a positional strategy for Max in the subgame $\mathcal{A}[U]$ for the objective $\mathrm{Avg}_{>0}$. This is sufficient to conclude that $\mathcal{A}[U]$ is almost-sure because Max can play as follows. If the play is in $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)$, Max applies a strategies to attempt a visit a priority-d state, then switches to an almost-sure strategy for the positive-average objective. Then she either starts these two steps again or in case the play is in $\mathcal{A}[U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)]$ Max apply an almost-sure strategy in this subgame.

(b) If the highest priority d in the subgame $\mathcal{A}[U]$ is odd, then denote S_1 the set of vertices with priority 1, a d-certificate is given by a finite sequence $(R_i)_{0 \leq i \leq \ U \ -1}$ of disjoint subsets of $U \quad \overline{\mathrm{R}}_{\mathrm{Min}}(S_d \setminus U, U)$ such that i) for every i we have $R_i \subseteq S \quad \bigcup_j \overline{\mathrm{R}}_{\mathrm{Max}}(R_{j<i}, U)$, ii) a $(d-1)$-certificate for every R_i, and iii) the collection of sets $\overline{\mathrm{R}}_{\mathrm{Max}}(R_i, U)$ is a partition of U. The intuition beyond this certificate is that Max can apply a positive strategy induced by ii) to win the game $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ if the play starts from some $\overline{\mathrm{R}}_{\mathrm{Max}R_i}$, second we show using the qualitative determinacy (c.f. Theorem 5.12) that iii) implies that this strategy is actually almost-sure for the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$. set

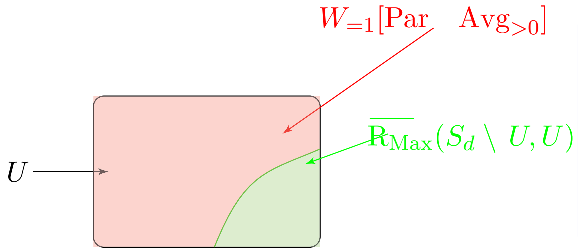
In order to provide a polynomial certificate, we proceed in three steps. First we characterize the set of almost-sure states (c.f. Propositions 5.19 and 5.21). Second we formally define what is a polynomial certificate (c.f. Definitions 5.22 and 5.23) and show that its is size polynomial in the number of states and priorities. Finally we show that the certificate can be checked in time polynomial in the number of states and priorities (c.f. (Lemma 5.25).

5.3.1 The Almost-sure Region

Lemma 5.18. *Let $\mathcal{A}$ be a stochastic game and $\mathcal{A}[U]$ be a subgame. Suppose that the highest priority d in $\mathcal{A}[U]$ is even and let S_d be the set of vertices with priority d. Then $\mathcal{A}[U]$ is almost-sure if and only if*

1. *$\mathcal{A}[U]$ is almost-sure for the positive-average objective.*

2. *$\mathcal{A}[U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)]$ is almost-sure for* Max


$W_{=1}[\mathrm{Par} \quad \mathrm{Avg}_{>0}]$
$\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)$
U

The idea of the above lemma is that if the subgame $\mathcal{A}[U]$ is almost-sure and if the highest priority d in $\mathcal{A}[U]$ is even then $\mathcal{A}[U]$ can be decomposed such that:

$$U = W_{=1}\left[\mathrm{Par} \quad \mathrm{Avg}_{>0}\right] \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U) \quad .$$

Figure 5.1: Decomposition of $\mathcal{A}[U]$ when the highest priority is even

Proof. Let $\mathcal{A}[U]$ be a subgame satisfying items 1. and 2. of Lemma 5.18. We show that $\mathcal{A}[U]$ is almost-sure for Max for the objective Par   Avg$_{>0}$. Let σ_{Sub}, $\sigma Attr$, and σ_{Avg} denote the almost-sure strategy in the subgame $\mathcal{A}[U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus U, U)]$, the attraction strategy to priority-d states in the subgame $\mathcal{A}[U]$, and the almost-sure strategy for the objective Avg$_{>0}$ in the subgame $\mathcal{A}[U]$ respectively. We define the application $Mode : (S \times A)^* \qquad Sub, Attr, Avg$ as follows:

$$Mode(s_0a_0\cdots s_na_n) = Sub \text{ if } \begin{cases} s_n \quad U \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_0 \setminus U, U)] \\ \Big[(Mode(s_0a_0\cdots s_{n-1}a_{n-1}) = Sub) \\ \Big(\frac{1}{n}\sum_{i=0}^{n-1} \geq \eta \quad Mode(s_0a_0\cdots s_{n-1}a_{n-1}) = Avg \Big) \Big] \end{cases} ,$$

$$Mode(s_0a_0\cdots s_na_n) = Attr \text{ if } \begin{cases} s_n \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_0 \setminus U, U)] \\ \Big[(n - \max \quad k \quad Mode(s_0a_0\cdots s_ka_k) = Attr \quad < [\ U\ ]) \\ \Big(\frac{1}{n}\sum_{i=0}^{n-1} \geq \eta \quad Mode(s_0a_0\cdots s_{n-1}a_{n-1}) = Avg \Big) \Big] \end{cases} ,$$

$$Mode(s_0a_0\cdots s_na_n) = Avg \text{ otherwise.}$$

We assume also that $Mode(\epsilon) = Avg$ where ϵ is the empty word.

The strategy σ that Max applies is as follows.

– For every $w \quad (S \times A)^*$ if the $Mode(w) = x$, then apply the strategy σ_x for x $Sub, Attr, Avg$.

We show that σ is almost-sure. Let $s \quad U$, then if

$$\tau,\ \mathbb{P}_s^{\sigma,\tau}(\quad N \geq 0, \quad n \geq N,\ Mode(S_0A_0\cdots S_nA_n) = Sub) = 1 \ .$$

Then Max plays consistent with strategy σ_{Sub}, and by definition of σ_{Sub} Max wins almost-surely.

If we have:

$$\tau,\ \mathbb{P}_s^{\sigma,\tau}(\quad n \geq 0,\ Mode(S_0A_0\cdots S_nA_n) = Attr) = 1\ .$$

First, we show that the parity objective is satisfied. Let A_n be the following sequence of events:

$$A_0 = \left\{S^\omega \quad \left(S_0 \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_0 \setminus U, U)\right) \quad (\ 0 \leq i \leq\ U\ ,\ \chi(S_i) = 0)\right\}\ ,$$

$$A_n = \left\{S^\omega \quad (\ i_0, \cdots, i_n),\ \left(\bigcap_{j=0}^{n} A_{i_j}\right) \quad \left(\ j \leq i_n,\ \left(S_j \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)\right) \quad (\chi(S_j) = 0)\right)\right\}\ .$$

Intuitively, a play of $\mathcal{M}$ belongs to A_n if it reaches the positive attractor to S_d n consecutive times and misses a state with priority-d. We show that that can happen only for finite number of time. Let m be the least transition probability of the $\mathcal{M}$, we have

$$\begin{aligned}\tau,\ (\ s \quad S),\ \mathbb{P}_s^{\sigma,\tau}(A_n) &\leq \left(1 - m^{\ U\ }\right)^{n+1} \\ &\leq \left(1 - m^{\ S\ }\right)^{n+1}\ .\end{aligned}$$

Since

$$\left(1 - m^{\ S\ }\right) < 1\ ,$$

we get

$$\tau,\ \sum_{n>0} \mathbb{P}_s^{\sigma,\tau}(A_n) < \qquad .$$

According to Borel-Cantelli Lemma we get:

$$\tau, \quad s \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U),\ \mathbb{P}_s^{\sigma,\tau}(\ \text{i.o. } A_k) = 0\ .$$

Hence a state with priority d is eventually visited, and the parity objective is satisfied with probability 1 when the play stays in $\overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)$.

Second, we prove that the positive-average objective is satisfied. By definition of σ_{Avg} there exists an integer $\eta > 0$ such that:

$$\tau, \quad s \quad W_{=1}[\mathrm{Avg}_{>0}]\ , \mathbb{P}_s^{\sigma_{Avg},\tau}\left(\quad n \geq 0,\ \frac{1}{n+1}\sum_{i=0}^{n} r(S_i) \geq \eta\right) = 1\ .$$

To show that σ satisfies the objective $\mathrm{Avg}_{>0}$ with probability 1, we still need to show that Max can make the average reward go above η, but this always possible since the play is happening in the almost-sure region for the positive-average condition it follows that

$$\tau, \quad s \quad W_{=1}[\mathrm{Avg}_{>0}],\ \mathbb{P}_s^{\sigma,\tau}\left(\quad n \geq 0,\ \frac{1}{n+1}\sum_{i=0}^{n} r(S_i) \geq \eta\right) = 1\ .$$

Thus the $\mathrm{Avg}_{>0}$ objective is achieved almost-surely. The above facts show that σ is almost-sure. This show that $\mathcal{A}[U]$ is almost-sure.

Let us show that any winning region satisfies items 1 and 2. Denote W the almost-sure region for Par $\mathrm{Avg}_{>0}$ objective played in $\mathcal{A}$. We prove that W satisfies items 1 and 2. That W satisfies item 1 follows from the fact that Max can win almost-surely Par $\mathrm{Avg}_{>0}$ in $\mathcal{A}[W]$. To see that item 2 holds, note that $\mathcal{A}[W \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_0 \setminus W, W)]$ is a trap for Max. So if she plays her almost-sure strategy σ defined on W, she wins almost-surely the Par $\mathrm{Avg}_{>0}$ objective. This terminates the proof. □

Proposition 5.19. *Let $\mathcal{A}$ be a stochastic game with a tail winning condition. Then the almost-sure region is given by the largest subset $W \subseteq S$ that induces a trap for* Min *and such that $\mathcal{A}[W]$ is almost-sure for* Max.

Proof. We show that the collection of subsets inducing a subgame and satisfying Lemma 5.18 is closed under union.

Let U_1 and U_2 be two subsets inducing subgames and satisfying Lemma 5.18, we show that $\mathcal{A}[U_1 \quad U_2]$ is almost-sure for Max for the objective Par   Avg$_{>0}$ i.e. we show that $\mathcal{A}[U_1 \quad U_2]$ satisfies Lemma 5.18.

First we show that if $\mathcal{A}[U_1]$ and $\mathcal{A}[U_2]$ are almost-sure for Avg$_{>0}$ then $\mathcal{A}[U_1 \quad U_2]$ is almost-sure for the objective Avg$_{>0}$ as well. Let σ_i be the almost-sure strategy for the objective Avg$_{>0}$ played in the subgame $\mathcal{A}[U_i]$ for $i \quad 1, 2$, then in the subgame $\mathcal{A}[U_1 \quad U_2]$ Max plays as follows:

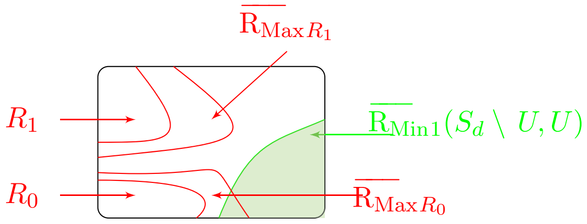
- If the play is in $\mathcal{A}[U_i]$, apply the strategy σ_i for $i \quad 1, 2$.

This strategy is clearly almost-sure since each $\mathcal{A}[U_i]$ is a trap for Min.

Second, since the condition is tail the almost-sure winning region W is a trap for Min and obviously $\mathcal{A}[W]$ is almost-sure. □

Lemma 5.20. *Let $\mathcal{A}$ be a stochastic game and $\mathcal{A}[U]$ a subgame. Suppose that the highest priority d in $\mathcal{A}[U]$ is odd, then $\mathcal{A}[U]$ is almost-sure if and only if there exists a sequence of disjoint subsets $(R_i)_{0 \leq i \leq \; U \; -1}$ such that*

1. *Every R_i is a trap for* Min *in $\mathcal{A}\left[U \quad \left(\overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus \; U, U) \quad \bigcup_{j=0}^{i} \overline{\mathrm{R}_{\mathrm{Max}}}(R_j, U)\right)\right]$,*
2. *every $\mathcal{A}[R_i]$ is almost-sure for the objective* Par   Avg$_{>0}$,
3. *$U = \bigcup_{i=0}^{\; U \; -1} \overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U)$,*


$\overline{\mathrm{R}_{\mathrm{Max}}}_{R_1}$
R_1
$\overline{\mathrm{R}_{\mathrm{Min1}}}(S_d \setminus \; U, U)$
R_0
$\overline{\mathrm{R}_{\mathrm{Max}}}_{R_0}$

The idea of the above lemma is that if the subgame $\mathcal{A}[U]$ is almost-sure and if the highest priority in $d \; \mathcal{A}[U]$ is odd then $\mathcal{A}[U]$ satisfies:

$$U = \bigcup_{i=0}^{\; U \; -1} \overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U) \; .$$

Figure 5.2: Decomposition of $\mathcal{A}[U]$ when the highest priority is odd

Proof. Let $\mathcal{A}[U]$ be a subgame induced by a subset $U \subseteq S$ and let $(R_i)_{0 \leq i \leq \; U \; -1}$ be a sequence of disjoint subsets of $S \quad \overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus \; U, U)$ such that 1,2 and 3 hold. We show that $\mathcal{A}[U]$ is almost-sure for the objective Par   Avg$_{>0}$. Max applies the following strategy σ. For any state s $\bigcup_{i=0}^{\; U \; -1} \overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U)$ we say that:

- s is *locked* if $s \quad \bigcup_{i=0}^{\; U \; -1} R_i$ and denote $ind(s)$ the least i such that $s \quad R_i$,

– s is *unlocked* if $s \quad \bigcup_{i=0}^{U\ -1} \overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U) \bigcup_{i=0}^{U\ -1} R_i$ and denote $ind(s)$ the least i such that $s \quad \overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U)$.

As long as the current state is unlocked, Max plays the attractor strategy to reach $R_{ind(s)}$ with positive probability. When the current state is locked, Max Max switches to her almost-sure strategy for the objective Par $\quad$ Avg$_{>0}$ in the subgame $\mathcal{A}[R_{ind(s)}]$ which exists according to condition 2. We show that using this strategy guarantees almost-surely that ultimately the current state S_n is locked forever and that $ind(S_n)$ remains ultimately constant. Precisely:

$$\mathbb{P}_s^\sigma(\quad 0 \le i \le \ U \ -1, \quad N \ge 0, \ \ n \ge N, \ S_n \quad R_i) = 1 \ . \tag{5.2}$$

Since the arena is finite, there exists $x > 0$ such that for every i playing the attractor strategy to R_i in $\overline{\mathrm{R}_{\mathrm{Max}}}(R_i, S)$ ensures to reach R_i in at most $\ U \ $ steps with probability at least x. As a consequence, according to condition 1, for every $0 \le m \le \ U \ -1$

$$\Big(\quad k, \ S_k \quad \overline{\mathrm{R}_{\mathrm{Max}}}(R_m, S)\Big) \ = \ \ (\quad k, \ S_k \quad R_{m-1} \qquad N \ge 0, \ \ n \ge N, \ S_n \quad R_m) \ \ , \tag{5.3}$$

$\mathbb{P}_s^\sigma$ almost-surely. Let M be the random variable with values in $\ \ 0 \ldots \ U \ -1 \ \ $ defined as follows:

$$M = \liminf_n ind(S_n) \ \ ,$$

then according to (5.3)

$$\mathbb{P}_s^\sigma(\quad N \ge 0, \ \ n \ge N, \ S_n \quad R_M) = 1 \ \ , \tag{5.4}$$

which shows (5.2) and terminates the proof of the direct implication.

Let us prove the converse implication, we proceed by induction on the size of $\ U\ $. First we show that if $\mathcal{A}[U]$ is almost-sure then the subgame $\mathcal{A}[U \quad \overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus U, U)]$ contains a non-empty set R such that $\mathcal{A}[R]$ is almost-sure for Max. Assume towards a contradiction the contrary, it follows that the arena $\mathcal{A}[U \quad \overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus U, U)]$ is almost-sure for Min which in turn shows that $\mathcal{A}[U]$ is almost-sure for Min since Min would have a strategy to either win in the subgame $\mathcal{A}[U \quad \overline{\mathrm{R}_{\mathrm{Min}}}(R_d \setminus U, U)]$ or visit a state with priority 1 infinitely often (using similar argument as in the proof of Lemma 5.18). Hence there exists a non-empty set R_0 in $U \quad \overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus U, U)$ such that R_0 is almost-sure for Max. If $S_1 = S$ we are over. Otherwise we can now use the same argument to build a subset $R_1 \subseteq U \quad \overline{\mathrm{R}_{\mathrm{Max}}}(R_0, U)$ such that $\mathcal{A}[R_1]$ is almost-sure for Max. Since at each step we obtain a subgame which contains at least one state less the result follows. □

Proposition 5.21. *Let $\mathcal{A}$ be a stochastic game such that the highest priority d is odd, then the almost-sure region is given by the largest trap satisfying Lemma 5.20.*

Proof. This is a direct corollary of Proposition 5.19 and Lemma 5.20. □

Now we are ready to give a formal definition of a certificate for the Problem 5.17.

5.3.2 Polynomial Size Certificate

Definition 5.22 (Even Certificate)**.** *Let $\mathcal{A}$ be a stochastic equipped with the objective* Par $\quad$ Avg$_{>0}$ *with d priorities such that the highest priority d is even, then a d-certificate for the almost-sure winning for a subgame $\mathcal{A}[U]$ is given by:*

A positional strategy σ for Max *in $\mathcal{A}[U]$,* *a* $(d-1)$*-certificate* C_{d-1} *for the almost-sure winning for the subgame* $\mathcal{A}[U \ \ \overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)]$.

Definition 5.23 (Odd Witness)**.** *Let* $\mathcal{A}$ *be a stochastic equipped with the objective* Par Avg$_{>0}$ *with d-priorities and such that the highest priority d is odd, then a d-certificate for the almost-sure winning for a subgame* $\mathcal{A}[U]$ *is given by:*

A sequence of disjoint subsets $(R_i)_{0\leq i\leq \ U \ -1} \subseteq S \ \ \overline{\mathrm{R}_{\mathrm{Min}}}(S_d \setminus U, U)$ *such that conditions 1 and 3 of Lemma 5.20 hold,*

a $(d-1)$*-certificate* C_{d-1} *for the almost-sure winning for the subgame* $\mathcal{A}[R_i]$ *for every* $0 \leq i \leq \ U \ -1$.

Lemma 5.24. *Let* $\mathcal{A}$ *a stochastic game and* $\mathcal{A}[U]$ *a subgame of* $\mathcal{A}$*. There exists a certificate of size* $O(nd)$ *where n is the size of U and d the number of priorities in U which shows that* $\mathcal{A}[U]$ *is almost-sure for* Max.

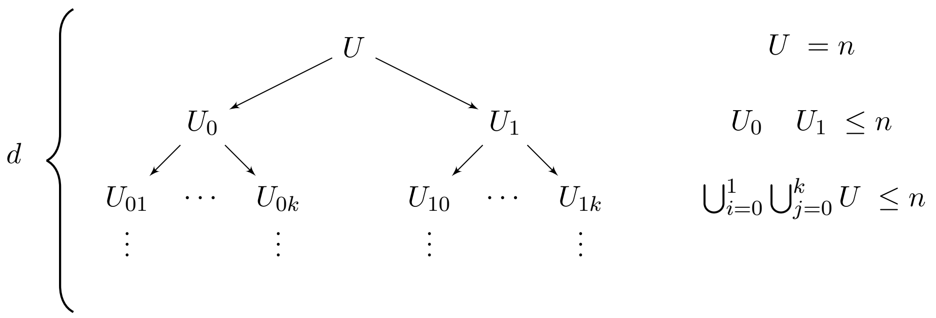


Figure 5.3: Inductive decomposition of the subgame $\mathcal{A}[U]$ according to Definitions 5.22 and 5.23.

Proof. Let $\mathcal{A}[U]$ be a subgame of a stochastic game $\mathcal{A}$. denote $C(n,d)$ the maximal size of a certificate for a subgame U with n vertices and d priorities. In each inductive step of the recursive definition of a certificate the size of U is reduced by at least one priority and one state. If the highest priority d is even then $\ \overline{\mathrm{R}_{\mathrm{Max}}}(S_d, S) \ \leq n$ thus,

$$C(n,d) \leq n + C(n-1, d-1) \ .$$

If the highest priority d is odd, the subsets R_i are disjoints hence $\sum_{i=0}^{\ U \ -1} \ R_i \ \leq n$ and

$$C(n,d) \leq n + \max_{\substack{n_1,\ldots,n_{\ U \ -1} \\ n_1+\cdots+n_{\ U \ -1} \leq n}} \sum_i C(n_i, d-1) \ ,$$

Since $C(n,1) \leq O(n)$, it follows that

$$C(n,d) \leq O(nd) \ .$$

□

5.3.3 Checking the Certificate in Polynomial Time

Lemma 5.25. *Let $\mathcal{A}$ be a stochastic game equipped with the objective* Par Avg$_{>0}$*, let $\mathcal{A}[U]$ be a subgame and let C be a certificate for $\mathcal{A}[U]$, then one can verify in time $O(dn^3)$ where d is the number of priorities of $\mathcal{A}$ and n the number of states in $\mathcal{A}$ that C is a valid certificate.*

Proof. Let C be a certificate for the almost-sure winning in the subgame $\mathcal{A}[U]$, first notice that if there is only one priority available in $\mathcal{A}[U]$, then either it is odd and $\mathcal{A}[U]$ surely losing or it is even and checking W amounts to checking if the strategy σ is almost-sure for the objective Avg$_{>0}$. For that consider the Markov decision process $\mathcal{A}[\sigma]$ induced by σ, in $\mathcal{A}[\sigma]$ then one can compute the value of every state for the mean payoff objective in $O(n^3)$ [Put94] and check these values are strictly positive. According to the proof of Theorem 3.32 this guarantees that σ is almost-sure for the Avg$_{>0}$ objective.

Assume by induction that the result holds for any subgame with less than d priorities and let $\mathcal{A}[U]$ be a subgame with d priorities.

If the highest priority d in $\mathcal{A}[U]$ is even then to check that C_d is a valid certificate, we perform the following steps:

- check that the positional strategy σ for Max is almost-sure for the objectives Avg$_{>0}$ in the subgame $\mathcal{A}[U]$.
- compute the set $\overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)$,
- check that C_{d-1} is a valid $(d-1)$-certificate for the subgame $\mathcal{A}[U \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)]$.

