On Non-Classical Stochastic Shortest Path Problems

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

vorgelegt an der

Technische Universität Dresden
Fakultät für Informatik

eingereicht von

Jakob Piribauer, M. Sc.
geboren am 6. August 1994 in Berlin

begutachtet von

Prof. Dr. Christel Baier
Technische Universität Dresden

Prof. Dr. Jan Křetínský
Technische Universität München

verteidigt am

15. Juni 2021

ABSTRACT

The stochastic shortest path problem lies at the heart of many questions in the formal verification of probabilistic systems. It asks to find a scheduler resolving the non-deterministic choices in a weighted Markov decision process (MDP) that minimizes or maximizes the expected accumulated weight before a goal state is reached. In the classical setting, it is required that the scheduler ensures that a goal state is reached almost surely. For the analysis of systems without guarantees on the occurrence of an event of interest (reaching a goal state), however, schedulers that miss the goal with positive probability are of interest as well. We study two non-classical variants of the stochastic shortest path problem that drop the restriction that the goal has to be reached almost surely. These variants ask for the optimal *partial expectation*, obtained by assigning weight 0 to paths not reaching the goal, and the optimal *conditional expectation* under the condition that the goal is reached, respectively. Both variants have only been studied in structures with non-negative weights.

We prove that the decision versions of these non-classical stochastic shortest path problems in MDPs with arbitrary integer weights are at least as hard as the Positivity problem for linear recurrence sequences. This Positivity problem is an outstanding open number-theoretic problem, closely related to the famous Skolem problem. A decidability result for the Positivity problem would imply a major breakthrough in analytic number theory. The proof technique we develop can be applied to a series of further problems. In this way, we obtain Positivity-hardness results for problems addressing the termination of one-counter MDPs, the satisfaction of energy objectives, the satisfaction of cost constraints and the computation of quantiles, the conditional value-at-risk – an important risk measure – for accumulated weights, and the model-checking problem of frequency-LTL.

Despite these Positivity-hardness results, we show that the optimal values for the non-classical stochastic shortest path problems can be achieved by weight-based deterministic schedulers and that the optimal values can be approximated in exponential time. In MDPs with non-negative weights, it is known that optimal partial and conditional expectations can be computed in exponential time. These results rely on the existence of a saturation point, a bound on the accumulated weight above which optimal schedulers can behave memorylessly. We improve the result for partial expectations by showing that the least possible saturation point can be computed efficiently. Further, we show that a simple saturation point also allows us to compute the optimal conditional value-at-risk for the accumulated weight in MDPs with non-negative weights.

Moreover, we introduce the notions of long-run probability and long-run expectation addressing the long-run behavior of a system. These notions quantify the long-run average probability that a path property is satisfied on a suffix of a run and the long-run average expected amount of weight accumulated before the next visit to a target state, respectively. We establish considerable similarities of the corresponding optimization problems with non-classical stochastic shortest path problems. On the one hand, we show that the threshold problem for optimal long-run probabilities of regular co-safety properties is Positivity-hard via the Positivity-hardness of non-classical stochastic shortest path problems. On the other hand, we show that optimal long-run expectations in MDPs with arbitrary integer weights and long-run probabilities of constrained reachability properties $(a \,\mathrm{U}\, b)$ can be computed in exponential time using the existence of a saturation point.

CONTENTS

CHAPTER ONE

INTRODUCTION

Modern software and hardware systems have reached a level of complexity that makes it outright impossible for a human to analyze and understand their behavior without auxiliary tools. To gain trust that a system behaves as intended, testing the system in a variety of environments suggests itself and is an integral part of any systems development process. Testing can help to detect many errors and, if the system works correctly on a vast variety of inputs, this can be sufficient assurance for the correctness in many application areas. Nevertheless, there are usually infinitely many possible executions of a system. So, testing cannot be exhaustive and testing alone cannot provide a guarantee that the system will behave correctly in all situations. In many safety-critical application areas, computer systems play such an important role that more rigid guarantees on the correctness of a system are of utter interest.

A great success story providing such guarantees was initiated in the late 1970s and early 1980s. In his seminal work [Pnu77], Pnueli suggested the use of temporal logics to reason about program correctness. Just a few years later, in the early 1980s, *model checking* was invented independently by Clarke and Emerson [CE81] and by Queille and Sifakis [QS82]. This technique takes a mathematical system model M with a set of states and transitions between these states and a formal specification φ of the correct executions of the system as input and answers the question whether all possible executions of M satisfies the specification φ. For the development of this novel verification paradigm, Pnueli and Clarke, Emerson, and Sifakis received the Turing award in 1996 and 2007, respectively. Model checking today constitutes one of the most important approaches in the area of formal verification that aims to provide rigorous mathematical guarantees of the correctness of a system.