Let us show that these three steps can be performed in polynomial time. In order to verify that positional strategy σ is almost-sure in polynomial time consider the Markov decision process $\mathcal{A}[\sigma]$ induced by σ, in $\mathcal{A}[\sigma]$ one can compute the value of every state for the mean payoff objective in $O(n^3)$ [Put94], the computation of the set $U \quad \overline{\mathrm{R}_{\mathrm{Max}}}(S_d \setminus U, U)$ can be done in time $O(n^2)$ and verifying the certificate C_{d-1} can be done in polynomial time by induction hypothesis. Let $T(n,d)$ be the time complexity of the verification parametrized by n the number of states of U and d the number of priorities in U, thus:

$$T(n,d) \leq n^3 + T(n-1, d-1) \ . \tag{5.5}$$

If the highest priority d in $\mathcal{A}[U]$ is odd then we proceed as follows. For each $0 \leq i \leq \ U \ -1$,

- check that C_i is a valid $(d-1)$-certificate in the subgame $\mathcal{A}[R_i]$,
- compute the attractor $\overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U)$,
- remove from U the set $\overline{\mathrm{R}_{\mathrm{Max}}}(R_i, U)$,
- repeat with $i \quad i+1$.

Computing the attractor set can be done in time $O(n^2)$.

Let $T(n,d)$ be the complexity of the verification parametrized by n the number of states of U and d the number of priorities in U, then:

$$T(n,d) \leq n^3 + \max_{\substack{n_1, \dots, n_{\ U \ -1} \\ n_1 + \cdots + n_{\ U \ -1} \leq n}} \sum_{i=1}^{n} T(n_i, d-1) \ . \tag{5.6}$$

From Equations (5.5) and (5.6) and the concavity of $x \qquad x^3$ we obtain

$$T(n,d) = \mathcal{O}(dn^3) \ .$$

□

Theorem 5.26. *Given a stochastic game equipped with parity and positive-average objective, whether* Max *has an almost-sure winning strategy from a state s can be decided in* NP.

Proof. An NP algorithm that solves this problem starts first by guessing a subset U containing state s. It first checks whether U induces a subgame, then according to Lemma 5.25 one can check in polynomial time whether $\mathcal{A}[U]$ is almost-sure. Hence the result. □

5.4 Computing the Values

In this section we give a deterministic version of the the NP algorithm presented in Section 5.3 and show that computation of the almost-sure region can be done in time $O(nmd + n^d)$.

Algorithm 2

Input: Stochastic game $\mathcal{A}$ with state space S.

Output: Outputs the almost-sure winning region for Max for the objective Max.

```
Let $d$ be the highest priority of $\mathcal{A}$.
$S \qquad S$
if $d$ is even then
  repeat
    Let $R$ be the almost-sure winning region for Max in the subgame $\mathcal{A}[S\ ]$ for the objective Avg$_{>0}$.
    Compute $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$, the positive attractor of Max to priority-$d$ states in $R$
    Let $R$ be the almost-sure winning region for Max in the subgame $\mathcal{A}[R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)]$ for the objective Par  Avg$_{>0}$
    Compute $\overline{\mathrm{R}}_{\mathrm{Min}}(R \quad R\ , R)$, the positive attractor of Min to $R \quad R$ in the subgame $\mathcal{A}[R]$
    $S \qquad R \quad \overline{\mathrm{R}}_{\mathrm{Min}}(R \quad R\ , R)$
  until $R\ = R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$
  return $S$
else if $d$ is odd then
  $R$
  repeat
    Compute $\overline{\mathrm{R}}_{\mathrm{Min}}(S_d, S\ )$, the positive attractor of Min to priority-$d$ states in $\mathcal{A}[S\ ]$
    Let $R$ be the almost-sure region for Max in the subgame $\mathcal{A} \quad \overline{\mathrm{R}}_{\mathrm{Min}}(S_s, S\ )$ for the objective Par  Avg$_{>0}$
    Compute $\overline{\mathrm{R}}_{\mathrm{Max}}(R, S\ )$, the positive attractor of Max to $R$ in $\mathcal{A}[S\ ]$
    $R \qquad R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(R, S\ )$
    $S \qquad S \quad \overline{\mathrm{R}}_{\mathrm{Max}}(R, S\ )$
  until $R =$
  return $R$
```

The algorithm considers two cases: (a) when the highest priority d is even, and (b) when the highest priority d is odd. The details of the two cases are as follows:

- (a) If the highest priority d in the game is even, then we compute the almost-sure states of Max as the fixed point of the procedure where in each iteration removes from $\mathcal{A}$ some states that are positive for Min. The subgame $R \subseteq \mathcal{A}$ contains the almost-sure states for the objective $\mathrm{Avg}_{>0}$ (Line 5) which is a necessary condition to win according to Lemma 5.18. We decompose R into $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$ and $R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$. $R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_f \setminus R, R)$ has strictly less priorities than R. The states in $R \quad R$ are positive for Min in the original game since $R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$ is a trap for Max, we remove $\overline{\mathrm{R}}_{\mathrm{Min}R \quad R'}$. The correctness argument is similar to the proof of Lemma 5.18, namely that when $R \ = R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$, Max wins almost-surely by applying an almost-sure strategy in $R \quad \overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$, that exists by an inductive argument, and by alternating between the attraction strategy and the positive-average strategy in $\overline{\mathrm{R}}_{\mathrm{Max}}(S_d \setminus R, R)$ as shown in the proof of Lemma 5.18.

- (b) The second part of the algorithm is when the highest priority d in the game is odd, the set of almost-sure states is computed in rounds as the union of the almost-sure region for the objective Par $\quad \underline{\mathrm{Avg}}_{>0}$ in the subgame $\mathcal{A} \quad \overline{\mathrm{R}}_{\mathrm{Min}}(S_d \setminus R, R)$. The correctness argument follows from two facts: First, according to Lemmas 5.20, Max wins almost-surely in the subgame induced by the union of $\overline{\mathrm{R}}_{\mathrm{Max}R'}$. Second, since Max cannot win in $\mathcal{A} \quad \overline{\mathrm{R}}_{\mathrm{Min}}(S_d \setminus R, R)$ we are ensured that the computed set is the largest set of almost-sure winning states.

Theorem 5.27 (Algorithmic Complexity)**.** *In stochastic games, one can compute the almost-sure region for the objective* Par $\quad \underline{\mathrm{Avg}}_{>0}$ *in time* $O(nmd + n^d)$*, where* m *is the time one needs to solve positive-average objectives.*

Proof. This problem is solved by Alg 2, the correctness follows from the arguments above. Let $O(m)$ be the time complexity one needs to solve positive-average objectives. Denote $T(d)$ the complexity of Alg 2 parametrized by the number of priorities in the input game. The computation of the attractors in lines 6,8,15,17 is subsumed by the computation of the almost-sure region for the objective positive-average since solving theses games lie in NP $\setminus$ CoNP. In each recursive call the set of states reduces by at least one state and one priority and since there are at most n recursive calls we get

$$T(d) \leq n(m + T(d-1)) \ ,$$

It follows that

$$T(d) \leq nmd + n^d \ ,$$

hence the result. □

5.5 Conclusion

In this chapter we studied the problem of almost-sure winning for stochastic games equipped with the objective Par $\quad \mathrm{Avg}_{>0}$ and the main result we obtain is: despite the fact that almost-sure strategies may require infinite memory, there exists an NP algorithm that computes the almost-sure region and an almost-sure strategy. Unfortunately this procedure does work only for the lim sup semantics. Indeed the correctness proof for the almost-sure strategy described does not provide any lower bound on the accumulated average reward and hence the main argument used breaks in the case of $\mathrm{Par} \quad \underline{\mathrm{Avg}}_{>0}$. However we believe that the almost-sure region for the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ and $\mathrm{Par} \quad \underline{\mathrm{Avg}}_{>0}$ are equivalent, we finish this chapter by the following conjecture:

Conjecture 5.28. *Let $\mathcal{A}$ be a stochastic game and let s be a state, then:*

$$s \quad W_{=1}[\mathrm{Par} \quad \mathrm{Avg}_{>0}] \qquad s \quad W_{=1}[\mathrm{Par} \quad \underline{\mathrm{Avg}}_{>0}]\ .$$

Part III

Partial Information Setting

Chapter 6
Probabilistic Automata

Contents

Abstract In this chapter, we study yet another generalization of Markov decision process, namely *probabilistic automata*. Probabilistic automata were initially introduced by Rabin with the motivation to generalize the notion of deterministic automata. Probabilistic automata are also known in game theory as *one-player games in the dark*. These are games where the player cannot differentiate between states and thus cannot observe the current state of the play. We study two problems: the emptiness problem and the value one problem. The emptiness problem is a classical problem in automata theory: given a probabilistic automaton $\mathcal{A}$, decide whether there is a word accepted by $\mathcal{A}$. The value 1 problem is more familiar in game theory: given a probabilistic automaton, decide whether there exist words accepted with probability arbitrarily close to 1, in other words decide whether the corresponding one-player game in the dark has value 1. For the former problem we give a new proof of undecidability. For the latter problem, we show that it is undecidable then we introduced a sub class called ♯-acyclic probabilistic automata and show that the value 1 problem is PSPACE-complete for this class.

6.1 Introduction

Probabilistic automata. Rabin invented a very simple yet powerful model of probabilistic machine called *probabilistic automaton*, which, quoting Rabin, *are a generalization of finite deterministic automata* [Rab63]. A probabilistic automaton has a finite set of states Q and reads input words over a finite alphabet A. The computation starts from the initial state i and consists in reading the input word sequentially; the state is updated according to transition probabilities determined by the current state and the input letter. The probability to accept a finite input word is the probability to terminate the computation in one of the final states $F \subseteq Q$.

From a language-theoretic perspective, several algorithmic properties of probabilistic automata are known: while language emptiness is undecidable [MHC03b, GO10, Paz71], language equivalence is decidable [CMR07, Sch61, Tze92] as well as other properties [CL89, CMRR08].

Rather than formal language theory, our motivation for this work comes from control and game theory: we aim at solving algorithmic questions about partially observable Markov decision processes and stochastic games. For this reason, we consider probabilistic automata as Markov decision processes where the controller cannot observe the current state (or games in the dark), we also refer to the controller in such a system as a *blind controller*. The blind controller is in charge of choosing the next input letter to be executed by the system. Here stands a major difference between the model of our interest in the present chapter and the model considered in previously. Indeed, a strategy for a blind controllers is nothing but sequence of letters and the unique way for to decide the following action depends only on the number of letters already chosen. As opposed to fully observable Markov decision process or a simple stochastic game where the next action is chosen also accordingly to state visited through out the play. In other words, the strategy of a blind controller is an input word of the automaton. Another difference to note is the fact that we will concentrate on reachability objectives mainly, since this model have not been much investigated from an algorithmic point of view and also because we believe that reachability objectives are fundamental objectives to study.

The value of a probabilistic automaton. With this game-theoretic interpretation in mind, we define the *value* of a probabilistic automaton as the supremum acceptance probability over all input words, and we would like to compute this value. Unfortunately, as a consequence of Paz undecidability result, the value of an automaton is not computable in general. However, the value 1 problem was conjectured by Bertoni [Ber74] to be decidable[1], we study the decidability of this problem and obtain both positive and negative results:

Contribution and result The contribution of this chapter concerns two different algorithmic problem. Concerning the emptiness of probabilistic automata:

- we give a new simple proof inspired from Bertoni s construction [Ber74, BMT77],
- we show that the emptiness problem is already undecidable for automata with two probabilistic transitions.

Concerning the value 1 problem:

- we show that the value 1 problem is undecidable as opposed to what Bertoni conjectured,
- we show that the value 1 problem is already undecidable for automata with one probabilistic transition,
- we introduce the class of $\sharp$-acyclic automata and show that the value 1 problem is PSPACE complete for this class.

[1] Bertoni formulated the value 1 problem in a different yet equivalent way: “Is the cut-point 1 isolated or not?”.

Outline of the chapter

- In Section 6.2, we introduce the model of probabilistic automata.
- In Section 6.3, we study the emptiness of probabilistic automata we start by giving a new simple proof of the emptiness problem then we show that even in the very restricted case where probabilistic automata are restricted to two probabilistic transitions, deciding the emptiness remains undecidable. The key point of the proof is the result of Proposition 6.15
- In Section 6.4, we turn our attention to the value 1 problem with a rather game theoretic motivation. We solve an old standing open problem on the value 1 problem by answering negatively to it. The undecidability result follows from Proposition 6.24.
- In Section 6.5 we identify a class of probabilistic automata for which the value 1 problem is decidable, the so called ♯-*acyclic* automata. This last result is fairly technical and the decidability of this class of automata follows from Lemma 6.38.

6.2 Playing in the Dark

Definition 6.1 (Probabilistic automata)**.** *A probabilistic automaton is tuple* $\mathcal{A} = (Q, A, p_a \quad _A, q_0, F)$ *where:*

Q *is a finite set of states,*

A *is a finite set of actions,*

$a \quad A$, $p_a \quad [0, 1]^{Q \times Q}$ *is the transition matrix associated with the action* a*,*

q_0 *is an initial state,*

F *is a set of accepting states.*

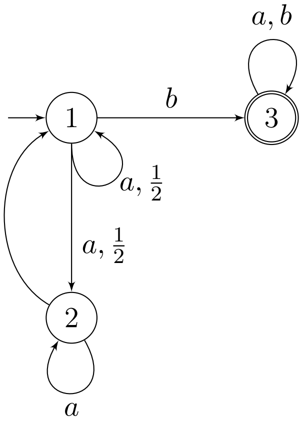


Figure 6.1: A probabilistic automaton.

Example 6.2. *Consider the automaton of Fig 6.1;*

$$\begin{aligned}
&Q = \quad 1,2,3 \quad ,\\
&A = \quad a,b \quad ,\\
&p_a = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; p_b = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\\
&q_0 = 1,\\
&F = \quad 3 \quad .
\end{aligned}$$

Let Q be a finite set of states, for a state $q \quad Q$, we denote by δ_q the Dirac distribution over q and for a subset $S \subseteq Q$ we denote by δ_S the uniform distribution over S. For a given distribution $\delta \quad \Delta(Q)$ and an action $a \quad A$

$$\delta \cdot a = \delta \cdot p_a \quad ,$$

and for word $w = w_0 \cdots w_n \quad A^*$,

$$\delta \cdot w = \delta \cdot p_{w_0} \cdots p_{w_n} \quad .$$

Definition 6.3 (Run of a probabilistic automaton). *Let $\mathcal{A}$ be probabilistic automata and $w \quad A^*$ be a finite word. A run of $\mathcal{A}$ on the input $w = w_0 \cdots w_n$ is the sequence of probability distributions $\delta_0, \ldots, \delta_n$ over S defined by:*

$$\begin{cases} \delta_0 = \delta_{s_0} \\ \delta_{i+1} = \delta_i \cdot w_i \end{cases}$$

Moreover, for every subset S of Q we denote

$$\mathbb{P}^w_{\mathcal{A}}(S) = \sum_{s \quad S} \delta_n(s) \quad .$$

Example 6.4. *Consider the example of Fig 6.1, the input word aab generates the following sequence:*

$$\begin{cases} \delta_0 = (1,0,0) \\ \delta_1 = \left(\frac{1}{2}, \frac{1}{2}, 0\right) \\ \delta_2 = \left(\frac{1}{4}, \frac{3}{4}, 0\right) \\ \delta_3 = \left(\frac{1}{4}, 0, \frac{3}{4}\right) \end{cases}$$

Definition 6.5 (Acceptance probability). *Let $\mathcal{A}$ be a probabilistic automaton and w be a word in A^*, the acceptance probability of w by $\mathcal{A}$ written $\mathbb{P}_A(w)$ is given by:*

$$\mathbb{P}_{\mathcal{A}}(w) = \mathbb{P}^w_{\mathcal{A}}(F) \quad .$$

In his seminal paper on probabilistic automata [Rab63], Rabbin defined the language accepted by a probabilistic automaton $\mathcal{A}$ as the set

$$\mathcal{L}(\mathcal{A}) = \quad w \quad A^* \quad \mathbb{P}_{\mathcal{A}}(w) \geq \lambda \quad ,$$

where $0 \leq \lambda \leq 1$ is called the cut-point. This definition raises a natural decision problem, the so called *Emptiness Problem.*

Problem 6.6 (Emptiness problem)**.** *Given a probabilistic automaton $\mathcal{A}$ and a rational $0 \leq \lambda \leq 1$, decide whether there exists a word $w \quad A^*$ such that $\mathbb{P}_{\mathcal{A}}(w) \geq \lambda$.*

In the case where λ is equal to 0 or 1 the Emptiness Problem turns out to be decidable. Actually deciding this problem when $\lambda = 0$ is always yes. When $\lambda = 1$, the problem reduces to the Universality Problem for non-deterministic automata over finite words which is PSPACE complete [Koz77]. We define also a strict version of the Emptiness Problem and refer to it as the *Strict Emptiness Problem* which is defined the same but $\mathbb{P}_{\mathcal{A}}(w) \geq \lambda$ is replaced by $\mathbb{P}_{\mathcal{A}}(w) > \lambda$. Deciding the latter version of the problem, when $\lambda = 0$ is nothing but deciding the Emptiness Problem for non-deterministic automata over finite words which is decidable in non-deterministic logarithmic space. When $\lambda = 1$ the answer again is trivial; always no.

In the case $0 < \lambda < 1$, the (Strict) Emptiness Problem is undecidable, the proof of this result is due to [Paz71]. Paz reduces the Emptiness Problem to some problem on context free grammars. Later, an alternative proof was given by Condon, Hanks and Madani [MHC03a]. They showed that the Emptiness Problem for two counters machines reduces to the Emptiness Problem. Paz was rather interested in expressiveness power of probabilistic automata, this could explain why his undecidability proof was spread on many sections of [Paz71] what makes it difficult to follow. Condon s et al. proof is more succinct but fairly technical. We propose an alternative proof, roughly speaking, the techniques used in our proof are inspired from the one used by Bertoni [Ber74]. The main idea of the reduction is that the emptiness problem is closely related to the equality problem (which is the variant where we ask for the set of words that are exactly accepted with probability λ) which reduces the PCP problem which is known to be undecidable.

6.3 Emptiness Problem for Probabilistic Automata

6.3.1 New Proof of Undecidability

In this section we show the undecidability of the (Strict) Emptiness Problem for the cut-point $\frac{1}{2}$ and for a restricted class of probabilistic automata, the so called *Simple Probabilistic Automata.*

Definition 6.7 (Simple Probabilistic Automata)**.** *A probabilistic automaton is called simple if every transition probability is in $\left\{0, \frac{1}{2}, 1\right\}$.*

Our proof is inspired from Bertoni s results on the so called *Equality Problem* [Ber74, BMT77].

Problem 6.8 (Equality Problem)**.** *Given a simple probabilistic automaton $\mathcal{A}$, decide whether there exists a word $w \quad A^*$ such that $\mathbb{P}_{\mathcal{A}}(w) = \frac{1}{2}$.*

Proposition 6.9 (Bertoni [Ber74])**.** *The equality problem is undecidable.*

The short and elegant proof of Bertoni is a reduction of the so called *Post Correspondence Problem* (PCP) which is known to be undecidable [Pap93] to the Equality Problem.

Problem 6.10 (PCP)**.** *Let $\varphi_1 : A \qquad 0,1\ ^*$ and $\varphi_2 : A \qquad 0,1\ ^*$ two functions, naturally extended to A^*. Is there a word $w \quad A^+$ such that $\varphi_1(w) = \varphi_2(w)$?*

Proof of Proposition 6.9. Given any instance $\varphi_1, \varphi_2 : A \qquad 0,1\ ^*$ of the PCP problem, we build an automaton $\mathcal{A}$ which accepts some word with probability $\frac{1}{2}$ if and only if PCP has a solution. Let $\psi : \ 0,1\ ^* \qquad [0,1]$ the injective mapping defined by:

$$\psi(a_0 \dots a_n) = \frac{a_n}{2} + \cdots + \frac{a_0}{2^n} \quad ,$$

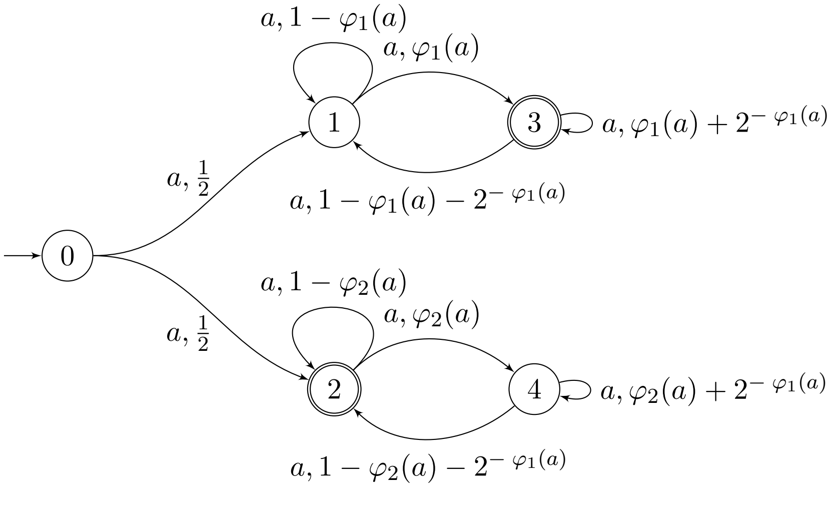

$a, 1-\varphi_1(a)$
$a, \varphi_1(a)$
1
3
$a, \varphi_1(a) + 2^{-\varphi_1(a)}$
$a, \frac{1}{2}$
$a, 1-\varphi_1(a) - 2^{-\varphi_1(a)}$
0
$a, 1-\varphi_2(a)$
$a, \varphi_2(a)$
$a, \frac{1}{2}$
2
4
$a, \varphi_2(a) + 2^{-\varphi_1(a)}$
$a, 1-\varphi_2(a) - 2^{-\varphi_1(a)}$

Figure 6.2: This automaton accepts a word with probability 1 if and only if there exists a solution to associated PCP instance.

and let $\theta_1 = \psi \circ \varphi_1$ and $\theta_2 = \psi \circ \varphi_2$. Let $\mathcal{A}_1 = (Q, A, M, q_0^1, q_F^1)$ the probabilistic automaton with two states $Q = \quad q_0^1, q_F^1 \quad$ and transitions:

$$a \quad A, M(a) = \begin{bmatrix} 1-\theta_1(a) & \theta_1(a) \\ 1-\theta_1(a)-2^{-\varphi_1(a)} & \theta_1(a)+2^{-\varphi_1(a)} \end{bmatrix} .$$

A simple computation shows that:

$$w \quad A^*, \ \mathbb{P}_{\mathcal{A}_1}(w) = \theta_1(w) \ . \tag{6.1}$$

A very similar construction produces a two-states automaton $\mathcal{A}_2$ such that:

$$w \quad A^*, \ \mathbb{P}_{\mathcal{A}_2}(w) = 1 - \theta_2(w) \ . \tag{6.2}$$

Let $\mathcal{A}$ be the disjoint union of these two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ plus a new initial state that leads with equal probability $\frac{1}{2}$ to one of the initial states q_0^1 and q_0^2 of $\mathcal{A}_1$ and $\mathcal{A}_2$. The automaton $\mathcal{A}$ is depicted in Fig 6.2. Then for every word $w \quad A^*$ and every letter $a \quad A$,

$$\begin{aligned} \left(w \quad A^*, \ \mathbb{P}_{\mathcal{A}}(aw) = \frac{1}{2} \right) \qquad & \left(w \quad A^*, \ \frac{1}{2}\mathbb{P}_{\mathcal{A}_1}(w) + \frac{1}{2}\mathbb{P}_{\mathcal{A}_2}(w) = \frac{1}{2} \right) \\ & (\ w \quad A^*, \ \theta_1(w) = \theta_2(w)) \\ & (\ w \quad A^*, \ \varphi_1(w) = \varphi_2(w)) \\ & \text{PCP has a solution,} \end{aligned}$$

where the first equivalence is by definition of $\mathcal{A}$, the second is by (6.1) and (6.2), the third holds because ψ is injective and the fourth is by definition of PCP. This completes the proof of Proposition 6.9. □

While the reduction of PCP to the Equality problem is relatively well-known, it may be less known that there exists a simple reduction of the Equality problem to the Emptiness and Strict Emptiness problems. The following proposition establishes a reduction from the Equality Problem to the Emptiness Problem and the Strict Emptiness Problem.

Proposition 6.11. *Given a simple probabilistic automaton $\mathcal{A}$, one can compute probabilistic automata $\mathcal{B}$ and $\mathcal{C}$ whose transition probabilities are multiple of $\frac{1}{4}$ and such that:*

$$\left(\quad w \quad A^+, \mathbb{P}_{\mathcal{A}}(w) = \frac{1}{2} \right) \qquad \left(\quad w \quad A^+, \mathbb{P}_{\mathcal{B}}(w) \geq \frac{1}{4} \right) \tag{6.3}$$

$$\left(\quad w \quad A^+, \mathbb{P}_{\mathcal{C}}(w) > \frac{1}{8} \right) \ . \tag{6.4}$$

Proof. The construction of $\mathcal{B}$ such that (6.3) holds is based on a very simple fact: a real number x is equal to $\frac{1}{2}$ if and only if $x(1-x) \geq \frac{1}{4}$. Consider the automaton $\mathcal{B}$ which is the cartesian product of $\mathcal{A}$ with a copy of $\mathcal{A}$ whose accepting states are the non accepting states of $\mathcal{A}$. Then for every word $w \quad A^*$, $\mathbb{P}_{\mathcal{A}_1}(w) = \mathbb{P}_{\mathcal{A}}(w)(1 - \mathbb{P}_{\mathcal{A}}(w))$, thus (6.3) holds.

The construction of $\mathcal{C}$ such that (6.4) holds is based on the following idea. Since $\mathcal{A}$ is simple, transition probabilities of $\mathcal{B}$ are multiples of $\frac{1}{4}$, thus for every word w of length $\ w\ $, $\mathbb{P}_{\mathcal{B}}(w)$ is a multiple of $\frac{1}{4^{|w|}}$. As a consequence, $\mathbb{P}_{\mathcal{B}}(w) \geq \frac{1}{4}$ if and only if $\mathbb{P}_{\mathcal{B}}(w) > \frac{1}{4} - \frac{1}{4^{|w|}}$. Adding three states to $\mathcal{B}$, one obtains easily a probabilistic automaton $\mathcal{C}$ such that for every non-empty word $w \quad A^*$ and letter $a \quad A$, $\mathbb{P}_{\mathcal{C}}(aw) = \frac{1}{2} \cdot \mathbb{P}_{\mathcal{B}}(w) + \frac{1}{2} \cdot \frac{1}{4^{|w|}}$, thus (6.4) holds. To build $\mathcal{C}$, simply add a new initial state that goes with equal probability $\frac{1}{2}$ either to the initial state of $\mathcal{B}$ or to a new accepting state q_f. From q_f, whatever letter is read, next state is q_f with probability $\frac{1}{4}$ and with probability $\frac{3}{4}$ it is a new non-accepting absorbing sink state q_*. □

Propositions 6.9 and 6.11 lead the following theorem

Theorem 6.12 (Paz [Paz71])**.** *The Emptiness and the Strict Emptiness Problems are undecidable for probabilistic automata. These problems are undecidable even for simple probabilistic automata and cut-point $\lambda = \frac{1}{2}$.*

Proof. According to Proposition 6.9 and Proposition 6.11 the Emptiness and the Strict Emptiness Problems are undecidable for cut-point $\frac{1}{2}$ and automata whose transition probabilities are multiples of $\frac{1}{8}$. The transformation of such automata into simple automata is easy. □

Earlier in this subsection, we have discussed some relations between non deterministic automata and probabilistic automata. The following corollary is another relation between these two automata-theoretic models.

Corollary 6.13. *The following problem is undecidable. Given a* non-deterministic *automaton on finite words, does there exists a word such that at least half of the computations on this word are accepting?*

6.3.2 Automata with Two Probabilistic Transitions

In this subsection we focus on a very special class of probabilistic automata, the one where the structure of the automaton contains only two probabilistic transitions. This is a very restricted class and the only interest of studying this class is to show that the emptiness problem remains undecidable even with such a strong restriction.

First let us define what is a probabilistic transition.

Definition 6.14. *A probabilistic transition is a couple $(q, a) \quad Q \times A$ such that there exists a state $t \quad S$ for which $0 < p_a(q, t) < 1$.*

We start first by studying the following problem.

Problem 6.15. *Given a simple probabilistic automaton $\mathcal{A}$ over an alphabet A with one probabilistic transition and given a rational language $L \subseteq A^*$, decide whether there exists $w \quad L$ such that $\mathbb{P}_{\mathcal{A}}(w) \geq \frac{1}{2}$.*

Proposition 6.16. *The Problem 6.15 is undecidable.*

Proof. We prove that the Problem 6.15 is undecidable by reducing the emptiness problem for probabilistic automata. We present a procedure that given a simple probabilistic automaton $\mathcal{A}$ over an alphabet A outputs a simple probabilistic automaton with one probabilistic transition $\mathcal{A}$ over an alphabet A together with a rational language $L \subseteq A^{\ *}$ such that

$$\left(\quad w \quad A^*,\ \mathbb{P}_{\mathcal{A}}(w) \geq \frac{1}{2} \right) \qquad \left(\quad w \quad L,\ \mathbb{P}_{\mathcal{A}'}(w) \geq \frac{1}{2} \right) \ . \tag{6.5}$$

Since we know by Theorem 6.12 that deciding the existence of a word w accepted with probability at least $\frac{1}{2}$ is not possible, it follows that Problem 6.15 is undecidable as well.

Roughy speaking $\mathcal{A}$ simulates $\mathcal{A}$; whenever a probabilistic transition is used in $\mathcal{A}$, it is simulated in $\mathcal{A}$ by the unique probabilistic transition of $\mathcal{A}$, denoted $(g, s) \quad Q \times A$. Whenever the automaton $\mathcal{A}$ reads the letter a, the automaton $\mathcal{A}$ faithfully simulates $\mathcal{A}$ by reading the following sequence of actions: $\widehat{a} = c(q_0, a) \cdot s \cdot t(q_0, a) \cdots c(q_{n-1}, a) \cdot s \cdot t(q_{n-1}, a) \cdot m$. The regular language $\mathcal{L}$ is used to check that $\mathcal{A}$ reads words of the form $\widehat{a}\widehat{b}\widehat{a}$, as otherwise the simulation of $\mathcal{A}$ can give arbitrary answers.

We now show how to construct the automaton $\mathcal{A}$, the alphabet A and the language L. Let $\mathcal{A}$ be a simple probabilistic automaton over an alphabet A and let Q be its set of states. The new set of states Q consist of all states Q plus a marked copy $\bar{Q} = \quad \bar{q} \quad q \quad Q$ plus three gadget states g, s_1, s_2

$$Q \ = Q \quad \bar{Q} \qquad g, s_1, s_2 \quad .$$

The new alphabet A is obtained as follows,

$$A \ = \quad s, m \qquad \bigcup_{q \quad Q, a \quad A} \quad c(q, a), t(q, a) \quad ,$$

where s stands for split, m stands for merge, $c(q, a)$ stands for check transition (a, q) and transition $t(a, q)$ stands for trigger transition (a, q). The semantics of this new action will become clearer after the transformation will be explicit.

We are now ready to start the simulation, we transform the action $a \quad A$ over a state $q \quad Q$ in the automaton $\mathcal{A}$ by the word

$$\hat{a} = c(q_0, a) \cdot s \cdot t(q_0, a) \cdots c(q_{n-1}, a) \cdot s \cdot t(q_{n-1}, a) \cdot m \ , \tag{6.6}$$

where $n = \quad Q$ and $\cdot$ denotes the concatenation operator. The transitions of $\mathcal{A}$ are as follows:

- For every letter $a \quad A$ and $q \quad Q$, the new letter $c(q, a)$ from state q leads *deterministically* to state g.
- The letter s from state g leads to state s_1 with probability $\frac{1}{2}$, and to state s_2 with probability $\frac{1}{2}$. Note that the latter action is the *only* probabilistic transition of $\mathcal{A}$.
- Any action $a = s$ from q leads with probability 1 to state i.

– The letter $t(q,a)$, sends the computation to states $\bar{r}$ and $\bar{s}$ where $p_a(q,r) = \frac{1}{2}$ and $p_a(q,s) = \frac{1}{2}$, otherwise if $p_a(q,r) = 1$ then the computation is sent to $\bar{r}$ from both s_1 and s_2.

– The letter m leads from state $\bar{q}$ leads to state q.

It is very important to notice that letters s has no effect on any state $q = g$ and that letters $c(q,a)$ and $t(q,a)$ have no effects on any state $q \quad = q$. Finally we define the language

$$L = \quad \widehat{w} \quad w \quad A^* > 0 \quad ,$$

where $\hat{w}$ is the natural extension of the transformation (6.6) over finite words. It is now straightforward that for any word $w \quad A^*$ we have

$$\mathbb{P}_{\mathcal{A}}(w) = \mathbb{P}_{\mathcal{A}'}(\bar{w}) \ .$$

Hence deciding if there exists w such that $\mathbb{P}_{\mathcal{A}}(w) \geq \frac{1}{2}$ is exactly the same as deciding whether there exists $w \quad L$ such that $\mathbb{P}_{\mathcal{A}'}(w \ ) \geq \frac{1}{2}$ and the result follows. □

The gadget used in the previous construction is depicted in Fig 6.4.

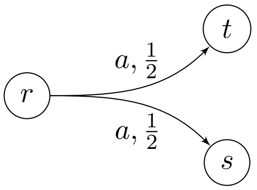


Figure 6.3: Probabilistic transition

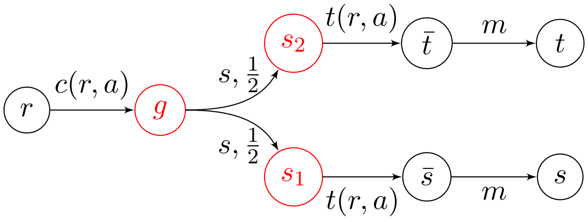


Figure 6.4: Gadget for a probabilistic transition

Theorem 6.17. *The emptiness problem for automata with two probabilistic transitions is undecidable.*

Proof. The undecidable problem described in Problem 6.15 reduces to the emptiness problem for simple probabilistic automata with two probabilistic transitions: given $\mathcal{A}$ and L, add a new initial state to $\mathcal{A}$ and from this new initial state, proceed with probability $\frac{1}{2}$ either to the original initial state of $\mathcal{A}$ or to the initial state of a deterministic automaton that checks whether the input word is in L. This new automaton accepts a word with probability more than $\frac{3}{4}$ if and only if the original automaton accepts a word with probability more than $\frac{1}{2}$. □

Once this result established, one can ask what about automata with one probabilistic transition? The intuition would suggest that it is easier to handle such model, but it turns out that the model is rather rich even with one probabilistic transition and decidability of the emptiness is not an obvious issue. For instance the value-one problem (Problem 6.22) is undecidable even for automata with one probabilistic transition (c.f. Proposition 6.25).

6.4 Value 1 Problem

In his seminal paper about probabilistic automata [Rab63], Rabin introduced the notion of *isolated cut-points.*

Definition 6.18. *A real number* $0 \leq \lambda \leq 1$ *is an isolated cut-point with respect to a probabilistic automaton* $\mathcal{A}$ *if:*

$$\varepsilon > 0, \quad w \quad A^*, \quad \mathbb{P}_{\mathcal{A}}(w) - \lambda \ \geq \varepsilon \ .$$

Rabin motivates the introduction of this notion by the following theorem:

Theorem 6.19 (Rabin [Rab63])**.** *Let* $\mathcal{A}$ *a probabilistic automaton and* $0 \leq \lambda \leq 1$ *a cut-point. If* λ *is isolated then the language* $\mathcal{L}_{\mathcal{A}}(\lambda) = \ \ u \quad A^* \ \ \mathbb{P}_{\mathcal{A}}(u) \geq \lambda$ *is rational.*

This result suggests the following decision problem.

Problem 6.20 (Isolation Problem)**.** *Given a probabilistic automaton* $\mathcal{A}$ *and a cut-point* $0 \leq \lambda \leq 1$*, decide whether* λ *is isolated with respect to* $\mathcal{A}$*.*

Bertoni [Ber74, BMT77] proved that the Isolation Problem is undecidable in general:

Theorem 6.21 (Bertoni [Ber74, BMT77])**.** *The Isolation Problem is undecidable.*

A closer look at the proof of Bertoni shows that the Isolation Problem is undecidable for a fixed λ, provided that $0 < \lambda < 1$.

However the same proof does not seem to be extendable to the case $\lambda \quad 0, 1$. This was pointed out by Bertoni in the conclusion of [BMT77]:

> Is the following problem solvable: $\delta > 0, \quad x, \ (p(x) > \delta)$? For automata with 1-symbol alphabet, there is a decision algorithm bound with the concept of transient state. We believe it might be extended but have no proof for it .

The open question mentioned by Bertoni is the Isolation Problem for $\lambda = 0$. Note that the case $\lambda = 1$ is essentially the same, since 0 is isolated in an automaton $\mathcal{A}$ if and only if 1 is isolated in the automaton obtained from $\mathcal{A}$ by turning final states to non-final states and vice-versa. When $\lambda = 1$, the Isolation Problem asks whether there exists some word accepted by the automaton with probability arbitrarily close to 1. We use the game-theoretic terminology and call this problem the Value 1 Problem.

Thus, the open question of Bertoni can be rephrased as the decidability of the following problem:

Problem 6.22 (Value 1 Problem)**.** *Given a probabilistic automaton* $\mathcal{A}$*, decide whether* $\mathcal{A}$ *has value* 1*.*

6.4.1 Undecidability of the Value 1 Problem

The following theorem solves the open problem left by Bertoni.

Theorem 6.23. *The Value* 1 *Problem is undecidable.*

The proof of Theorem 6.23 is inspired from the techniques used by Baier et al. in [BBG08] to prove that the emptiness problem for Büchi probabilistic automata is undecidable. The proof of Theorem 6.23 relies on the following proposition.

Proposition 6.24. *Let* $0 < x < 1$ *and* $\mathcal{A}_x$ *be the probabilistic automaton depicted on Fig. 6.5. Then* $\mathcal{A}_x$ *has value* 1 *if and only if* $x > \frac{1}{2}$.

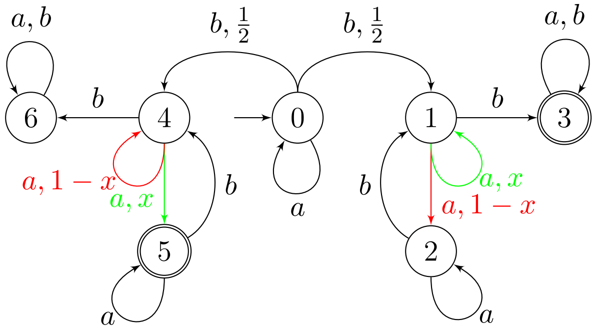


Figure 6.5: This automaton has value 1 if and only if $x > \frac{1}{2}$.

Proof. We shall prove:

$$\left(x > \frac{1}{2}\right) \qquad \left(\ \varepsilon > 0, \ w \quad A^*, \mathbb{P}_{\mathcal{A}_x}(w) \geq 1 - \varepsilon\right) \ . \tag{6.7}$$

In order to prove this equivalence we notice that: $\mathbb{P}_{\mathcal{A}_x}^{a^n b}(1 \quad 3) = x^n$ and $\mathbb{P}_{\mathcal{A}_x}^{a^n b}(4 \quad 6) = (1-x)^n$. Let $(n_k)_{k \quad \mathbb{N}}$ an increasing sequence of integers. By reading the word $w = a^{n_0}ba^{n_1}b \ldots a^{n_i}b$, we get:

$$\begin{cases} \mathbb{P}_{\mathcal{A}_x}^{w}(1 \quad 3) = 1 - \prod_{k \geq 0} \left(1 - x^{n_k}\right) \\ \mathbb{P}_{\mathcal{A}_x}^{w}(4 \quad 6) = (1-x)^{n_1} + (1-(1-x)^{n_1})(1-x)^{n_2} + \ldots \\ \qquad = 1 - \prod_{k \geq 0} (1-(1-x)^{n_k}) \leq \sum_{k \geq 0} (1-x)^{n_k} \end{cases}$$

If $x \leq \frac{1}{2}$ then $\mathbb{P}_{\mathcal{A}_x}^{w}(1 \quad 3) \leq \mathbb{P}_{\mathcal{A}_x}^{w}(4 \quad 6)$, hence maximizing the quantity will also maximize the quantity $\mathbb{P}_{\mathcal{A}_x}^{w}(4 \quad 6)$. Therefore no word w can be accepted with arbitrarily high probability if $x \leq \frac{1}{2}$ which proves the converse implication of (6.7).

Assume that $x > \frac{1}{2}$, we exhibit an increasing sequence of integers $(n_k)_{k \quad \mathbb{N}}$ such that for every $\varepsilon > 0$ we have:

$$\begin{cases} \sum_{k \geq 0} x^{n_k} = \\ \sum_{k \geq 0} (1-x)^{n_k} \leq \varepsilon \end{cases} \tag{6.8}$$

Let $C \quad \mathbb{R}$ and $n_k = \ln_x(\frac{1}{k}) + C$, notice that $\sum_{k\geq 0}(x)^{n_k} = x^C \cdot \sum_{k\geq 0}\frac{1}{k} = \quad$. On the other hand we have:

$$\begin{aligned}1 - x &= x^{\ln_x(1-x)}\\ &= x^{\frac{\ln(1-x)}{\ln x}}\end{aligned}$$

There exists $\beta > 1$ such that: $1 - x = x^{\beta}$, hence $\sum_{k\geq 0}(1-x)^{n_k} = \sum_{k\geq 0} x^{\beta n_k}$. So: $\sum_{k\geq 0} x^{\beta n_k} = x^{\beta C}\sum_{k\geq 0} x^{\beta \ln_x(\frac{1}{k})} = x^{\beta C}\sum_{k\geq 0}\frac{1}{k^{\beta}}$. Since this series converges, we satisfy (6.8) by choosing a suitable constant. Now because if (6.8) holds, it follows that $\mathbb{P}^{w}_{\mathcal{A}_x}(4 \quad 6) < \varepsilon$ and

$$\sum_{k\geq 0} x^{n_k} = \quad = \quad \prod_{k\geq 0}\left(1 - x^{n_k}\right) = 0 \ .$$

It is easy to see that a sequence of finite words $(a^{n_0}ba^{n_1}b\ldots a^{n_i}b)_{i \quad \mathbb{N}}$ is accepted with probability arbitrarily close to 1. □

Now we are ready to prove Theorem 6.23

Proof of Theorem 6.23. Given a probabilistic automaton $\mathcal{B}$ with alphabet A such that $a, b \quad \mathcal{B}$, we combine $\mathcal{B}$ and the automaton $\mathcal{A}_x$ on Fig.6.5 to obtain an automaton $\mathcal{C}$ which has value 1 if and only if there exists a word w such that $\mathbb{P}_{\mathcal{A}}(w) > \frac{1}{2}$. The input alphabet of $\mathcal{C}$ is $A \quad b \quad$ plus a new letter $\sharp$. $\mathcal{C}$ is computed as follows. First, the transitions in $\mathcal{A}_x$ on letter a are deleted. Second, we make two copies $\mathcal{A}_4$ and $\mathcal{A}_1$ of the automaton $\mathcal{B}$, such that the initial state of $\mathcal{A}_4$ is 4 and the initial state of $\mathcal{A}_1$ is 1. From states of $\mathcal{A}_4$ and $\mathcal{A}_1$ other than the initial states, reading letter b leads to the sink state 6. Third, from a state s of $\mathcal{A}_4$ the transition on the new letter $\sharp$ is deterministic and leads to 5 if s is a final state and to 4 if s is not a final state. Fourth, from a state s of $\mathcal{A}_1$ the transition on the new letter $\sharp$ is deterministic and leads to 1 if s is a final state and to 2 if s is not a final state. Fifth, the final states of $\mathcal{C}$ are 5 and 3. Sixth, states $0, 3, 6, 5$ and 2 are absorbing for letters in A.

Then suppose there exists w such that $\mathbb{P}_{\mathcal{A}}(w) > \frac{1}{2}$ and let us show that $\mathcal{C}$ has value 1. Let $\epsilon > 0$ and let $u_\epsilon = ba^{i_0}ba^{i_1}ba^{i_2}b\cdots a^{i_k}$ be a word accepted by $\mathcal{B}$ with probability $1 - \epsilon$. Then by construction of $\mathcal{C}$,

$$\mathbb{P}_{\mathcal{C}}(b(w\sharp)^{i_0}b(w\sharp)^{i_1}b(w\sharp)^{i_2}b\cdots(w\sharp)^{i_k}) \geq \mathbb{P}_{\mathcal{A}}(u_\epsilon) \geq 1 - \epsilon,$$

thus $\mathcal{C}$ has value 1.