The inherent properties of the system to be analyzed determine the mathematical model to use. On the one hand, a system might exhibit *non-deterministic* behavior. In concurrent systems for example, the order and the precise timing in which computations take place cannot be predicted exactly. Hence, each system state might have multiple possible successor states depending on the order of the concurrently executed computations. In order to verify that the system behaves correctly, it is required to verify that all possible resolutions of these non-deterministic choices of successors lead to a correct computation with respect to the given specification. Other reasons to model a system with non-deterministic choices include user interactions or the use of the system in unknown environments. The main model for purely non-deterministic systems is provided by *transition systems*.

On the other hand, it can be reasonable to assume a *probabilistic* behavior of a system. In probabilistic programs, randomization is explicitly employed and leads to a transition structure that chooses successors according to a given probability distribution. Also quantum processes can lead to precisely known probabilistic behavior. For other systems, there might be sufficiently much data – for example on the failure of hardware components, on the message loss in a lossy channel, or on the behavior of the environment – to assume a probability distribution over possible successor states. Purely probabilistic systems can be modeled by *Markov chains* in which the successors of each state are chosen according to a specified probability distribution in each step. *Markov decision processes* (MDPs) combine non-deterministic and probabilistic behavior. In each state, an action can be chosen non-deterministically from a set of enabled actions and afterwards the successor is chosen according to a probability distribution associated to the state-action pair.

The necessity to verify systems that exhibit probabilistic behavior lead to the development of *probabilistic model checking* [Var85, VW86, CY95] soon after the invention of model checking. The probabilistic model-checking problem does not address the satisfaction along all possible executions anymore, but rather asks for the probability that an execution satisfies a specification. In the presence of non-deterministic choices, the problem turns into an optimization problem asking for the minimal or maximal possible satisfaction probability when ranging over all possible resolutions of the non-deterministic choices. The specifications are usually given in a (non-probabilistic) temporal logic, such as linear temporal logic (LTL) or computation tree logic (CTL), or as an automaton. Temporal logics can further be extended by operators expressing that the (optimal) satisfaction probability of formulas satisfies an inequality constraint such as in probabilistic computation tree logic (PCTL) [HJ94].

1.1 Stochastic shortest path problems

Besides the question whether a system execution satisfies a certain specification that can either hold or not on an execution, quantitative aspects play an important role when checking that a system behaves as intended. These aspects are for example the termination time, the utility achieved, e.g., the number of successfully completed tasks, the costs of an execution, the energy or resource consumption, and many quantities more that are accumulated during an execution. We model such quantities with weight functions that assign positive and negative weights to transitions in the model. In MDPs, once a resolution of the non-deterministic choices is fixed by specifying a *scheduler*, we obtain a random variable assigning the accumulated weight to runs in the model. The expected value of this random variable is subject to typical verification questions like "What is the worst-case expected termination time of the probabilistic program?". Such questions lead us to the *stochastic shortest path problem*: The problem asks maximize or minimize the expected accumulated weight before reaching a target state in a weighted finite-state MDP. For the expected value of the accumulated weight before reaching a goal state to be well-defined, it is necessary that a goal state is reached almost surely. The problem generalizes the well-known shortest path problem on weighted graphs. An early formulation of the problem can be found in [EZ62] and the problem has subsequently also been studied under the name *first passage problem* [Der70, Whi83]. It is known that this *classical* stochastic shortest path problem is solvable in polynomial time [BT91, dA99, BBD$^+$18].

Example 1.1. Consider the model of a probabilistic program depicted in Figure 1.1. The weight function *wgt* denotes the time required for each transition. Depending on a user input, the program moves to state s or t. In both of these states, the successor is chosen randomly with probability 1/2 for both possible successors. After the probabilistic transition, the system will either terminate after some more time or another user input is requested, leading to state s or t again. If we want to provide a guarantee on the worst- or best-case expected termination time among all possible sequences of user inputs, we have to find a scheduler making decisions in the state "user input" that maximizes or minimizes the expected time until termination. For the scheduler that always moves to state s, we compute the expected termination time as follows: 3 time units are required to reach s. The loop back to "user input" is taken once in expectation leading to an expected time of 3 time units. To see this, note that the loop is taken a first time with probability 1/2, a second time with probability 1/4, and so on. Finally, if the loop is not taken anymore, it takes 6 time units until termination. So, the expected termination time is 12 time units. Analogously, the expected termination time when always moving to t is 13. It is known that the maximal or minimal value in a classical stochastic shortest path problem is obtained by a *memoryless* scheduler that always chooses the same action in each state [BT91, dA99, BBD$^+$18]. So, in fact we can now guarantee that the expected termination time of the example probabilistic program lies between 12 and 13 for all (infinitely many) possible sequences of user inputs. ◁

In the MDP literature, a variety of objectives addressing the expected value of accumulated weights has been studied. Besides the classical stochastic shortest path prob-