Now suppose that for every $w \quad A^*, \mathbb{P}_{\mathcal{A}}w \leq \frac{1}{2}$ and let us show that $\mathcal{C}$ has not value 1. Let $w \quad (A \quad b, \sharp \ )^*$. Factorize w in $w = u_0v_0\sharp u_1v_1\sharp u_kv_k\cdots$ such that $u_i \quad b^*$ and $v_i \quad A^*$. Then by construction of $\mathcal{C}$ and by hypothesis, $\mathbb{P}_{\mathcal{C}}(w\ ) \leq \mathbb{P}_{\mathcal{A}_{\frac{1}{2}}}(u_0au_1au_2a\cdots u_ka) \leq \operatorname{Val}\mathcal{A}_{\frac{1}{2}}$. Thus $\operatorname{Val}\mathcal{C} \leq \operatorname{Val}\mathcal{A}_{\frac{1}{2}}$ and according to Proposition 6.24, $\operatorname{Val}\mathcal{C} < 1$. □

6.4.2 Automata with one Probabilistic Transition

The following proposition was obtained in joint work we did together with Nathanaël Fijalkow and appeared in [FGO11], we give an improved construction of the one published in the technical report.

Proposition 6.25. *Let $\mathcal{A}$ be a simple probabilistic automaton, then there exists a computable probabilistic automaton $\mathcal{B}$ which contains one probabilistic transition and*

$$\operatorname{Val}_{\mathcal{A}} = 1 \qquad \operatorname{Val}_{\mathcal{B}} = 1 \ . \tag{6.9}$$

The idea used in the proof is very similar to the one used in Theorem 6.17. Recall in that theorem, we constructed an automaton $\mathcal{A}$ with 1 probabilistic transition and a regular language L, then we used a second probabilistic transition to run in parallel the computation in $\mathcal{A}$ and $\mathcal{A}_L$ the automaton that recognizes L. In the proof of Proposition 6.25, we will run the computation first in an automaton which is a slight modification of $\mathcal{A}$ and then plug into $\mathcal{A}_L$. Roughly speaking, this construction works because we are interested in the value of the automaton and not exact acceptance probability.

Proof of Proposition 6.25. Let $\mathcal{A}$ a probabilistic automaton. We construct an automaton $\mathcal{B}$ that simulates a computation of $\mathcal{A}$ using only probabilistic transition. The automaton $\mathcal{B}$ is obtained by composing two automata; $\mathcal{A}$ and $\mathcal{A}_L$.

First we construct an automaton $\mathcal{A}$ as follows: the state space is Q and the set of actions A such that the new set of states Q consists of all states Q plus a marked copy $\bar{Q} = \quad \bar{q} \quad q \quad Q$ plus three gadget states g, s_1, s_2 and state i.

$$Q = Q \quad \bar{Q} \quad g, s_1, s_2, i \quad .$$

The new alphabet A is obtained as follows,

$$A = \ s, m, f \quad \bigcup_{q \quad Q, a \quad A} c(q,a), t(q,a) \quad ,$$

where, as in the proof of Proposition 6.15, s stands for split, m stands for merge, $(c(a,q))$ stands for check transition (a,q), action f stands for finish, state i stands for idle, and transition $t(a,q)$ stands for trigger transition (a,q).

The automaton $\mathcal{A}_L$ is the finite deterministic automaton with initial state q_L and set of accepting states F_L. that recognizes the following language:

$$L = \ \hat{w} \quad w \quad A^* \quad ,$$

where $\hat{w}$ is the natural extension over finite words of the transformation

$$a \quad A,\ \hat{a} = c(q_0,a) \cdot s \cdot t(q_0,a) \cdots c(q_{n-1},a) \cdot s \cdot t(q_{n-1},a) \cdot m \ ,$$

where $n = \ Q$ and $\cdot$ denotes the concatenation operator. Note that from the definition of L we have that $q_L \quad F_L$.

The automaton $\mathcal{B}$ consists of the composition of $\mathcal{A}$, $\mathcal{A}_L$, and a sink   such that the initial state of $\mathcal{B}$ is q_0 and the accepting states of $\mathcal{B}$ are the one of $\mathcal{A}_L$. The transition of $\mathcal{B}$ are as follows:

- For every letter $a \quad A$ and $q \quad Q$, the new letter $(c(a,q))$ from state q leads *deterministically* to state g.

- The letter s from state g leads to state s_1 with probability $\frac{1}{3}$, to state s_2 with probability $\frac{1}{3}$, and with probability $\frac{1}{3}$ to state i. Note that the latter action is the *only* probabilistic transition of $\mathcal{A}$.

- Any action $a = s$ from q leads with probability 1 to state i.

- The letter $t(q,a)$, sends the computation to states $\bar{r}$ and $\bar{s}$ where $p_a(q,r) = \frac{1}{2}$ and $p_a(q,s) = \frac{1}{2}$, otherwise if $p_a(q,r) = 1$ then the computation is sent to $\bar{r}$ from both s_1 and s_2.

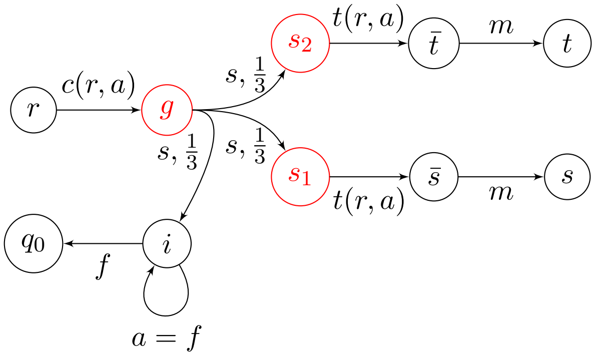


Figure 6.6: New gadget for a probabilistic transition

- The action m leads from state $\bar{q}$ leads to state q.
- From any state $q \quad F$, the action f leads with probability 1 to q_L.
- From state i, the action f leads with probability 1 to q_0.
- From any state in F_L, the action f leads to q_L.
- From any state $q \quad\ i, F, F_L$, the action f leads to .

It is very important to notice that the action s has no effect on any state $q = g$ and that actions $c(q,a)$ and $t(q,a)$ have no effects on any state $q \ = q$. Finally, whenever an action does not fall in one of the previous cases, it has no effect on the computation. The gadget used to simulate the computation is depicted in Fig 6.6.

Let us show the direct implication of (6.9). Let $w \quad A^*$ be a word such that $\ w \ = k$, we get that

$$\mathbb{P}_{\mathcal{B}}(\widehat{w} \cdot f) = \left(\frac{2}{3}\right)^{n+k} \mathbb{P}_{\mathcal{A}}(w) \ ,$$

and

$$\mathbb{P}_{\mathcal{B}}^{\widehat{w}\cdot f}(q_0 \qquad q_0) = 1 - \left(\frac{2}{3}\right)^{n+k} \ .$$

Denote $x = \left(\frac{2}{3}\right)^{n+k}$ we get

$$\begin{aligned} m \quad \mathbb{N}^*, \ \mathbb{P}_{\mathcal{B}}((\widehat{w} \cdot f)^m) &= \left(1 - \mathbb{P}_{\mathcal{B}}^{(\widehat{w}\cdot f)^m}(q_0 \qquad q_0)\right) \mathbb{P}_{\mathcal{A}}(w) \\ &= (1 - (1-x)^m)\, \mathbb{P}_{\mathcal{A}}(w) \ . \end{aligned}$$

Hence we get that

$$\sup_{m \quad \mathbb{N}} \mathbb{P}_{\mathcal{B}}((\widehat{w} \cdot f)^m) = \mathbb{P}_{\mathcal{A}}(w) \ .$$

Consequently, the direct implication of (6.9) follows.

Let us show the converse implication of (6.9). Assume that $\mathrm{Val}_{\mathcal{B}} = 1$. Let $\varepsilon > 0$ and w a word such that

$$\mathbb{P}_{\mathcal{B}}(w) \geq 1 - \varepsilon \ . \qquad (6.10)$$

By construction of $\mathcal{B}$ we can write

$$w = u_0 \cdot f \cdot u_1 \cdots f \cdot u_k \cdot f \ \ ,$$

where u_i does not contain the letter f for $0 \leq i \leq k$. Let $(x_i)_{0\leq i\leq k}$, $(y_i)_{0\leq i\leq k}$, and $(z_i)_{0\leq i\leq k}$ be the sequences that respectively give $\delta_{q_0}(u_0 f \cdots u_i f)(q_0)$, $\delta_{q_0}(u_0 f \cdots u_i f)(\quad)$, and $\delta_{q_0}(u_0 f \cdots u_i f)(q_L)$. We also denote $\check{L} = \quad u \quad L \quad v \quad A^*$, $\widehat{v} = u$ and if $u \quad \check{L}$ we denote $\check{u} \quad A^*$ the word such that $\widehat{\check{u}} = u$. We show that:

$$\quad 0 \leq i \leq k,\ u_i \quad \check{L} \text{ and } \mathbb{P}_{\mathcal{A}}(\check{u}_i) \geq 1 - \varepsilon \ \ . \tag{6.11}$$

Let

$$j = \min_{0\leq i<k} \left\{ z_i < \frac{2}{3} \quad z_{i+1} \geq \frac{2}{3} \right\} \ \ . \tag{6.12}$$

Let us show that j is always defined. Assume that $j = \quad$ it follows that $z_k < \frac{2}{3}$ which contradicts the fact that $\mathbb{P}_{\mathcal{B}}(w) > 1 - \varepsilon$.

By definition of $\mathcal{B}$, we have that

$$\begin{aligned} z_{j+1} &\leq z_j + \left(\frac{2}{3}\right)^{\ u_j\ } \mathbb{P}_{\mathcal{A}}(\check{u}_j) x_j \\ &\leq z_j + \frac{2}{3}(1 - z_j) \ \ . \end{aligned}$$

Using (6.12) we get

$$\frac{2}{3} \leq z_{j+1} \leq \frac{8}{9} \ \ . \tag{6.13}$$

Let

$$M = \max_{\substack{j\leq i\leq k \\ u_i \quad L}} \mathbb{P}_{\mathcal{A}}(\check{u}_i) \ \ .$$

Then we can write

$$\quad j \leq l \leq k,\ z_{l+1} \leq z_l + x_l \left(\frac{2}{3}\right)^{\ u_l\ } M \ \ . \tag{6.14}$$

$$\quad j \leq l \leq k,\ y_{l+1} \geq y_l + x_l \left(\frac{2}{3}\right)^{\ u_l\ } (1 - M) \ \ . \tag{6.15}$$

Denote equation (6.14) A_l and equation (6.15) B_l for every $l \geq j$. It follows that

$$\left(\sum_{l=j}^{k} A_l\right) \equiv \left(z_k \leq z_j + M \sum_{l=j}^{k} x_l \left(\frac{2}{3}\right)^{\ u_l\ } \right) \ \ . \tag{6.16}$$

$$\left(\sum_{l=j}^{k} B_l\right) \equiv \left(y_k \geq y_j + (1 - M) \sum_{l=j}^{k} x_l \left(\frac{2}{3}\right)^{\ u_l\ } \right) \ \ . \tag{6.17}$$

Using (6.10) and by definition of $(z_i)_{0\leq i\leq k}$ and definition of $(y_i)_{0\leq i\leq k}$ we get that

$$1 - \varepsilon \leq z_j + M \sum_{l=j}^{k} x_l \left(\frac{2}{3}\right)^{\ u_l\ } \ \ .$$

$$\varepsilon \geq y_j + (1 - M) \sum_{l=j}^{k} x_l \left(\frac{2}{3}\right)^{\ u_l\ } \ \ .$$

Since $y_i \geq 0$ for $0 \leq i \leq k$ we have

$$\frac{\varepsilon}{1-M} \geq \sum_{l=j}^{k} x_l \left(\frac{2}{3}\right)^{u_l} \quad . \tag{6.18}$$

Then (6.13) and (6.18) give

$$1-\varepsilon \leq \frac{8}{9} + M\frac{\varepsilon}{1-M} \quad .$$

Or

$$M \geq 1 - \frac{\varepsilon}{1-\frac{8}{9}} \quad ,$$

and (6.11) follows which terminates the proof of the proposition. □

The following theorem is a straightforward corollary of Proposition 6.25

Theorem 6.26. *The value 1 problem is undecidable for automata with one probabilistic transition.*

6.5 ♯-acyclic Probabilistic Automata

In this section, we introduce a new class of probabilistic automata, the so called ♯-*acyclic probabilistic automata*, for which the value 1 problem is decidable.

At first glance, the Value 1 Problem may seem quite similar to decision problems about omega-regular languages. For example, if the input alphabet has only one letter then the automaton is a Markov chain and transient states will be ultimately left almost-surely, which can be encoded by fairness constraints. However, Theorem 6.23 suggests that the Value 1 Problem cannot be solved using known decision procedures about finite-state automata.

The value 1 problem for a simple probabilistic automata can be rephrased as a "quantitative" decision problem about non-deterministic automaton on finite words: does there exists words such that among all computation paths, the proportion of non-accepting computation paths is arbitrarily small?

6.5.1 Subset construction for ♯-acyclic automata

To get a decision algorithm for the value 1 problem, our starting point is the usual subset construction for non-deterministic automata, however the quantitative aspect of the above problem requires the subset construction to be customized. Precisely, we use not only the usual action of a letter a on a subset $S \subseteq Q$ of states but consider also another action $a^\sharp$ with intuition that this operation simulates the effect of reading the action a an arbitrarily large number of time. Roughly speaking, each action a induces a Markov chain, $a^\sharp$ deletes states that are transient in the Markov chain induced by a.

Definition 6.27 (Actions of letters and ♯-reachability)**.** *Let $\mathcal{A}$ a probabilistic automaton with alphabet A and set of states Q. Given $S \subseteq Q$ and $a \quad A$, we denote:*

$$S \cdot a = \quad t \quad Q \quad s \quad S, M_a(s,t) > 0 \quad .$$

A state $t \quad Q$ is a-reachable from $s \quad Q$ if for some $n \quad \mathbb{N}$, $\mathbb{P}_{\mathcal{A}}(a^n(s,t)) > 0$. A state $s \quad Q$ is a-recurrent if for any state $t \quad Q$,

$$(t \textit{ is } a\textit{-reachable from } s) \;=\; (s \textit{ is } a\textit{-reachable from } t) \quad .$$

A set $S \subseteq Q$ is a-stable if $S = S \cdot a$. If S is a-stable, we denote:

$$S \cdot a^\sharp = \quad s \quad S \quad s \text{ is a-recurrent} \quad .$$

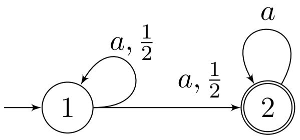


Figure 6.7: A probabilistic automaton with one action

Definition 6.28. *The* support graph $\mathcal{G}_\mathcal{A}$ *of a probabilistic automaton $\mathcal{A}$ with alphabet A and set of states Q is the directed graph whose vertices are the non-empty subsets of Q and whose edges are the pairs (S, T) such that for some letter $a \quad A$, either $(S \cdot a = T)$ or $(S \cdot a = S$ and $S \cdot a^\sharp = T)$.*

Example 6.29. *In the automaton of Fig 6.7 (which is essentially a Markov chain), the action of a on the support $\quad 1, 2 \quad$ is stable:*

$$\quad 1, 2 \quad \cdot a = \quad 1, 2 \quad ,$$

but the iteration of the action a is:

$$\quad 1, 2 \quad \cdot a^\sharp = \quad 2 \quad ,$$

since state 2 is the only recurrent state in the Markov chain induced by $(\quad 1, 2 \quad , a)$. The full support graph is depicted in Fig 6.8.

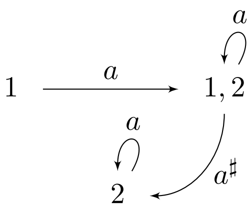


Figure 6.8: The support graph of the automaton depicted in Fig 6.7

Reachability in the support graph of $\mathcal{A}$ is called ♯-reachability in $\mathcal{A}$. Note that if $T \ \subseteq T$ and T is ♯-reachable from S then so is T. The class of ♯-acyclic probabilistic automata is defined as follows.

Definition 6.30 (♯-acyclic probabilistic automata)**.** *A probabilistic automaton is* ♯-acyclic *if the only cycles in its support graph are self-loops.*

Obviously, this acyclicity condition is quite strong. However, it does not forbid the existence of cycles in the transition table, see for example the automaton depicted on Fig. 6.9. Note also that the class of ♯-acyclic automata enjoys good properties, for example it is closed under cartesian and parallel product.

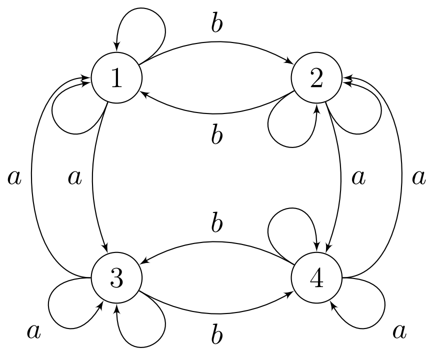


Figure 6.9: This automaton is ♯-acyclic .

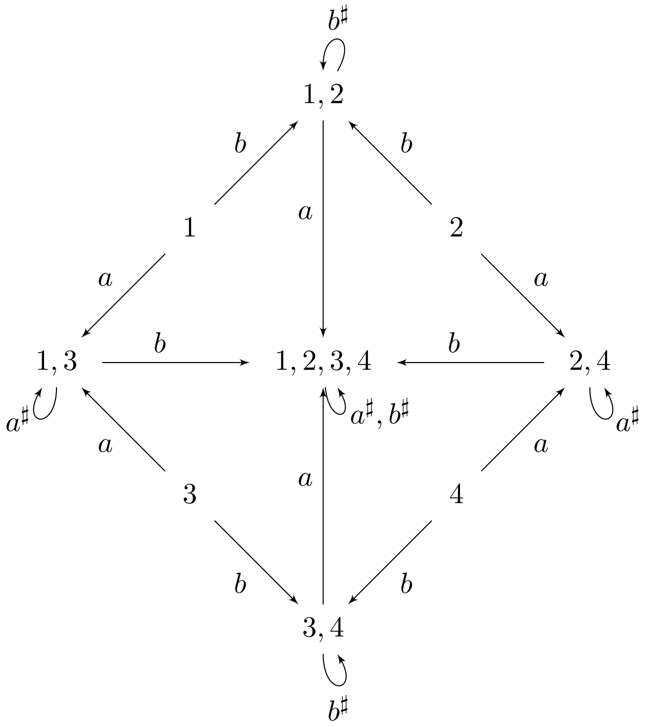


Figure 6.10: The support graph of the automaton depicted in Fig 6.9.

6.5.2 Decidability of ♯-acyclic automata

Theorem 6.31. *Let $\mathcal{A}$ be a probabilistic automaton with initial state q_0 and final states F. Suppose that $\mathcal{A}$ is ♯-acyclic . Then $\mathcal{A}$ has value 1 if and only if F is ♯-reachable from q_0 in $\mathcal{A}$.*

The proof of Theorem 6.31 relies on the notion of limit-paths.

Definition 6.32 (Limit paths and limit-reachability)**.** *Let $\mathcal{A}$ be a probabilistic automaton with states Q and alphabet A. Given two subsets S, T of Q, we say that T is limit-reachable from S in $\mathcal{A}$ if* *there exists a sequence $w_0, w_1, w_2, \ldots \in A^*$ of finite words such that for every state $s \in S$:*

$$\mathbb{P}_{\mathcal{A}}(w_n(s,T)) \xrightarrow[n]{} 1 .$$

The sequence $w_0, w_1, w_2, \ldots$ is called a limit path *from S to T, and T is said to be limit-reachable from S in $\mathcal{A}$.*

Note that if $T' \subseteq T$, then whenever T' is limit-reachable from S, then so is T. In particular $\mathcal{A}$ has value 1 if and only if F is limit reachable from $\{q_0\}$.

Theorem 6.31 essentially states that for ♯-acyclic automata, ♯-reachability and limit-reachability coincide. In the general case, may the probabilistic automaton be ♯-acyclic or not, ♯-reachability implies limit-reachability.

Proposition 6.33. *Let $\mathcal{A}$ be a probabilistic automaton with states Q and $S, T \subseteq Q$. If T is ♯-reachable from S in $\mathcal{A}$ then T is limit-reachable from S in $\mathcal{A}$.*

Proof. Proposition 6.33 is a consequence of the two following facts.

First, if there is an edge from S to T in the support graph of $\mathcal{A}$, then T is limit reachable from S: let $S, T \subseteq Q$ and $a \in A$. If $S \cdot a = T$, then the sequence constant equal to a is a limit path from S to T. If $S \cdot a = S$ and $S \cdot a^\sharp = T$ then by definition of $S \cdot a^\sharp$, $(a^n)_{n \in \mathbb{N}}$ is a limit path from S to T.

Second, limit-reachability is a transitive relation: let $S_0, S_1, S_2 \subseteq Q$ such that S_1 is limit-reachable from S_0 and S_2 is limit-reachable from S_1. Let $(u_n)_{n \in \mathbb{N}}$ a limit-path from S_0 to S_1 and $(v_n)_{n \in \mathbb{N}}$ a limit-path from S_1 to S_2. Then $(u_n v_n)_{n \in \mathbb{N}}$ is a limit-path from S_0 to S_2. □

The converse implication of Theorem 6.31 is not true in general. For example, consider the automaton depicted on Fig. 6.11. There is only one final state; state 3. The initial state is not represented, it leads with equal probability to states 1, 2 and 3. The transitions from states 1, 2 and 3 are either deterministic or have probability $\frac{1}{2}$. It turns out that this automaton has value 1, because $((b^n a)^n)_{n \in \mathbb{N}}$ is a limit-path from $\{1,2,3\}$ to $\{3\}$. However, $\{3\}$ is *not* reachable from $\{1,2,3\}$ in the support graph, as can be seen on Fig. 6.12. Thus, limit-reachability does not imply ♯-reachability in general. This automaton is *not* ♯-acyclic , because his support graph contains the following cycle: $\{1,2,3\}$ is b-stable and $\{1,2,3\} \cdot b^\sharp = \{1,3\}$ while $\{1,3\} \cdot a = \{1,2,3\}$.

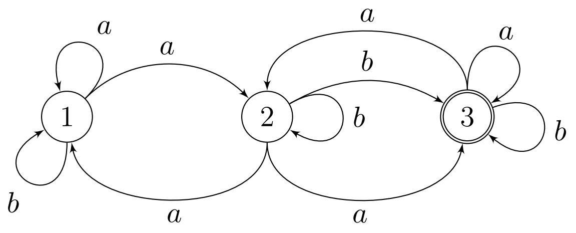


Figure 6.11: This automaton has value 1 and is not ♯-acyclic .

Now we shall prove that for ♯-acyclic automata, limit-reachability implies ♯-reachability. We use the following notions.

Definition 6.34 (Stability and ♯-stability)**.** *Let $\mathcal{A}$ be a probabilistic automaton with state space Q. Then $\mathcal{A}$ is* stable *if for every letter $a \in A$, Q is a-stable and $\mathcal{A}$ is* ♯-stable *if it is stable and for every letter $a \in A$ $Q \cdot a^\sharp = Q$.*

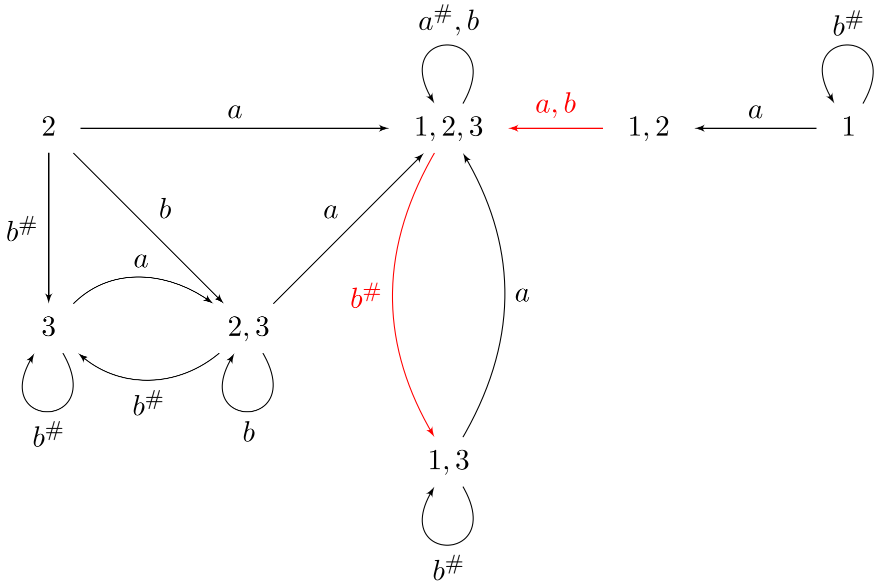
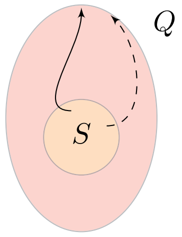


Figure 6.12: The support graph of the automaton depicted in Fig 6.11.

The main idea of the proof is to show that whenever there are two supports that are limit-reachable, one can construct a path in the support graph between these two supports. We do this by induction on the depth of the support graph. To handle the basic cases of the induction we use the following lemmata.

Lemma 6.35 (Blowing lemma)**.** *Let $\mathcal{A}$ be a $\sharp$-acyclic probabilistic automaton with state space Q and $S \subseteq Q$. Suppose that $\mathcal{A}$ is $\sharp$-acyclic and $\sharp$-stable and that Q is limit-reachable from S in $\mathcal{A}$. Then Q is $\sharp$-reachable from S in $\mathcal{A}$.*



The largest circle depicts the state space of some $\sharp$-acyclic automaton denoted Q and the smallest one a subset of state S. The left-hand side arrow represents a limit path from S to Q, then according to the blowing lemma if the automaton is $\sharp$-stable, there exists a path from S to Q in the support graph, which is represented by the dashed arrow.

Figure 6.13: Blowing lemma in image

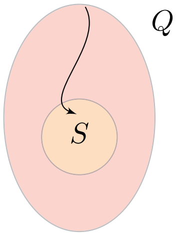
Proof. If $S = Q$ there is nothing to prove. If $S = Q$, we prove that there exists $S_1 \subseteq Q$ such that $S \subsetneq S_1$ and S_1 is $\sharp$-reachable from S. Since $S \subsetneq Q$ and since there exists a limit-path from S to Q there exists at least one letter a such that $S \cdot a \subseteq S$. Since $\mathcal{A}$ is $\sharp$-acyclic , there exists n $\mathbb{N}$ such that $S \cdot a^{n+1} = S \cdot a^n$ i.e. $S \cdot a^n$ is a-stable. Let $S_1 = (S \cdot a^n) \cdot a^\sharp$. To prove that $S \subsetneq S_1$, we prove that S_1 contains both S and $S \cdot a$. Let s S. By definition, every state t of $S \cdot a^n$ is a-accessible from s. Since $\mathcal{A}$ is $\sharp$-stable, state s is a-recurrent and by definition of a-recurrence, s is a-accessible from t. Since $S \cdot a^n$ is a-stable, s $S \cdot a^n$ and since s is a-recurrent s $(S \cdot a^n) \cdot a^\sharp = S_1$. The proof that $S \cdot a \subseteq S_1$ is similar.

If $S_1 = Q$ the proof is complete.

If $S_1 \subsetneq Q$ we proceed by induction and build an increasing sequence $S \subsetneq S_1 \subsetneq S_2 \subsetneq \ldots \subsetneq S_n = Q$ such that for every $1 \leq k < n$, S_{k+1} is limit-reachable from S_k. Since limit-rechability is transitive (see proof of Proposition 6.33), this completes the proof of the blowing lemma. □

The following lemma states a crucial property to establish the decidability of ♯-acyclic automata; once a computation on a input word has reached the entire state space there is no possibility to shrink it back.

Lemma 6.36 (Flooding lemma)**.** *Let $\mathcal{A}$ be a probabilistic automaton with states Q. Suppose that $\mathcal{A}$ is ♯-acyclic and ♯-stable. Then Q is the only set of states limit-reachable from Q in $\mathcal{A}$.*



We keep the same convention; The largest circle depicts the state space of some ♯-acyclic automaton denoted Q and the smallest one a subset of state S. Then according to the claim of the flooding lemma if the automaton ♯-stable, the full arrow that represents a limit-path from Q to S cannot exists i.e. if the support of the initial distribution is Q, then this support remains unchanged whatever sequence of words is being read.

Figure 6.14: Flooding lemma in image

Even though the flooding property seems to be natural, it does not hold true in general. For instance:

Example 6.37. *In Fig 6.15 is depicted a probabilistic automaton, we don't specify initial nor accepting states. Notice that the automaton is not ♯-acyclic since $\ 1 \ \cdot a = \ 3 \ $ and $\ 3 \ \cdot a = \ 1 \ $. It is clear that $\ 1, 2, 3, 4 \ $ is ♯-stable Nevertheless, the sequence $(ab)^n$ is a limit-path from $\ 1, 2, 3, 4 \ $ to $\ 1, 4 \ $ as $(\delta_Q \cdot (ab)^n)(\ 1, 4 \ ) = 1 - \frac{1}{2^n}$.*

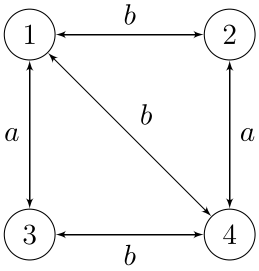


Figure 6.15: A probabilistic automaton for which the Flooding lemma does not hold.

Proof of Lemma 6.36. Assume toward a contradiction that $(u_n)_n \ \mathbb{N}$ is a limit path from Q to some subset of states $T \subsetneq Q$. We prove that $T = Q$.

Let $A_T = \ a \ \ A \ \ T \cdot a = T \ $. First, we prove that for every letter $a \ \ A_T$, $Q \ \ T$ is a-stable. Otherwise there would be $a \ \ A_T$ and $t \ \ T$ which is a-reachable from some state $s \ \ Q \ \ T$, since $\mathcal{A}$ is ♯-stable, s and t are both a-recurrent, and by definition of a-recurrence, since t is a-reachable from s, s would be a-reachable from t as well. But $s \quad Q \quad T$ and $t \quad T$, which contradicts the a-stability of T for every $a \quad A_T$.

Second, we prove that $u_n \quad A_T^*$ for only finitely many $n \quad \mathbb{N}$. Since for every $a \quad A_T$, $Q \quad T$ is a-stable, then during the computation $\delta_Q = \delta_0, \delta_1, \ldots, \delta_{\ u_n\ }$ on the word u_n, $\sum_{s \ Q \ T} \delta_k(s)$ is constant. Thus, for every $n \quad \mathbb{N}$,

$$\begin{aligned} \mathbb{P}^{u_n}_{\mathcal{A}}(s \qquad Q) &= \sum_{s\ Q\ T} (\delta_Q \cdot u_n)(s) \\ &= \sum_{s\ Q\ T} \delta_Q(s) \\ &= \frac{\ Q\ -\ T\ }{\ Q\ } > 0\ , \end{aligned}$$

where the inequality follows from the fact that $T \subsetneq Q$. Since $(u_n)_{n \ \mathbb{N}}$ is a limit-path from Q to T, we know that $\mathbb{P}_{\mathcal{A}}(u_n(s, Q))$ converges to 0 hence the inequality can hold only for finitely many $n \quad \mathbb{N}$.

Now we show that there exists $T_1 \subseteq Q$ such that:

(i) $T_1 = T$,

(ii) T is $\sharp$-reachable from T_1 in $\mathcal{A}$,

(iii) and T_1 is limit-reachable from Q in $\mathcal{A}$.

Since any infinite subsequence of a limit-path is a limit-path, and since we proved that $u_n \quad A_T^*$ for only finitely many $n \quad \mathbb{N}$, we can assume w.l.o.g. that for every $n \quad \mathbb{N}$, $u_n \quad A_T^*$. Thus for every $n \quad \mathbb{N}$, there exists $v_n \quad A^*$, $a_n \quad A \quad A_T$ and $w_n \quad A_T^*$ such that $u_n = v_n a_n w_n$. W.l.o.g. again, since A is finite and δ_Q is compact, we can assume that $(a_n)_{n \ \mathbb{N}}$ is constant equal to a letter $a \quad A \quad A_T$ and that $(\delta_Q \cdot v_n)_{n \ \mathbb{N}}$ converges to a probability distribution $\delta \quad \delta_Q$.

The choice of T_1 depends on $\mathrm{Supp}(\delta) \cdot a$.

If $\mathrm{Supp}(\delta) \cdot a = T$ then we choose $T_1 = \mathrm{Supp}(\delta)$. Then (i) holds because $a \quad A_T$, (ii) holds because $T = T_1 \cdot a$ and (iii) holds because $(v_n)_{n \ \mathbb{N}}$ is a limit-path from Q to T_1.

If $\mathrm{Supp}(\delta) \cdot a = T$ then we choose $T_1 = \mathrm{Supp}(\delta) \cdot a$. Then (i) clearly holds and (iii) holds because $(v_n a)_{n \ \mathbb{N}}$ is a limit path from Q to T_1 in $\mathcal{A}$. To prove that (ii) holds, consider the restriction $\mathcal{A}[T, A_T]$ of automaton $\mathcal{A}$ to states T and alphabet A_T. Then $(w_n)_{n \ \mathbb{N}}$ is a limit-path from T_1 to T in $\mathcal{A}[T, A_T]$. Moreover, since $\mathcal{A}$ is $\sharp$-acyclic and $\sharp$-stable, $\mathcal{A}[T, A_T]$ also is. Thus, we can apply the blowing lemma to $\mathcal{A}[T, A_T]$ and T_1, which proves that T is $\sharp$-reachable from T_1 in $\mathcal{A}[T, A_T]$, thus in $\mathcal{A}$ as well.

If $T_1 = Q$, the proof is complete. Otherwise, as long as $T_n = Q$, we use condition (iii) to build inductively a sequence $T = T_0, T_1, T_2, \cdots T_n$ such that for every $0 \leq k < n$, $T_k = T_{k+1}$ (condition (i))and T_k is $\sharp$-reachable from T_{k+1} in $\mathcal{A}$ (condition (ii)). Since $\mathcal{A}$ is $\sharp$-acyclic , $T_n = Q$ after at most $2^{\ Q\ }$ inductive steps.

Since $\sharp$-reachability is transitive, this proves that T is $\sharp$-reachable from Q. Since $\mathcal{A}$ is $\sharp$-stable, the only set $\sharp$-reachable from Q is Q thus $T = Q$, which completes the proof of the flooding lemma. □

Lemma 6.38 (Inductive step)**.** *Let $\mathcal{A}$ be a probabilistic automaton with states Q and $S_0, T \subseteq Q$. Suppose that $\mathcal{A}$ is $\sharp$-acyclic and T is limit-reachable from S_0. Then either $S_0 \subseteq T$ or there exists $S_1 = S_0$ such that S_1 is $\sharp$-reachable from S_0 in $\mathcal{A}$ and T is limit-reachable from S_1.*

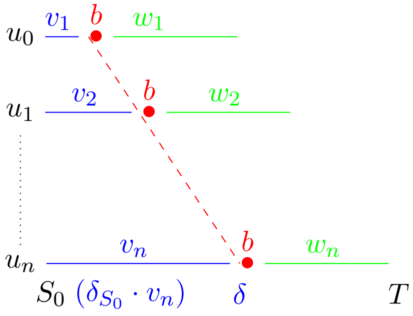


Figure 6.16: Construction of a sharp word from a limit-path

Proof. We prove by induction on $\ Q\ $. if $\ Q\ = 1$, then there is nothing to prove. Assume by induction that the result holds when $\ Q\ < n$ and assume that that $\ Q\ = n$ for some $n\ \ \mathbb{N}$. Let $(u_n)_{n\ \mathbb{N}}$ be a limit-path from S_0 to T. Let $A_0 = \ a\ \ A\ \ S_0 \cdot a = S_0\ $. For every $n\ \ \mathbb{N}$, let v_n be the longest prefix of u_n in A_0^*. Since every infinite subsequence of a limit-path is a limit-path, and since δ_Q is compact, we can suppose w.l.o.g. that $(\delta_{S_0} \cdot v_n)_{n\ \mathbb{N}}$ converges to some distribution $\delta\ \ \delta_Q$.

Suppose first that $\mathrm{Supp}(\delta) = S_0$. If $u_n\ \ A_0^*$ ♯-for infinitely many $n\ \ \mathbb{N}$ then $S_0 \subseteq T$. Otherwise, since A is finite we can suppose w.l.o.g. that there exists a letter $a\ \ A\ \ A_0$ such that for every $n\ \ \mathbb{N}$, $v_n a$ is a prefix of u_n. Let also w_n such that $u_n = v_n a w_n$. Let $S_1 = S_0 \cdot a$. Then $S_1 = S_0$ because $a\ \ A_0$ and S_1 is clearly ♯-reachable from S_0. Moreover $(w_n)_{n\ \mathbb{N}}$ is a limit-path from S_1 to T, this completes the proof.

Suppose now that $\mathrm{Supp}(\delta) = S_0$. Let $\mathcal{A}[S_0, A_0]$ the probabilistic automaton obtained from $\mathcal{A}$ by restriction to the alphabet A_0 and to the state space S_0. By definition of A_0, $\mathcal{A}[S_0, A_0]$ is stable and it is ♯-acyclic because $\mathcal{A}$ is. Either $\mathcal{A}[S_0, A_0]$ is ♯-stable or there exists an action a such that $S_0 \cdot a^\sharp = S_0$. In the latter case, let $S_1 = S_0 \cdot a^\sharp$, then $S_1 = S_0$, S_1 is ♯-reachable from S_0, and because $S_1 \subsetneq S_0$ and T is limit-reachable from S it follows that T is limit-reachable from S_1. If $\mathcal{A}[S_0, A_0]$ is ♯-stable, then according to the flooding lemma, the only support limit-reachable from S_0 is S_0, it follows that T is limit-reachable from S_0. This completes the proof. □

Using Lemma 6.38, one can construct inductively a path in the support graph between any pair of limit-reachable supports and thus yields the following proposition.

Proposition 6.39. *Let $\mathcal{A}$ be a probabilistic automaton with states Q and $S_0, T \subseteq Q$. Suppose that $\mathcal{A}$ is ♯-acyclic and T is limit-reachable from S_0 in $\mathcal{A}$. Then T is ♯-reachable from S_0 in $\mathcal{A}$.*

Proposition 6.39 establishes the direct implication of Theorem 6.31 and thus the decidability of the Value 1 Problem.

6.5.3 Complexity result

Theorem 6.40. *The value 1 problem for ♯-acyclic automata is* PSPACE*-complete.*

The proof of the above theorem follows from the following lemmata.

Lemma 6.41 (Upper bound)**.** *The value 1 for ♯-acyclic automata is* PSPACE.

Proof. Using *on-the-fly* techniques, one can construct the support graph using an non deterministic polynomial space, Savitch theorem [Sav70] terminates the proof. □

Problem 6.42 (Intersection of automata)**.** *Let $\mathcal{A}_1, \cdots, \mathcal{A}_n$ be a family of finite state deterministic automata over the same alphabet A, decide whether there exists $w \quad A^*$ such that w is accepted by $\mathcal{A}_i$ for all $1 \leq i \leq n$.*

This problem is known to be PSPACE-complete [Koz77].

Lemma 6.43 (Hardness)**.** *The value 1 for $\sharp$-acyclic automata is* PSPACE*-hard.*

Proof. We reduce the problem of intersection of automata to the value 1 problem for $\sharp$-acyclic automata.

Let $\mathcal{A}_1, \cdots, \mathcal{A}_n$ be a family of finite state deterministic automata over the same alphabet A, denote Q_i the set of states of the automaton $\mathcal{A}_i$ and F_i the set of accepting states for the automaton $\mathcal{A}_i$ for $1 \leq i \leq n$. We construct the probabilistic automaton $\mathcal{A}$ such that

- The set of state is $q_0 \quad \bigcup_{i=1}^{n} Q_i$,
- the set of actions is $\$ \quad A$,
- the set of accepting states is $\bigcup_{i=1}^{n} F_i$,
- the transitions of $\mathcal{A}$ are the same of the transitions of A_i plus a fresh transition from q_0 to q_0^i with probability $\frac{1}{n}$ where q_0^i is the initial state of the $\mathcal{A}_i$ for $0 \leq i \leq n$.

We show that $\mathcal{A}$ has value 1 if and only if there exists $w \quad A^*$ such that w is accepted by each $\mathcal{A}_i$.

We show the direct implication. Assume that $\mathcal{A}$ has value 1, since the only probabilistic transition is from q_0 to q_0^i it follows that there exists a word w accepted with probability 1. By construction of $\mathcal{A}$ we know that $w = \$u$ for some $u \quad A^*$ and u is accepted by each $\mathcal{A}_i$.

We show the converse implication. Let $w \quad A^*$ such that w is accepted by each $\mathcal{A}_i$. It follows directly that the word $\$w$ is accepted with probability 1.

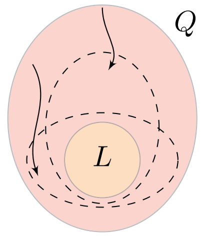
To establish the PSPACE-hardness, First notice that if each automaton $\mathcal{A}_i$ is $\sharp$-acyclic then the automaton $\mathcal{A}$ is. Second the PSPACE-completeness of Problem 6.42 holds even if the input automata are $\sharp$-acyclic . □

6.6 Discussion

The automaton depicted in Fig 6.9 is $\sharp$-acyclic . Moreover if the set of accepting state is $1, 2, 3, 4$, the this automaton has value 1 for any choice of the initial state. This is consequence of the fact that any computation will eventually reach the support $1, 2, 3, 4$ and never leaves it. This property is actually shared by all $\sharp$-acyclic automata and it called the leaf property. Formally, a leaf is:

Definition 6.44 (Leaf)**.** *Let $\mathcal{A}$ be a probabilistic automaton with states Q. A non-empty subset $R \subseteq Q$ is called a leaf if for every letter $a \quad A$, $R \cdot a = R$ and $R \cdot a^\sharp = R$.*

Lemma 6.45 (Leaf property)**.** *Let $\mathcal{A}$ be a probabilistic automaton with states Q. Suppose that $\mathcal{A}$ is $\sharp$-acyclic . Then there exists a unique leaf $\sharp$-accessible from Q. Every set limit-reachable from Q contains this leaf.*



The largest circle depicts the state space of some $\sharp$-acyclic automaton denoted Q and the smallest one depicts the leaf L. According to the leaf property, the leaf L is unique. Moreover, the limit-reachable sets (dashed circles) contain L and L is $\sharp$-reachable from Q.

Figure 6.17: Illustration of the Leaf property

Proof. Since $\mathcal{A}$ is $\sharp$-acyclic , there exists a leaf S $\sharp$-reachable from Q. Let T be another leaf, we shall prove that $S = T$.

We start with proving $T \subseteq S$. According to Proposition 6.33, there is a limit-path $(u_n)_{n \quad \mathbb{N}}$ from Q to S. A fortiori, $(u_n)_{n \quad \mathbb{N}}$ is a limit-path from T to S. Moreover, since T is a leaf, it is a-stable for every letter a thus $(u_n)_{n \quad \mathbb{N}}$ is a limit-path from T to $T \setminus S$. Moreover, since T is a leaf, the automaton $\mathcal{A}[T]$ obtained from $\mathcal{A}$ by restriction to T is $\sharp$-acyclic and $\sharp$-stable. According to the flooding lemma applied to $\mathcal{A}[T]$, $T = T \setminus S$, thus $T \subseteq S$.

By symmetry, $T = S$. Now we prove that every set limit-reachable from Q contains the leaf. Let R limit-reachable from Q and $(u_n)_{n \quad \mathbb{N}}$ a limit-path from Q to R. Since for every $a \quad A$, s is a-stable, then a fortiori $(u_n)_{n \quad \mathbb{N}}$ is a limit-path from S to $R \setminus S$. According to the flooding lemma applied to $\mathcal{A}[S]$, $R = R \setminus S$, thus $R \subseteq S$. □

This last property of $\sharp$-acyclic probabilistic automata concludes this section.

6.7 Conclusion

In this chapter we tackled two algorithmic problems for probabilistic automata. The first one is the emptiness problem and the result obtained are the followings:

- a simplified proof of the undecidability of the emptiness problem,
- the undecidability holds already for automata with two probabilistic transitions.

The second problem is the value problem and the result obtained are:

- the undecidability of the value 1 problem,
- the value 1 problem is undecidable even for automata with one probabilistic transition,
- the introduction of $\sharp$-acyclic probabilistic automata; a decidable sub class for the value 1 problem.

As far as we keep a game theoretic motivation for the study of probabilistic automata, interesting research directions are as follow:

1. identify decidable sub classes for the emptiness problem,
2. extend the sub class of $\sharp$-acyclic automata to a larger one,
3. extend the decidability result obtained for $\sharp$-acyclic automata to other model such as partially observable Markov decision processes.

4. introduce decidable sub classes such that the answer to value 1 problem depends quantitatively on the transition probabilities as opposed to $\sharp$-acyclic automata.

The first research direction seems to be very challenging since even for $\sharp$-acyclic automata the emptiness problem remains undecidable.

For the second direction together with Nathaël fijalkow, we introduced a new decidable subclass for the value 1 problem that subsumes the $\sharp$-acyclic automata but also other classes such as hierarchical automata [CSV09]. This new sub class is called Leaktight [FGO12] automata and decision procedure relies on algebraic techniques especially the work of Simon namely the forest factorization theorem [Sim90].

We also managed to extend the decidability result for $\sharp$-acyclic automata to the case of partially observable Markov decision processes, this result is presented in Chapter 7 where we present an EXPTIME algorithm to solve the value 1 problem for the so called $\sharp$-acyclic partially observable Markov decision processes.

Chapter 7

Partially Observable Markov Decision Processes

Contents

Abstract We consider Partially Observable Markov Decision Processes (POMDP) with reachability objectives. Whereas the existence of an almost-sure or a positive strategy is decidable for such POMDPs, the values are not computable and even the value 1 problem is undecidable. In this problem one asks whether the supremum over all possible strategies the probability to achieve the reachability objective is equal to 1. Our main result is to identify a class of POMDPs for which the value 1 problem can be decided in EXPTIME.

7.1 Introduction

Partially Observable Markov Decision Processes are the natural extension of Markov decision processes to the setting of partial information games. In a partially observable Markov decision process, the setting is the same as in Chapter 3 with the difference that each state is labeled with a color. The decision maker cannot observe the states themselves but only their colors, thus if two plays are colored the same way, its choice should be the same in both cases; in other words the strategies for the controller are mappings from colors sequences to actions.

While in a fully observable Markov decision process ω-regular objectives such as parity games can be solved in polynomial time [CY95, CJH04], in POMDP it is not the case and even deciding whether the value of a POMDP with reachability objective is 1 or greater than $\frac{1}{2}$ is not decidable [Paz71, MHC03b, GO10]. We proved in the previous chapter that this undecidability result holds even if all states are labeled with the same color i.e. for probabilistic automata (c.f. Chapter 6) and identify a decidable subclass.

Contribution and result We extend the decidability result of Chapter 6 to the case of POMDPs, we consider a class of POMDPs called $\sharp$-*acyclic* POMDPs and we show that the value 1 problem is decidable for this class. The proof is based on the generalization of the operation of iteration and The construction of a perfect information two-player game $\mathcal{G}$ that abstracts the behavior of a $\sharp$-acyclic POMDP $\mathcal{M}$: the two-player game is won by the first player if and only if $\mathcal{M}$ has value 1 as opposed probabilistic automata where the problem of the value one is reduced to a reachability problem over graphs. Another difference holds in the complexity of the decision procedure used for the two models. Indeed, while for $\sharp$-acyclic probabilistic automata the value 1 problem is decided in PSPACE, the value 1 problem for $\sharp$-acyclic POMDP is decidable in EXPTIME.

Outline of the chapter

– In Section 7.2 we introduce POMDPs and notations related to this model.

– In Section 7.3 we introduce the class of the so called $\sharp$-*acyclic POMDPs* and state our main theorem.

– In Section 7.4 we define the so called *knowledge game* and prove the main result.

7.2 Partially Observable Markov Decision Processes

Definition 7.1 (POMDP)**.** *A Partially Observable Markov Decision Process (POMDP) is a tuple* $\mathcal{M} = (Q, A, \mathcal{O}, p, \mathrm{Obs}, \delta_0)$ *where:*

Q is a finite set of states,

A is a finite set of actions,

$\mathcal{O}$ is a finite set of observation,

p is a function $p : Q \times A \quad \Delta(Q)$,

Obs is a function $\mathrm{Obs} : \mathcal{O} \quad 2^Q$,

δ_0 is an initial distribution in $\Delta(Q)$.

We assume that for $(o, o\ ) \quad \mathcal{O}$ we have $\mathrm{Obs}(o) \setminus \mathrm{Obs}(o\ ) = \qquad o = o$ and for a subset $S \subseteq Q$, we write $\mathrm{Obs}^{-1}(S) = o$ if $S \subseteq \mathrm{Obs}(o)$.

Remark 7.2. *We assume that for every state $q \quad Q$ and every action $a \quad A$ the function $p(q, a)$ is defined, i.e. every action can be played from every state.*

Example 7.3. *Consider the POMDP depicted in Fig 7.1. The initial distribution is at random between states 2 and 3 and the play is winning if it reaches . The states with similar colors cannot be distinguished.*

Definition 7.4. *Let $S \subseteq Q$ be a support and a letter a, we define $\mathrm{Acc}(S, a)$ as the following set of states:*

$$\mathrm{Acc}(S, a) = \quad q \quad Q \quad s \quad S, p(s, a)(q) > 0 \quad .$$

As opposed to probabilistic automata, in a POMDP the controller can refine its knowledge about the play using the observations.

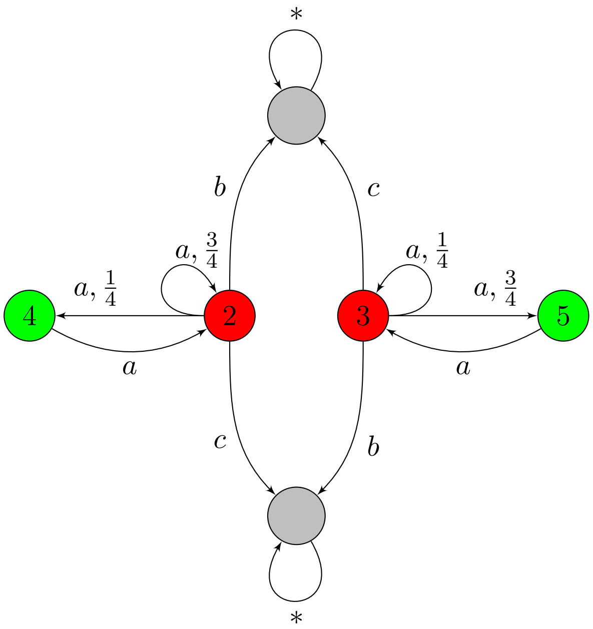


Figure 7.1: Partially observable Markov decision process

Definition 7.5 (Outcome of actions)**.** *Let $\mathcal{M}$ be a POMDP, for a subset $S \subseteq Q$ and an action $a \quad A$ we write*

$$S \cdot a = \quad R \quad o \quad \mathcal{O}, R = \mathrm{Acc}(S, a) \setminus \mathrm{Obs}(o) \quad ,$$

and for a collection of subsets $\mathcal{R} \subseteq 2^Q$ we write

$$\mathcal{R} \cdot a = \bigcup_{S \quad \mathcal{R}} S \cdot a \ .$$

Remark 7.6. *According to Remark 7.2 it follows that $S \cdot a$ is always nonempty.*

Since the states are not fully observable, in order to ensure a given objective, the controller chooses the next action to apply in function of the initial distribution, the sequence of action played, and the sequence of observations observed. Such strategies are said to be *observational*. Formally,

Definition 7.7 (Observational strategy)**.** *An observational strategy is a function $\sigma : \mathcal{O}^*\mathcal{O} \quad \Delta(A)$.*

In the very general case of stochastic two-player signal games the use of randomized strategy (c.f Section 3.3) is necessary for the value to exist [BGG09]. In the case of POMDP according to [Gim09, CDGH10] it is sufficient to use pure strategies (c.f Section 3.3).

For a given strategy $\sigma : \mathcal{O}^*\mathcal{O} \quad A$ and an initial distribution δ_S, we define the measure $\mathbb{P}^{\sigma}_{\delta_S}$ (c.f. Section 3.3). We also define the random variable O_n with values in $\mathcal{O}$ that gives the observation after n steps:

$$O_n(s_0 a_0 s_1 a_1 \cdots) = \mathrm{Obs}^{-1}(s_n) \ .$$

As usual S_n denotes the random variable that gives the current state at steps n and A_n denotes the random variable that gives the action played at step n.

Definition 7.8 (Knowledge)**.** *Let $\mathcal{M}$ be a POMDP and δ_0 be an initial distribution, the knowledge K_n is*

$$\begin{aligned} K_0 &= \mathrm{Supp}(\delta_0) \ , \\ K_{n+1} &= \mathrm{Acc}(K_n, A_n) \setminus O_{n+1} \ . \end{aligned}$$

In the sequel we will concentrate on reachability objective, hence when referring to the value of a POMDP it is implied that the objective is a reachability objective.

Problem 7.9 (Value 1 Problem)**.** *Let $\mathcal{M}$ be a POMDP, $\delta_0 \quad \Delta(Q)$ be an initial distribution, and $T \subseteq Q$ be a subset of target states. Decide whether:*

$$\sup_{\sigma} \mathbb{P}^{\sigma}_{\delta_0}(\quad n \quad \mathbb{N}, \ S_n \quad T) = 1 \ .$$

In the rest of the present chapter, for a POMDP $\mathcal{M}$ with initial distribution δ_0 we use the notation $\mathrm{Val}_{\mathcal{M}}$ to denote the value of $\mathcal{M}$ when the initial distribution is δ_0 for the reachability objective.

For technical reasons, we suppose that the states of the support of the initial distribution are associated to the same observation. Formally, we suppose that there exists $o \quad \mathcal{O}$ such that $\mathrm{Supp}(\delta_0) \subseteq \mathrm{Obs}(o)$. This does not restrict the model since

Proposition 7.10. *For every POMDP $\mathcal{M}$ there exists a POMDP $\mathcal{M}$ computable in polynomial time such that:*

$$\mathrm{Val}_{\mathcal{M}} = 1 \qquad \mathrm{Val}_{\mathcal{M}'} = 1 \ ,$$

$$o \quad \mathcal{O}, \ \mathrm{Supp}(\delta_0) \subseteq \mathrm{Obs}(o) \ .$$

Proof. Let $\mathcal{M} = (Q , A , \mathcal{O} , p , \mathrm{Obs} , \delta_0)$ be the POMDP obtained as follows:

- $Q = Q \quad q_\$ \ ,$
- $A = A \quad \$ \ ,$
- $\mathcal{O} = \mathcal{O} \quad o_\$ \ ,$
- $p : Q \times A \quad \Delta(Q)$ such that $p (q_\$, \$)(q) = \delta_0(q)$, $p (q, \$)(q) = 1$, and $p (q, a)(q) = p(q, a)(q)$ for every $q \quad Q$ and $a \quad A$.
- $\mathrm{Obs} : \mathcal{O} \quad 2^{Q'}$ such that $\mathrm{Obs}(o_\$) = \quad q_\$ \ ,$
- $\delta_0(q_\$) = 1.$

Let us show the direct implication,
assume that for every $\varepsilon > 0$ there exists a strategy σ such that

$$\mathbb{P}^{\sigma}_{\delta_0}(\quad n \geq 0, \ S_n \quad T) > 1 - \varepsilon \ ,$$

we define the strategy σ that consists in playing \$ when the observation $o_\$$ and then playing accordingly to σ. The strategy ensures

$$\mathbb{P}^{\sigma'}_{\delta'_0}(\quad n \geq 0, \ S_n \quad T) > 1 - \varepsilon \ ,$$

since

$$q \quad \mathrm{Supp}(\delta_0),\ \sigma\ (o_\$)(q) = \delta_0(q)\ \ .$$

Let us prove the converse implication, assume that for every $\varepsilon > 0$ there exists a strategy σ such that

$$\mathbb{P}^{\sigma'}_{\delta_0}(\ \ n \geq 0,\ S_n \quad T) > 1 - \varepsilon\ \ ,$$

we define the strategy σ the following way:

$$o \quad \mathcal{O}^*, \sigma(o) = \sigma(o_\$ o)\ \ ,$$

using similar arguments as in the first part of the proof, one can see that σ satisfies

$$\mathbb{P}^{\sigma}_{\delta_0}(\ \ n \geq 0,\ S_n \quad T) > 1 - \varepsilon\ \ .$$

□

Example 7.11. *Consider the POMDP of Fig 7.1, the value of that game is 1 when the initial distribution is the uniform distribution over the set* $2, 3$. *Indeed, consider the strategy that plays long sequences of* a^2 *then compares the frequencies of observing* $o = \mathrm{Obs}^{-1}(\ 2, 3\ )$ *and* $o\ = \mathrm{Obs}^{-1}(\ 4, 5\ )$*; If* o *was observed more than* o *then with high probability the initial state is* 2 *and by playing* b *state is reached with very high probability. Otherwise with high probability the play is in* 3 *and by playing* c *again the play is winning very probability. Note that the controller can differentiate between state* 2 *and* 3 *with arbitrarily high probability as he can just play longer sequences of* a^2 *but he cannot win almost-surely since she always has to take a risk and chooses between actions* b *and* c *at sometime. This example shows that the strategies ensuring the value 1 can be quite elaborated: the choice not only depends on the time and the sequence of observations observed, but also depends on the empirical frequency of the observations.*

Remark 7.12. *If the POMDP of Fig 7.1 we had* $p(2, a)(2) = p(3, a)(3) = \frac{1}{2}$*, then the value is strictly less than 1. Hence the transition probabilities do matter.*

7.3 ♯-acyclic Partially Observable Markov Decision Processes

The value 1 problem is undecidable in general, our goal is to generalize the result obtained in Chapter 6 and show that the value 1 problem is decidable for the so called ♯*-acyclic POMDP*. But before, we introduce the notion of limit-reachability in order to state the value 1 problem in an alternative way.

Definition 7.13 (Limit-reachability)**.** *Let* S *be a support and* $\mathcal{R}$ *be a collection of supports, we say that* $\mathcal{R}$ *is limit-reachable from* S *if there exists a sequence of strategies* $(\sigma_n)_{n \quad \mathbb{N}}$ *such that for every* $\varepsilon > 0$ *there exists* $n \quad \mathbb{N}$ *which satisfies:*

$$\mathbb{P}^{\sigma_n}_{\delta_S}\Big(\ \ m \quad \mathbb{N},\ \ R \quad \mathcal{R},\ \mathbb{P}^{\sigma_n}_{\delta_S}(S_m \quad R \quad O_0 A_0 \cdots O_m) \geq 1 - \varepsilon\Big) \geq 1 - \varepsilon\ \ .$$

The sequence $(\sigma_n)_{n \quad \mathbb{N}}$ *is called a limit-strategy.*

For a POMDP $\mathcal{M}$, if the target set of states is T, we consider the collection $\mathcal{T}$ that consists of the nonempty subsets of T. Our decision procedure is based on the fact that $\mathcal{M}$ has value 1 if and only if $\mathcal{T}$ is limit-reachable from the support of the initial distribution.

Definition 7.14 (Observable target)**.** *Let T be a set of target states, we say that T is observable if there exists $\mathcal{O}$ $\subseteq \mathcal{O}$ such that for every state s we have*

$$T = \bigcup_{o \quad \mathcal{O}'} \mathrm{Obs}(o) \ . \tag{7.1}$$

When Equation (7.1) does not hold, we say that the set T is unobservable.

Proposition 7.15. *Let $\mathcal{M}$ be a POMDP, δ_0 and initial distribution. Assume that T is observable then, $\mathcal{M}$ has value 1 if and only if $\mathcal{T}$ is limit-reachable from* $\mathrm{Supp}(\delta_0)$*.*

Proof. Let σ a strategy, and assume w.l.o.g. that T is a singleton $\ \ t\ \ $, then $\mathcal{T}$ is as well. Since T is observable we have:

$$\mathbb{P}^{\sigma}_{\delta_0}(S_n \quad T \quad O_0 \cdots O_n) = \mathbb{1}_{O_n = \mathrm{Obs}^{-1}(\ t\ )} = \mathbb{1}_{S_n \ \ T} \ .$$

And since $\mathbb{1}_{S_n \ \ T} \geq 1 - \varepsilon$ if and only if $S_n \quad T$ it follows that for every $\varepsilon > 0$ and every strategy σ we have:

$$\mathbb{P}^{\sigma}_{\delta_S}\big(\ m \quad \mathbb{N},\ \mathbb{P}^{\sigma}_{\delta_S}(S_m \quad T \quad O_0 A_0 \cdots O_m) \geq 1 - \varepsilon\big) \geq 1 - \varepsilon$$
$$\mathbb{P}^{\sigma}_{\delta_S}(\ m \quad \mathbb{N},\ S_m \quad T) \geq 1 - \varepsilon \ .$$

This terminates the proof. □

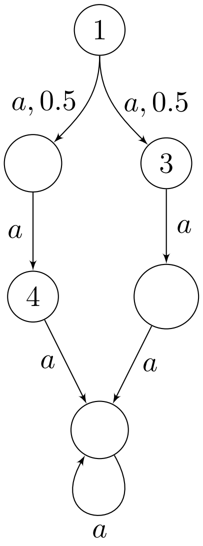


Figure 7.2: The value of this POMDP is 1 but $\mathcal{T}$ is not limit-reachable from Supp(δ_0).

Fig 7.2 shows that Proposition 7.15 does not hold in the case where the objectives are not visible. Indeed, assume that $\mathcal{O} = \quad o$ and that the target set is $\quad , \quad$, then the value of this game is 1 since $\mathbb{P}^{a^\omega}_1(\ n \quad \mathbb{N},\ S_n \qquad , \quad) = 1$, but $\quad , \quad$ is not limit-reachable from Supp(δ_0), since for every $n \quad \mathbb{N}$ we have:

$$\mathbb{P}^{a^\omega}_1\left(S_n \quad \{\ \ ,\ \ \}\right) \leq \frac{1}{2} \ .$$

Fortunately, for the value 1 problem there exists a construction such that for every POMDP $\mathcal{M}$ with unobservable objective, there exists a POMDP $\mathcal{M}$ with observable objective such that $\mathcal{M}$ has value 1 if only if $\mathcal{M}$ has value 1. Hence our decision procedure holds also for unobservable objectives.

Proposition 7.16. *For every POMDP $\mathcal{M}$, there exists a POMDP $\mathcal{M}$ computable in linear time such that:*

the target set in $\mathcal{M}$ is observable.

$\text{Val}_{\mathcal{M}} = 1 \qquad \text{Val}_{\mathcal{M}'} = 1.$

Proof. Let $\mathcal{M}$ be a POMDP and let T a set of target states. We construct $\mathcal{M} = (Q\ , A\ , \mathcal{O}\ , p\ , \text{Obs}\ , \delta_0)$ such that:

- $Q \ = (Q \times \ 0,1 \ ) \qquad , \quad .$
- $A \ = A \qquad \$$ such that for every $s \quad Q$, $p\ ((s,0),\$)(\quad) = 1$ and $p\ ((s,1),\$)(\quad) = 1$.
- $p\ : Q\ \times A \qquad \Delta(Q)$ such that for every state $q,t \quad Q$, action $a \quad A$ and $i \quad 0,1$ we have:

$$p\ ((s,i),a)(t,1) = \begin{cases} p(s,a)(t) \text{ if } (s \quad T) \quad (i=1)\ , \\ 0 \text{ otherwise.} \end{cases}$$

$$p\ ((s,i),a)(t,0) = \begin{cases} p(s,a)(t) \text{ if } (s \quad T) \quad (i=0)\ , \\ 0 \text{ otherwise.} \end{cases}$$

- $\mathcal{O} \ = \mathcal{O} \quad o\ , o$ such that $\text{Obs}(o\ ) = \quad$ and $\text{Obs}(o\ ) = \quad .$
- $\text{Obs}\ : \mathcal{O} \qquad 2^{Q'}$ such that $\text{Obs}\ (o) = \text{Obs}(o) \times \ 0,1\ .$
- for every $q \quad Q$, $\delta_0(q,0) = \delta_0(q)$
- $T \ =$

We show that $\text{Val}_{\mathcal{M}'} = 1$ if and only if $\text{Val}_{\mathcal{M}} = 1$.

Assume that $\text{Val}_{\mathcal{M}'} = 1$ and let σ and $\varepsilon > 0$ such that

$$\mathbb{P}^{\sigma'}_{\delta'_0}(\ n \quad \mathbb{N}^*,\ S_{n-1} = \quad) \geq 1 - \varepsilon\ ,$$

hence

$$\mathbb{P}^{\sigma'}_{\delta'_0}(\ n \quad \mathbb{N}^*,\ S_{n-1} \quad Q \times \ 1\ ) \geq 1 - \varepsilon\ .$$

Let $\sigma : \mathcal{O}^*\mathcal{O} \qquad \mathcal{O}$ be the restriction σ defined on every history h such that $h \quad \mathcal{O}^*\mathcal{O}$, then we have

$$\mathbb{P}^{\sigma}_{\delta'_0}(\ n \quad \mathbb{N}^*,\ S_{n-1} \quad Q \times \ 1\ ) \geq 1 - \varepsilon\ .$$

It follows that:

$$\mathbb{P}^{\sigma}_{\delta_0}(\ n \quad \mathbb{N},\ S_n \quad T) \geq 1 - \varepsilon\ .$$

Assume that $\text{Val}_{\mathcal{M}} = 1$ and let σ and $\varepsilon > 0$ such that:

$$\mathbb{P}^{\sigma}_{\delta_0}(\ n \quad \mathbb{N},\ S_n \quad T) \geq 1 - \varepsilon\ .$$

Let σ be a strategy such that for every $h \quad \mathcal{O}^*\mathcal{O}$ we have

$$\sigma\ (h) = \begin{cases} \sigma(h) \text{ if } \mathbb{P}^{\sigma}_{\delta_0}(S_n \quad Q\times \quad 1 \quad h) < 1-\varepsilon \\ \$ \text{ if } \mathbb{P}^{\sigma}_{\delta_0}(S_n \quad Q\times \quad 1 \quad h) \geq 1-\varepsilon \end{cases}$$

Since by construction of $\mathcal{M}$ we have

$$\mathbb{P}^{\sigma}_{\delta'_0}(\quad n \quad \mathbb{N},\quad m\geq n,\ S_m \quad Q\times \quad 1\ )\geq 1-\varepsilon\ ,$$

it follows that the action $\$$ is chosen at sometime thus

$$\mathbb{P}^{\sigma'}_{\delta'_0}(\quad n \quad \mathbb{N},\ S_n = \quad)\geq 1-\varepsilon\ ,$$

which terminates the proof. □

In the sequel we consider only reachability objectives with observable target sets.

The following Proposition implies that the value 1 problem depends only the support of the initial distribution.

Proposition 7.17. *Let $S \subseteq Q$ be a support, $\delta \quad \Delta(S)$ be a distribution over S, $\mathcal{R}$ be a collection of supports, and $(\sigma_n)_{n \quad \mathbb{N}}$ be sequence of strategies. Assume that:*

$$\varepsilon > 0,\quad n \quad \mathbb{N},\ \mathbb{P}^{\sigma_n}_{\delta}\big(\ m \quad \mathbb{N},\quad R \quad \mathcal{R},\ \mathbb{P}^{\sigma_n}_{\delta}(S_m \quad R \quad O_0A_0\cdots O_m)\geq 1-\varepsilon\big)\geq 1-\varepsilon\ ,$$

then $(\sigma_n)_{n \quad \mathbb{N}}$ is a limit-strategy from S to $\mathcal{R}$

Proof. If $\delta = \delta_S$ then there result is trivial. If not, the result follows from the fact that for every $E \quad (Q\times A)^\omega$, $\varepsilon > 0$, and $n \quad \mathbb{N}$:

$$\left(\sum_{s \quad S}\delta(s)\mathbb{P}^{\sigma_n}_s(E)\geq 1-\varepsilon\right) \quad = \quad \left(\frac{1}{S}\sum_{s\quad S}\mathbb{P}^{\sigma_n}_s(E)\geq 1-\frac{S}{\min_{s\quad S}\ \delta(s)}\varepsilon\right)\ .$$

□

7.3.1 Iteration of actions

The key notion in the definition of ♯-acyclic POMDPs is the one of *iteration of actions.* As for probabilistic automata, we define the operation of iteration of actions.

Definition 7.18 (Stability)**.** *Let $S \subseteq Q$ be a support and $a \quad A$ be an action, then S is a-stable if $S\cdot a = \quad S\ $.*

Definition 7.19 (a-recurrence)**.** *Let $S \subseteq Q$ be a support and $a \quad A$ be an action. Assume that S is a-stable then a state $r \quad S$ is a-reachable from s if there exists $n > 0$ such that $p(s,a^n)(r) > 0$. A state $s \quad S$ is a-recurrent if for any state $r \quad S$,*

$$(r \textit{ is a-reachable from } s) \quad = \quad (s \textit{ is a-reachable from } r)\ .$$

Let S be a support and a an action, $S\cdot a^\sharp$ is the collection of all possible outcomes after repeating a an arbitrarily large number of time. Formally,

Definition 7.20 (Iteration)**.** *Let S and S be two supports and a an action such that*

$S \quad S \cdot a$,

$S \quad$ is the largest a-stable subset of S,

then

$$S \cdot a^\sharp = \{a\text{-recurrent states of } S \ \} \quad (S \cdot a \quad S \ ) \ .$$

Remark 7.21. *Note that since $S \cdot a$ is always nonempty and since there exists always an a-recurrent state the collection of sets $S \cdot a^\sharp$ is always nonempty.*

Definition 7.22 (♯-stability)**.** *Let $S \subseteq Q$ be a support and $a \quad A$ be a letter, then S is $a^\sharp$-stable if $\ S \quad S \cdot a$ and $S \cdot a^\sharp = \quad S \ $.*

Proposition 7.23. *Assume $S \quad S \cdot a^\sharp$, then S is $a^\sharp$-stable.*

Proof. Let S be a support, $S \ $ the largest a-stable subset of S, and $a \quad A$ an action. By Definition 7.20, if $S \quad S \cdot a^\sharp$ then S is the set of a-recurrent states in $S \ $. Since $S \ \subseteq S$ it follows that $S = S \ $ and by definition of $S \ $, S is a-stable □

In the rest of the chapter, we denote $A^\sharp$ the set $\{a^\sharp \quad a \quad A\}$.

Proposition 7.24. *Let S be a support and $a \quad A$ an action, then $S \cdot a$ is limit-reachable from S. Moreover if $\ S \quad S \cdot a$, then $S \cdot a^\sharp$ is also limit-reachable from S.*

Proof. Consider the constant sequence $\sigma_n = \sigma$ such that $\sigma(O^*O) = a$ is a limit-strategy from S to $S \cdot a$, since we have:

$$\mathbb{P}^{\sigma}_{\delta_S}(K_1 \quad S \cdot a) = 1 \ .$$

And by definition of the knowledge:

$$\mathbb{P}^{\sigma}_{\delta_S}(S_1 \quad K_1 \ \ O_0 A_0 O_1) = 1 \ .$$

Hence

$$\mathbb{P}^{\sigma_n}_{\delta_S}\left(\mathbb{P}^{\sigma}_{\delta_S}(S_1 \quad K_1 \ \ O_0 A_0 O_1) = 1\right) = 1 \ .$$

Choosing $m = 1$ and $R = K_1$ in Definition 7.13 finishes the prove of the first part.

Assume that $\ S \quad S \cdot a$, we show that the sequence σ_n defined as follows

$$\sigma_n(o^k) = \begin{cases} a \text{ if } k \leq n \quad \mathrm{Obs}^{-1}(S) = o \ , \\ \text{play anything otherwise.} \end{cases}$$

is a limit-strategy from S to $S \cdot a^\sharp$ i.e. satisfies the equation of Definition 7.13. To show that σ_n is a limit-strategy from S to $S \cdot a$, denote $o = \mathrm{Obs}^{-1}(S)$, $S \ $ the largest a-stable support included in S, and $S \ $ the set of a-recurrent states in $S \ $.

We show that

$$\mathbb{P}^{\sigma_n}_{\delta_S}\left(S_{n+1} \quad S \qquad 0 \leq i \leq n, \ (O_i = o) \quad (A_i = a)\right) \underset{n}{\longrightarrow} \quad 1 \ . \tag{7.2}$$

We distinguish between two cases.

If $S \quad S \ = \ $, let $x = \min_{s \ S \ S'} \sum_{t \ S} p(s,a)(t)$, then since $S \quad S \ = \ $ we have $x > 0$, it follows that:

$$\mathbb{P}^{\sigma_n}_{\delta_S}\left(S_{n+1} \quad S \quad S \quad (S_n \quad S \quad S \ ) \quad (A_n = a)\right) \leq (1 - x) \ .$$

Hence

$$\begin{aligned}
&\mathbb{P}^{\sigma_n}_{\delta_S}(S_{n+1} \quad S \quad S \qquad 0 \le i \le n,\ (O_i = o) \quad (A_i = a))\\
&= \mathbb{P}^{\sigma_n}_{\delta_S}(\ \ k \le n+1,\ S_k \quad S \quad S \qquad 0 \le i \le n,\ (O_i = o) \quad (A_i = a)) \le (1-x)^n \underset{n}{\text{——}} \quad 0\ ,\\
&= \quad \mathbb{P}^{\sigma_n}_{\delta_S}(S_n \quad S \qquad 0 \le i \le n,\ (O_i = o) \quad (A_i = a)) \underset{n}{\text{——}} \quad 1\ .
\end{aligned} \tag{7.3}$$

On the other hand, since $(S\ , a)$ induces a Markov chain, it follows that

$$\mathbb{P}^{\sigma_n}_{\delta_S}(S_{n+1} \quad S \quad (\ \ m \le i \le n,\ S_i \quad S\ ) \quad (\ \ 0 \le i \le n,\ A_i = a)) \underset{n}{\text{——}} \quad 1\ .$$

Equation (7.2) follows.

In the case where $S \quad S\ = \quad$, Equation (7.3) applies directly and Equation 7.2 follows.

To see that the sequence $(\sigma_n)_n$ is a limit-strategy from S to $S{\cdot}a$, notice that while being consistent with one of the strategies σ_n, either there exists $0 \le i \le n$ such that $O_i = o$ hence $K_i \quad S \cdot a \quad S$ and

$$\mathbb{P}^{\sigma_n}_{\delta_S}(S_i \quad K_i) = 1\ ,$$

or for every $0 \le i \le n,\ O_i = o$. Thus $K_i = S,$ then by Equation (7.2):

$$\mathbb{P}^{\sigma_n}_{\delta_0}(S_n \quad S\ ) \underset{n}{\text{——}} \quad 1\ ,$$

noticing that $S \qquad S \cdot a^\sharp$ terminates the proof. □

7.3.2 ♯-acyclic POMDP

Definition 7.25 (Knowledge graph)**.** *Let $\mathcal{M}$ be a POMDP, the knowledge graph $\mathcal{G}_\mathcal{M}$ of $\mathcal{M}$ is the labelled graph obtained as follows:*

The states are the non empty subsets of Q,

The triple (S, a, T) is an edge if $\ T \qquad S \cdot a$ and the triple $(S, a^\sharp, T)$ is an edge if $\ S \qquad S \cdot a$ and $\ T \qquad S \cdot a^\sharp$.

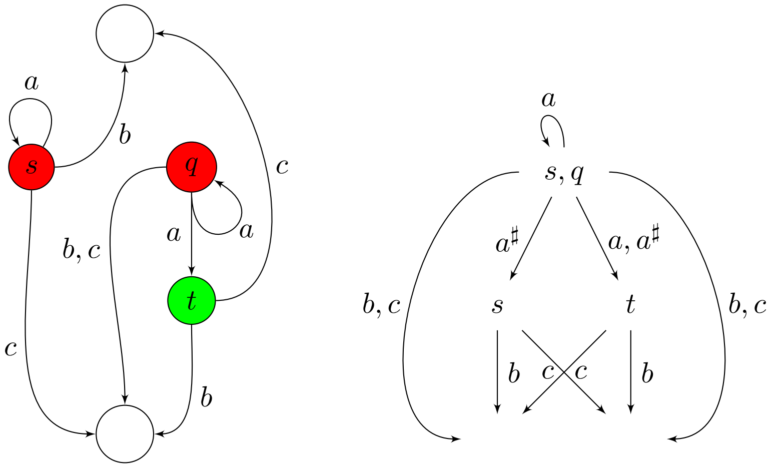
Example 7.26. *In Fig 7.3(a) is depicted a POMDP, where the initial are states s and q cannot be distinguished. In Fig 7.3(b) is the knowledge graph associated to it.*

Definition 7.27 (♯-acyclic POMDP)**.** *Let $\mathcal{M}$ be a POMDP and $\mathcal{G}_\mathcal{M}$ the associated knowledge graph. $\mathcal{M}$ is ♯-acyclic if the only cycles in $\mathcal{G}_\mathcal{M}$ are self loops.*

The main result is the following.

Theorem 7.28. *Given a ♯-acyclic POMDP $\mathcal{M}$ and an initial distribution δ_0, it is decidable whether* $\mathrm{Val}(\mathrm{Supp}(\delta_0)) = 1$*. Moreover it depends only on the support of δ_0.*

To prove the main theorem we define a perfect information two-player game played on the knowledge graph. We show that winning strategies exist if and only if the POMDP has value 1. Details of the game and the proof of Theorem 7.28 are given in Section 7.4.



(a) A ♯-acyclic POMDP (b) The knowledge graph

Figure 7.3: A POMDP and its knowledge graph

7.4 Deciding the Value 1

Our goal is to show that we can decide whether a ♯-acyclic POMDP has value 1. We prove that given a POMDP $\mathcal{M}$, there exists a two-player (verifier and falsifier) perfect information game where verifier wins if and only if $\mathrm{Val}_{\mathcal{M}} = 1$.

7.4.1 The knowledge game

We first explain how to construct the game and how it is played. For a given POMDP $\mathcal{M}$, an initial distribution δ_0, and a set of target states T. Let $\mathcal{G}_{\mathcal{M}}$ be the knowledge graph associated to $\mathcal{M}$. Assume that $\mathrm{Supp}(\delta_0) = S$, the knowledge game is played on $\mathcal{G}_{\mathcal{M}}$ as follows:

- Verifier chooses either an action a A or if S $S \cdot a$ he can also choose an action a $A^{\sharp}$,
- falsifier chooses a successor R $S \cdot a$ and R $S \cdot a^{\sharp}$ in the second case.
- the play continues from the new state R.

Verifier wins if the game reaches a support R such that $R \subseteq T$.

Definition 7.29 (♯-reachability). *Let S and R be two supports, R is ♯-reachable from S if verifier has a strategy to reach R or one of its nonempty subsets from S in the knowledge game.*

We say that a collection of supports $\mathcal{R}$ is ♯-reachable from a support S if there exists a strategy for verifier to reach a support in $\mathcal{R}$ against any strategy of falsifier in the knowledge game.

Example 7.30. *In the POMDP of Fig 7.3, assume that the initial distribution δ_0 is at random between state s and q. The value of the initial distribution is 1 because the controller can play long sequences of a and if the only observation observed is red, then with probability arbitrarily close to 1 the play is in state s otherwise with probability 1 the game is in state q. On the other hand, verifier has a strategy to win from s, q . This strategy consists in choosing action $a^{\sharp}$ from the initial state, then playing action c if falsifier's choice is t and action b if falsifier's choice is s .*

7.4.2 Proof of Theorem 7.28

The proof of Theorem 7.28, is split into Proposition 7.31 and Proposition 7.33. The former proposition establishes that if verifier has a winning strategy in the knowledge game $\mathcal{G}_{\mathcal{M}}$, then $\mathrm{Val}_{\mathcal{M}} = 1$. This proposition not only proves the direct implication of Theorem 7.28, but also shows that direct implication holds whether the POMDP is $\sharp$-acyclic or not.

Proposition 7.31. *Let $\mathcal{M}$ be a POMDP. Assume that verifier has a winning strategy in $\mathcal{G}_{\mathcal{M}}$, then* $\mathrm{Val}_{\mathcal{M}} = 1$.

The proof of Proposition 7.31 follows from the next lemma.

Lemma 7.32. *Let S be a support and $\mathcal{T}$ be a collection of supports such that $\mathcal{T}$ is $\sharp$-reachable from S then either $S \quad \mathcal{T}$ or there exists a collection of support $\mathcal{R}$ such that*

i) $\mathcal{R} \setminus \mathcal{T} = \quad .$

ii) *$\mathcal{R}$ is $\sharp$-reachable from S.*

iii) *$\mathcal{T}$ is limit-reachable from every support in $\mathcal{R}$.*

Proof. If $S \quad \mathcal{T}$ then there is nothing to prove. Assume that $S \quad \mathcal{T}$ and let σ be a positional strategy for verifier that allows her to reach $\mathcal{T}$ from S in $\mathcal{G}_{\mathcal{M}}$.

Let $\mathcal{R}$ be the collection of supports such that:

$$\mathcal{R} = \Big\{ R \quad a \quad A \quad A^{\sharp},\ (R \cdot a \subseteq \mathcal{T}) \quad (R \quad \mathcal{T}) \Big\} \ .$$

This collection is nonempty since $S \,/\, \mathcal{T}$ and $\mathcal{T}$ is $\sharp$-reachable from S. *i)* holds by choice of $\mathcal{R}$. *ii)* holds because Either $s \quad \mathcal{R}$ or since S is a winning support, the strategy σ allows verifier to reach $\mathcal{R}$ from S in $\mathcal{G}_{\mathcal{M}}$. *iii)* holds because according to Proposition 7.24 the collection $S \cdot a$ is limit-reachable from S. □

Proof of Proposition 7.31. Let $\mathcal{M}$ be a POMDP and δ_0 be an initial distribution and assume that $\mathcal{T}$ is $\sharp$-reachable from $\mathrm{Supp}(\delta_0)$ in $\mathcal{G}_{\mathcal{M}}$. By Lemma 7.32, we know that there exists $\mathcal{R}_0$ such that:

i) $\mathcal{R}_0 \setminus \mathcal{T} = \quad .$

ii) $\mathcal{R}_0$ is $\sharp$-reachable from $\mathrm{Supp}(\delta_0)$.

iii) $\mathcal{T}$ is limit-reachable from every support in $\mathcal{R}_0$.

Since $\mathcal{R}_0$ is $\sharp$-reachable from $\mathrm{Supp}(\delta_0)$, applying Lemma 7.32 again we can construct a collections of supports $\mathcal{R}_1$ such that

i) $\mathcal{R}_1 \setminus \mathcal{R}_0 = \quad .$

ii) $\mathcal{R}_1$ is $\sharp$-reachable from $\mathrm{Supp}(\delta_0)$.

iii) $\mathcal{R}_0$ is limit-reachable from every support in $\mathcal{R}_1$.

Since $\mathrm{Supp}(\delta_0)$ is a winning support, repeating this inductive construction, there exists $n \leq 2^Q$ such that $\mathrm{Supp}(\delta_0) \quad \mathcal{R}_n$ and since limit-reachability is a transitive property it follows that $\mathcal{T}$ is limit-reachable from $\mathrm{Supp}(\delta_0)$, thus according to Proposition 7.15 it follows $\mathrm{Val}_{\mathcal{M}} = 1$ and hence the result. □

Proposition 7.33. *Let $\mathcal{M}$ be a $\sharp$-acyclic POMDP and δ_0 be an initial distribution. Assume that* $\mathrm{Val}(\mathcal{M}) = 1$ *then, verifier has a winning strategy.*

The proof follows from Lemma 7.36.

In order to prove Lemma 7.36, we need the two following tool lemmata.

Lemma 7.34 (Shifting lemma)**.** *Let $f : Q^\omega \qquad 0, 1$ be the indicator function of a measurable event, $\delta \quad \Delta(Q)$ an initial distribution, and σ a strategy. Then*

$$\mathbb{P}^\sigma_\delta(f(S_1, S_2, \cdots) = 1 \quad A_0 = a \quad O_1 = o) = \mathbb{P}^{\sigma'}_{\delta'}(f(S_0, S_1, \cdots) = 1)\ ,$$

where $(q \quad Q)$, $\delta\ (q) = \mathbb{P}^\sigma_\delta(S_1 = q \quad A_0 = a \quad O_1 = o)$, $\sigma\ (o_2 o_3 \cdots o_n) = \sigma(o o_2 o_3 \cdots o_n)$.

Proof. Using basic definitions, this holds when f is the indicator function of a union of events over S^ω, and the class of events that satisfy this property is a monotone class. □

Lemma 7.35. *Assume that $\mathcal{M}$ is $\sharp$-acyclic and that $\mathcal{O}$ is a singleton, then $\mathcal{M}$ is a $\sharp$-acyclic probabilistic automaton.*

Proof. The result follows for the fact that for any action a and support S, if the set $\mathcal{O}$ is a singleton then $S \cdot a$ is a singleton as well, and thus the knowledge graph coincide with construction of previous chapter, namely the support graph (c.f. Definition 6.5). □

Lemma 7.36. *Let S be a support and $\mathcal{T}$ is a collection of supports. Assume that $\mathcal{T}$ is limit-reachable from S, then either $S \quad \mathcal{T}$ or there exists a collection of supports $\mathcal{R}$ such that*

i) $S \quad \mathcal{R}$,

ii) $\mathcal{R}$ is $\sharp$-reachable from S,

iii) $\mathcal{T}$ is limit-reachable from every support in $\mathcal{R}$.

Proof. If $S \quad \mathcal{T}$, then there is nothing to prove. Assume that $S \quad \mathcal{T}$ and $\mathrm{Obs}^{-1}(o) = S$. Since $\mathcal{T}$ is limit-reachable from S, there exists a sequence of strategies $(\sigma_n)_n \ \mathbb{N}$ such that is a limit-strategy from S to $\mathcal{T}$. For every $n \geq 0$, let

$$d_n = \min_k \left\{ \left[S \quad S \cdot \sigma_n(o^k) \right] \quad \left[S \quad S \cdot \sigma_n(o^k) \quad S \quad S \cdot \sigma_n(o^k)^\sharp \right] \right\}\ ,$$

and let

$$A_S = \left\{ a \quad A \quad S \cdot a^\sharp = \quad S \quad \right\}\ .$$

Fig 7.4 shows the construction and the behavior of the sequence $(d_n)_n \ \mathbb{N}$. According to Proposition 7.23, we know that for every $n \geq 0$ and for every $i < d_n$ we have:

$$S \cdot \sigma_n(o^i)^\sharp = \quad S \quad .$$

Consider now the sequence $(d_n)_n \ \mathbb{N}$ of integers induced by the limit-strategy $(\sigma_n)_n \ \mathbb{N}$ and denote $(u_n)_n \ \mathbb{N}$ the sequence of words such that $u_n = \sigma_n(o) \ldots \sigma_n(o^{d_n - 1})$.

Let us show that it is not possible that for infinitely many n, $d_n = \quad$. Assume towards a contradiction that there exists infinitely many n such that $d_n = \quad$. For every n consider the infinite sequence of distributions $(\delta_S \cdot \sigma(o^i))_i \ \mathbb{N}$. Since $\Delta(Q)$ is compact, the sequence $(\delta_S \cdot \sigma_n(o^i))_i \ \mathbb{N}$ converges to some limit δ_n. Now because $(\sigma_n)_n \ \mathbb{N}$ is a limit-strategy from S to $\mathcal{T}$ and $S \quad \mathcal{T}$;

there exists infinitely many n such that $\mathrm{Supp}(\delta_n) = S$. On the other hand, $\mathcal{M}[S, A_S]$ induces a probabilistic automaton (c.f. Chapter 6). Moreover, according to Lemma 7.35, the probabilistic automaton induced by $\mathcal{M}[S, A_S]$ is $\sharp$-acyclic (c.f. Section 6.5), thus according to the flooding lemma (c.f. Lemma 6.36) the unique limit-reachable support from S is S, thus $\mathrm{Supp}(\delta_n) = S$ contradicts the flooding lemma. Since d_n is infinite for only finitely many n, we assume without loss of generalities that for every n, $d_n <$ and that the sequence $(d_n)_{n \quad \mathbb{N}}$ is increasing. Again using the flooding lemma on the probabilistic automaton $\mathcal{M}[S, A_S]$, we get that the support of the limit of the sequence $(\delta_S \cdot u_n)_{n \quad \mathbb{N}}$ where $u_n = \sigma_n(o) \dots \sigma_n(o^{d_n - 1})$ is exactly S. Since A is finite assume without loss of generalities that $\sigma_n(o^{d_n})$ is constant equal to $a \quad A \quad A^\sharp$. if $S \quad S \cdot \sigma_n(o^{d_n}$, then let $\mathcal{R} = S \cdot a$ and if $S \quad S \cdot \sigma_n(o^{d_n} \quad S \quad S \cdot \sigma_n(o^{d_n})^\sharp$ then let $\mathcal{R} = S \cdot a^\sharp$. *i)* holds because a does not ($\sharp$-)stabilize S. *ii)* holds because the strategy that plays a from S in $\mathcal{G}_\mathcal{M}$ allows Verifier to reach a support in $\mathcal{R}$. Let us show that *iii)* holds.

We show that for every $R \quad \mathcal{R}$, the collection $\mathcal{T}$ is limit-reachable from R. $m \quad \mathbb{N}, \varepsilon > 0, T \quad \mathcal{T}, \sigma_n$, and δ_S we denote

$$A_m(\varepsilon, T, \sigma_n, \delta_S) = \mathbb{P}^{\sigma_n}_{\delta_S}(S_m \quad T \quad O_0 A_0 O_1 A_1 \cdots O_m) \geq 1 - \varepsilon \ .$$

By Definition 7.13 and since $S \quad \mathcal{T}$ we can write

$$\varepsilon > 0, \quad n \quad \mathbb{N},\ \mathbb{P}^{\sigma_n}_{\delta_S}(\quad m \geq d_n, \quad T \quad \mathcal{T},\ A_m(\varepsilon, T, \sigma_n, \delta_S)) \geq 1 - \varepsilon \ . \tag{7.4}$$

We first show that for every $R \quad \mathcal{R}$ we have:

$$\mathbb{P}^{\sigma_n}_{\delta_S}\left(\bigcup_{m>0}\bigcup_{T \quad \mathcal{T}} A_m(\varepsilon, T, \sigma_n, \delta_S) \quad \left(\bigwedge_{i=0}^{d_n-1}(O_i = o \quad A_i = a)\right) \quad \left(O_{d_n} = \mathrm{Obs}^{-1}(R)\right)\right) \underset{n}{\overline{\qquad}} \quad 1 \ . \tag{7.5}$$

Let $\varepsilon > 0$ and $n \quad \mathbb{N}$ such that Equation (7.4) holds. Denote P_n the left-hand side of Equation (7.4), $P_n(R)$ the left-hand side of Equation (7.5), and $\alpha(R)$ the quantity

$$\mathbb{P}^{\sigma_n}_{\delta_S}\left(\left(\bigwedge_{i=0}^{d_n-1}(O_i = o \quad A_i = a)\right) \quad \left(O_{d_n} = \mathrm{Obs}^{-1}(R)\right)\right) \ ,$$

for some $R \quad \mathcal{R}$. Then we have by Equation (7.4):

$$P_n = \sum_{R \quad \mathcal{R}} \alpha(R) P_n(R) \ .$$

Since $\mathcal{R} = S \cdot a$ it follows that for every $R \quad \mathcal{R}$, we have $\alpha(R) > 0$ and $\sum_{R \quad \mathcal{R}} \alpha(R) = 1$. Thus Equation (7.4) yields the following equation

$$P_n(R) \underset{n}{\overline{\qquad}} \quad 1 \ ,$$

and Equation (7.5) follows.

Applying the shifting lemma to Equation (7.5) we obtain for every $o_2 \cdots o_m \quad \mathcal{O}^*$ and every $R \quad \mathcal{R}$:

$$\mathbb{P}^{\sigma'_n}_{\delta'_R}\left(\bigcup_{m \geq d_n}\bigcup_{T \quad \mathcal{T}} \mathbb{P}^{\sigma'_n}_{\delta'_R}(S_{m-1} \quad T \quad O_0 = o_{d_n}, A_1 = a_{d_n+1}, \cdots, O_{m-1} = o_m) \geq 1 - \varepsilon\right) \geq 1 - \varepsilon \ , \tag{7.6}$$

where

$$q \quad Q,\ \delta_R(q) = \mathbb{P}^{\sigma'_n}_{\delta_S}\left(S_1 = q \quad \left(\bigwedge_{i=0}^{d_n-1}(O_i = o \quad A_i = a)\right) \quad \left(O_{d_n} = \mathrm{Obs}^{-1}(R)\right)\right) \ ,$$

and

$$h \quad \mathcal{O}^*\mathcal{O},\ \sigma_n(h) = \sigma_n(o^{d_n-1}\mathrm{Obs}^{-1}(R)h) \ .$$

According to Proposition 7.17, it follows that

$$\mathbb{P}^{\sigma'_n}_{\delta_R}\left(\bigcup_{m\geq d_n}\bigcup_{T \quad \mathcal{T}}\mathbb{P}^{\sigma'_n}_{\delta_R}(S_{m-1} \quad T \quad O_0 = o_{d_n}, A_1 = a_{d_n+1}, \cdots, O_{m-1} = o_m) \geq 1-\varepsilon\right) \underset{n}{\rule{1cm}{0.4pt}} \quad 1 \ . \tag{7.7}$$

This last equation terminates the proof *iii*) and yields Lemma 7.36. □

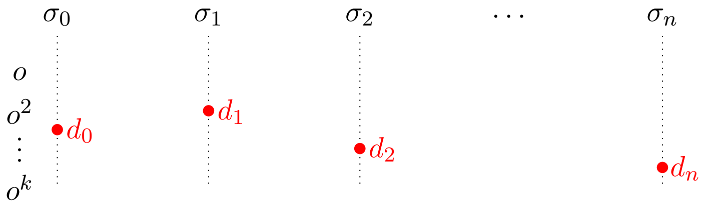


Figure 7.4: Construction of the sequence $(d_n)_n \ \mathbb{N}$

Proof of Proposition 7.33. Let $\mathcal{M}$ be a $\sharp$-acyclic POMDP and δ_0 be an initial distribution. Assume that $\mathrm{Val}(\delta_0) = 1$ then by Proposition 7.15 we know that there exists a limit-strategy $(\sigma_n)_n \ \mathbb{N}$ from the support $\mathrm{Supp}(\delta_0)$ to $\mathcal{T}$ a collection of supports that consists of subsets of T. Thanks to Lemma 7.36, we know that from the support of the initial distribution, one can define a collection of supports $\mathcal{R}_0$ and an action a such that $\mathcal{R} = \mathrm{Supp}(\delta_0)\cdot a$ and such that the items *i*), *ii*), and *iii*) of Lemma 7.36 hold.

If $\mathcal{R} \subseteq \mathcal{T}$ then the proof is done since the strategy that consists in playing the action a from $\mathrm{Supp}(\delta_0)$ is winning for verifier.

If not we construct inductively a DAG whose nodes are the nonempty supports, the edges are labelled by the actions, and the leafs are supports in $\mathcal{T}$ the following way:

- the root is labelled by $\mathrm{Supp}(\delta_0)$,
- for each node labelled with a nonempty support R either $R \quad \mathcal{T}$ and R is a leaf or there exists an action $a \quad A \quad A^\sharp$ such that the sons of R are the elements of the collection $R\cdot a$ where the $R\cdot a$ is the collection constructed by Lemma 7.36 and the edges $(R, R\ )$ for every $R\ $ in $R\cdot a$ is labelled by a.

Now because $\mathcal{M}$ is $\sharp$-acyclic , we know that this construction terminates in at most $2^{\ Q\ }$ steps and that the DAG obtained is the unfolding of a winning strategy for verifier. □

Proposition 7.31 and Proposition 7.33 lead the following theorem:

Theorem 7.37. *Given a $\sharp$-acyclic POMDP $\mathcal{M}$ and an initial distribution δ_0. Verifier has a winning strategy in the knowledge game $\mathcal{G}_{\mathcal{M}}$ if and only if* $\mathrm{Val}(\mathcal{M}) = 1$.

Theorem 7.28 follows directly from Theorem 7.37 and from the fact that deciding the winner in a perfect information reachability game is decidable.

Proposition 7.38 (Upper bound)**.** *The value 1 problem is* EXPTIME.

Proof. It is consequence of the fact that deciding the value 1 problem reduces to solving a reachability game of size exponential in the description of the POMDP. □

7.5 Conclusion

In this chapter we extended the decidability result obtained for probabilistic automata to the framework of partial observation Markov decision processes. We defined the class of $\sharp$-acyclic POMDP that extends $\sharp$-acyclic automata to the case of partial observation. In order to decide the value 1 for this new class we generalize the operation of iteration and use perfect information reachability games to abstract the asymptotic behavior of the automaton. The decision procedure obtained runs in EXPTIME whereas the value 1 problem is decidable for the class of $\sharp$-acyclic automata is decidable in PSPACE. We do not know whether our decision procedure for $\sharp$-acyclic POMDP can run in PSPACE since it needs to remember collection of support. Providing a lower bound for our the $\sharp$-acyclic POMDP is one research direction. We also believe that the class of $\sharp$-acyclic POMDP can be extended, this is one of our next goal. Finally, we highlight the fact $\sharp$-acyclic POMDP depends qualitatively on the transition probability, we believe that this class can be extend in a way such that the decision procedure depends quantitatively on the transition probabilities.

Part IV

Conclusion and References

Chapter 8

Conclusion

Contents

8.1 Summary

In this thesis we studied and designed algorithms for solving the value 1 problem in two different but yet related frame work.

In the first part we studied perfect information games equipped with boolean combination of objectives. We first designed algorithm for parity and positive-average for Markov decision processes and for stochastic games. Our algorithm construct the almost-sure region, this gives the set of states with value 1 as we know by [GH10] that they coincide for stochastic games with tail objectives. Second we designed an algorithm to solve boolean combination of positive-average objectives.

In the second part we studied reachability objectives for partial information games. This model is known to be undecidable for many problems. Indeed the value problem is known to be undecidable since the work of Paz [Paz71]. We solve the value 1 problem which was open since the work of Bertoni [Ber74, BMT77]. Unfortunately this problem turned out to be undecidable. In order to overcome this undecidability, we introduced sub classes of games for which the problem is decidable.

8.2 Discussion and Open Problems

In this section we discuss some of the results obtained in this thesis and future research directions.

8.2.1 Markov Decision Process

Regarding Markov decision processes, we started first by designing an algorithm that computes the almost-sure region for the objective Par $\mathrm{Avg}_{>0}$ in polynomial time, then we show that the almost-sure strategy can be with finite memory. A direct consequence of the existence of finite memory almost-sure strategies is that the same approach holds true for Par $\underline{\mathrm{Avg}}_{>0}$ and that it can be effectively used for control synthesis.

The other result we obtained concerns boolean combination of objectives. We give a constructions for combination of positive-averages and we leave the case of boolean combination with parity and positive-average open.

This open problem stated is solved in the special case of parity and positive-average with $\limsup$ semantics. The next step is to try to solve this problem in the case of parity and $\liminf$ semantics, we believe that the solution of this problem has to go through a generalization of the equations of Theorem 4.18 in order to satisfy the parity condition but we couldn t achieve this in this thesis.

8.2.2 Stochastic Games

In the case of stochastic games, we extended part of the results obtained for Markov decision processes to the setting of stochastic games. Namely we give an algorithm that computes the almost-sure region for the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ in NP. This result raises some open questions:

- is there an algorithm that computes the value of state for the objective $\mathrm{Par} \quad \mathrm{Avg}_{>0}$ lies in $\mathrm{NP} \setminus \mathrm{CoNP}$?
- can the memory requirement for the almost-sure strategies be bounded from above?

We believe that the answer for both questions is yes and we are currently investigating those problems.

8.2.3 Probabilistic Automata and Beyond

First, the decidability result obtained for POMDP shows that studying probabilistic automata in order to understand the framework of partial information games was a fruitful approach. Indeed, the techniques used to define the class of $\sharp$-acyclic POMDP were generalization of the one used for probabilistic automata.

Second, the study of probabilistic automata reveled some surprising results. For instance, the result concerning automata with one probabilistic transition shows that the border of undecidability is easily reached.

We believe that the following directions are interesting:

- Investigate interesting classes of automata for which the emptiness problem is decidable.
- Investigate interesting classes for which the value 1 is decidable and depends quantitatively on the transition probabilities.

Bibliography

[BBC$^+$11] Tomás Brázdil, Václav Brozek, Krishnendu Chatterjee, Vojtech Forejt, and Antonín Kucera. Two views on multiple mean-payoff objectives in markov decision processes. In *LICS*, pages 33–42, 2011. (Cited on pages 5 and 52.)

[BBE10a] Tomás Brázdil, Václav Brozek, and Kousha Etessami. One-counter stochastic games. In *FSTTCS*, pages 108–119, 2010. (Cited on pages 18 and 57.)

[BBE$^+$10b] Tomás Brázdil, Václav Brozek, Kousha Etessami, Antonín Kucera, and Dominik Wojtczak. One-counter Markov decision processes. In *SODA*, pages 863–874, 2010. (Cited on page 32.)

[BBG08] Christel Baier, Nathalie Bertrand, and Marcus Größer. On decision problems for probabilistic büchi automata. In *FoSSaCS*, pages 287–301, 2008. (Cited on page 87.)

[BCG05] Christel Baier, Frank Ciesinski, and Marcus Größer. Probmela and verification of markov decision processes. *SIGMETRICS Performance Evaluation Review*, 32(4):22–27, 2005. (Cited on page 21.)

[Ber74] A. Bertoni. The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In *Proc. of the 4th GI Jahrestagung*, volume 26 of *LNCS*, pages 107–112. Springer, 1974. (Cited on pages 6, 78, 81, 86 and 121.)

[BGG09] Nathalie Bertrand, Blaise Genest, and Hugo Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *LICS*, pages 319–328, 2009. (Cited on page 105.)

[BL69] J. Richard Büchi and Lawrence H. Landweber. Definability in the monadic second-order theory of successor. *J. Symb. Log.*, 34(2):166–170, 1969. (Cited on page 3.)

[BMT77] Alberto Bertoni, Giancarlo Mauri, and Mauro Torelli. Some recursive unsolvable problems relating to isolated cutpoints in probabilistic automata. In *Proceedings of the Fourth Colloquium on Automata, Languages and Programming*, pages 87–94, London, UK, 1977. Springer-Verlag. (Cited on pages 6, 78, 81, 86 and 121.)

[Bor21] Émile Borel. La théorie du jeu et les équations intégrales à noyau symétrique. *Comptes Rendus de l'Académie des Sciences*, 173:1304–1308, 1921. (Cited on page 1.)

[BS78] Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case.* Academic Press, Inc., Orlando, FL, USA, 1978. (Cited on page 23.)

[CD10] Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In *ICALP (2)*, pages 599–610, 2010. (Cited on page 37.)

[CD11] Krishnendu Chatterjee and Laurent Doyen. Energy and mean-payoff parity markov decision processes. In *MFCS*, pages 206–218, 2011. (Cited on pages 5 and 38.)

[CDGH10] Krishnendu Chatterjee, Laurent Doyen, Hugo Gimbert, and Thomas A. Henzinger. Randomness for free. In *MFCS*, pages 246–257, 2010. (Cited on page 105.)

[CDHR10] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Generalized mean-payoff and energy games. *CoRR*, abs/1007.1669, 2010. (Cited on page 37.)

[Cha07] Krishnendu Chatterjee. Concurrent games with tail objectives. *Theor. Comput. Sci.*, 388(1-3):181–198, 2007. (Cited on page 27.)

[CHJ05] Krishnendu Chatterjee, Tom Henzinger, and Marcin Jurdzinski. Mean-payoff parity games. In *LICS 05*, June 2005. (Cited on pages 37, 38 and 61.)

[CJH03] Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger. Simple stochastic parity games. In *CSL*, pages 100–113, 2003. (Cited on pages 31 and 42.)

[CJH04] Krishnendu Chatterjee, Marcin Jurdzi ski, and Thomas A. Henzinger. Quantitative stochastic parity games. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA 04, pages 121–130, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics. (Cited on pages 30 and 103.)

[CL89] Anne Condon and Richard J. Lipton. On the complexity of space bounded interactive proofs (extended abstract). In *Foundations of Computer Science*, pages 462–467, 1989. (Cited on page 78.)

[CL08] Thomas Colcombet and Christof Löding. The non-deterministic mostowski hierarchy and distance-parity automata. In *ICALP (2)*, pages 398–409, 2008. (Cited on page 2.)

[Cla08] Edmund M. Clarke. The birth of model checking. In *25 Years of Model Checking*, pages 1–26, 2008. (Cited on page 3.)

[CMR07] Corinna Cortes, Mehryar Mohri, and Ashish Rastogi. L_p distance and equivalence of probabilistic automata. *International Journal of Foundations of Computer Science*, 18(4):761–779, 2007. (Cited on page 78.)

[CMRR08] Corinna Cortes, Mehryar Mohri, Ashish Rastogi, and Michael Riley. On the computation of the relative entropy of probabilistic automata. *International Journal of Foundations of Computer Science*, 19(1):219–242, 2008. (Cited on page 78.)

[Con92] Anne Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992. (Cited on pages 28 and 61.)

[CSV09] Rohit Chadha, A. Prasad Sistla, and Mahesh Viswanathan. Power of randomization in automata on infinite strings. In *International Conference on Concurrency Theory*, pages 229–243, 2009. (Cited on page 102.)

[CY90] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *ICALP'90*, volume 443 of *LNCS*, pages 336–349. Springer, 1990. (Cited on page 21.)

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995. (Cited on pages 30 and 103.)

[dAH00] Luca de Alfaro and Thomas A. Henzinger. Concurrent omega-regular games. In *LICS*, pages 141–154, 2000. (Cited on pages 24, 31 and 42.)

[EJ91] E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS*, pages 368–377, 1991. (Cited on page 2.)

[FGO11] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Pushing undecidability of the isolation problem for probabilistic automata. April 2011. (Cited on page 88.)

[FGO12] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. In *LICS*, pages 295–304, 2012. (Cited on pages 6 and 102.)

[GH82] Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65, 1982. (Cited on page 2.)

[GH10] Hugo Gimbert and Florian Horn. Solving Simple Stochastic Tail Games. page 1000, 01 2010. (Cited on pages 27, 63, 64 and 121.)

[Gil57] Dean Gillette. Stochastic games with zero stop probability. *Contributions to the Theory of Games*, 3:179–187, 1957. (Cited on page 33.)

[Gim07] Hugo Gimbert. Pure stationary optimal strategies in Markov decision processes. In *STACS*, pages 200–211, 2007. (Cited on page 44.)

[Gim09] Hugo Gimbert. Randomized Strategies are Useless in Markov Decision Processes. July 2009. (Cited on page 105.)

[GO10] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *International Colloquium on Automata, Languages and Programming*, pages 527–538, 2010. (Cited on pages 6, 78 and 103.)

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002. (Cited on page 29.)

[GZ06] Hugo Gimbert and Wieslaw Zielonka. Deterministic priority mean-payoff games as limits of discounted games. In *ICALP (2)*, pages 312–323, 2006. (Cited on page 37.)

[GZ07a] Hugo Gimbert and Wieslaw Zielonka. Limits of multi-discounted markov decision processes. In *LICS*, pages 89–98, 2007. (Cited on page 37.)

[GZ07b] Hugo Gimbert and Wieslaw Zielonka. Perfect information stochastic priority games. In *ICALP*, pages 850–861, 2007. (Cited on page 37.)

[KEY07] M. Vardi K. Etessami, M. Kwiatkowska and M. Yannakakis. Multi-objective model checking of markov decision processes. In *Proc of TACAS'07*, volume 4424, pages 50–65, 2007. (Cited on page 21.)

[KNP07] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, 2007. (Cited on page 21.)

[Koz77] Dexter Kozen. Lower bounds for natural proofs systems. In *Proc. of 18th Symp. Foundations of Comp Sci.*, pages 254–266, 1977. (Cited on pages 81 and 100.)

[LJ64] C. E. Lemke and Jr. Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423, 1964. (Cited on page 2.)

[LL69] T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoffs. *SIAM Review*, 11:604 – 607, 1969. (Cited on page 33.)

[Mar75] D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975. (Cited on page 63.)

[Mar98] Donald A. Martin. The determinacy of blackwell games. *J. Symb. Log.*, 63(4):1565–1581, 1998. (Cited on page 63.)

[MHC03a] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147:5–34, 2003. (Cited on pages 6 and 81.)

[MHC03b] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003. (Cited on pages 78 and 103.)

[Mor94] J.D. Morrow. *Game Theory for Political Scientists.* Princeton University Press, 1994. (Cited on page 1.)

[MS] Maitra and Sudderth. Stochstic games with borel payoffs. *A Maitra, W Sudderth - Stochastic Games and Applications, NATO ..., 2003 - ratio.huji.ac.il.* (Cited on page 63.)

[MS85] David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985. (Cited on page 2.)

[MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of rabin, mcnaughton and safra. *Theor. Comput. Sci.*, 141(1&2):69–107, 1995. (Cited on page 2.)

[Mye91] Roger B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, 1991. (Cited on page 1.)

[Nas50] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, January 1950. (Cited on page 1.)

[NM44] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944. (Cited on page 1.)

[Nor97] J. R. Norris. *Markov chains.* Cambridge University Press, 1997. (Cited on page 17.)

[OW96] Martin J. Osborne and Paul S. Walker. A note on the early history of the theory of strategic games from waldegrave to borel by robert w. dimand and mary ann dimand. *History of Political Economy*, 28(1):81–82, Spring 1996. (Cited on page 1.)

[Pap93] Christos H. Papadimitriou. *Computational Complexity.* Addison Wesley, November 1993. (Cited on page 81.)

[Paz71] Azaria Paz. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press, Inc., Orlando, FL, USA, 1971. (Cited on pages 6, 78, 81, 83, 103 and 121.)

[Put94] Martin L. Putterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994. (Cited on pages 18, 31, 33 and 71.)

[QS82] Jean-Pierre Queille and Joseph Sifakis. A temporal logic to deal with fairness in transition systems. In *FOCS*, pages 217–225, 1982. (Cited on page 3.)

[Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. (Cited on pages 30, 77, 80 and 86.)

[Rab69] Michael O. Rabin. Decidability of Second Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969. (Cited on page 2.)

[Sav70] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970. (Cited on page 99.)

[Sch61] Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4, 1961. (Cited on page 78.)

[Sha53] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. (Cited on page 2.)

[Sim90] Imre Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990. (Cited on pages 2 and 102.)

[Smi82] J.M. Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982. (Cited on page 1.)

[TAHW09] Maria Mateescu Thomas A. Henzinger and Verena Wolf. Sliding-window abstraction for infinite markov chains. In *Proc. of CAV'09*, volume 5643, pages 337–352, 2009. (Cited on page 21.)

[Tze92] Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM Journal on Computing*, 21(2):216–227, 1992. (Cited on page 78.)

[Vel11] Yaron Velner. The complexity of mean-payoff automaton expression. *CoRR*, abs/1106.3054, 2011. (Cited on page 52.)

[Zer13] Ernst Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the Fifth International Congress Mathematics*, pages 501–504, Cam -bridge, 1913. Cambridge University Press. (Cited on page 1.)

[Zie98] Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998. (Cited on page 2.)

[Zie04] Wieslaw Zielonka. Perfect-information stochastic parity games. In *FoSSaCS*, pages 499–513, 2004. (Cited on page 38.)