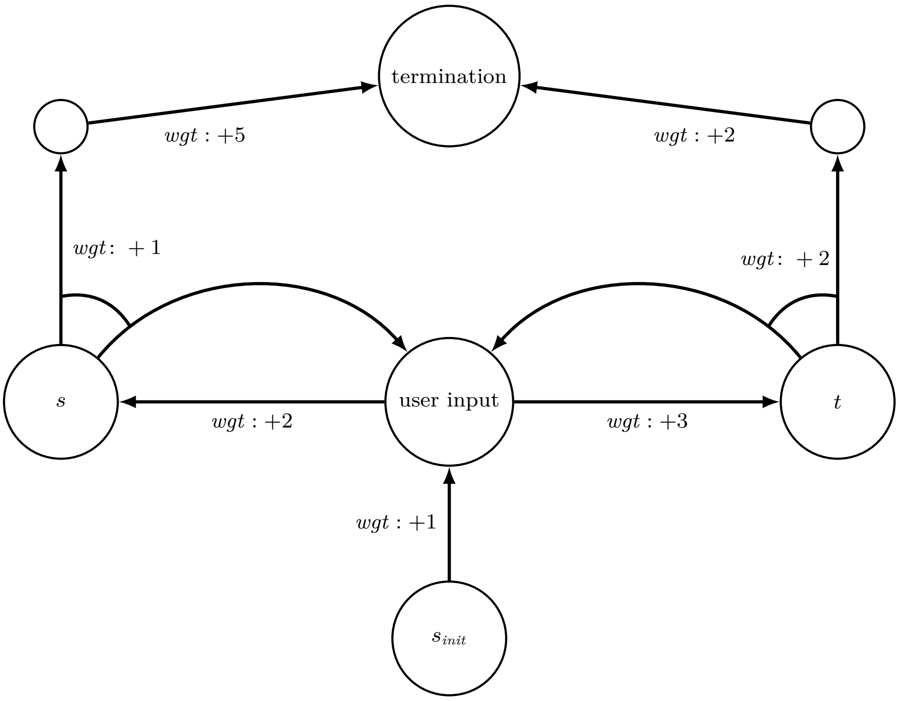


Figure 1.1: Example of a stochastic shortest path problem. The two possible transitions from state s and t are taken with probability 1/2 each.

lem, other objectives address the expected total weight, discounted sums of accumulated weights, mean payoffs, and the accumulation of weights during a finite time horizon (see, e.g., [Put94, Kal11]). To integrate problems addressing expected values of accumulated weights, expectation operators have been integrated into temporal logics as in probabilistic reward computation tree logic [AHK03] and probabilistic alternating-time logic with rewards [CFK$^+$13a]. Furthermore, the optimizations of expected accumulated weights in various forms are implemented in the PRISM [KNP11] and the STORM [DJKV17] model checkers including the classical stochastic shortest path problem.

Non-classical stochastic shortest path problems. The stochastic shortest path problem as just presented has one major restriction: It requires the target to be reached with probability 1. The limitation to schedulers reaching the goal almost surely, however, is often too restrictive. First, there are models that have no such scheduler. Second, even if such schedulers exist, the expectation of the accumulated weight of schedulers missing the goal with a positive probability might be of interest as well. Important applications in which this is the case include the semantics of probabilistic programs (see, e.g., [GKM14, KGJ$^+$15, BEFH16, CFG16, OGJ$^+$18]) where no guarantee for almost sure termination can be given. The analysis of program properties at termination time gives rise to stochastic shortest path problems in which the goal (halting configuration) is not reached almost surely. Other examples are the fault-tolerance analysis, e.g., expected costs of repair mechanisms, in selected error scenarios that can appear with some positive, but small probability, or the trade-off analysis with conjunctions of utility and cost constraints that are achievable with positive probability, but not almost surely (see, e.g., [BDK$^+$14]).

This motivates the switch to variants of the stochastic shortest path problem where the requirement to reach the target with probability 1 is relaxed. One option studied in [CFK$^+$13a] and implemented in [CFK$^+$13b] is to seek for a scheduler optimizing the expectation of the random variable that assigns weight 0 to all paths not reaching the goal and the accumulated weight of the shortest prefix reaching the goal to all other paths. We refer to this expectation as *partial expectation*. A second option is to consider the *conditional expectation* of the accumulated weight until reaching the goal under the condition that the goal is reached as done in [BKKW17]. Partial expectations are suitable to describe situations in which some reward or cost is accumulated but only retrieved if a certain goal is met. In particular, partial expectations can be an appropriate replacement for the classical expected weight before reaching the goal if we want to include schedulers which miss the goal with some – possibly very small – positive probability. In contrast to conditional expectations, the resulting scheduler still has an incentive to reach the goal with a high probability, while schedulers maximizing the conditional expectation might reach the goal with a very small positive probability. Conditional expectations can be particularly useful to analyze the costs or utilities achieved in events with smaller probabilities, such as the incurred repair costs in case of an unlikely error scenario. In [CFK$^+$13a] and [BKKW17], partial and conditional expectations, respectively, have been addressed in systems with non-negative weights. The general variants with weight functions that take positive and negative values have not been studied in the literature. In this thesis, we focus on these general variants of these *non-classical* stochastic shortest path problems.

Conditional expectations also play a crucial role in risk management: The *conditional value-at-risk*, also known as expected shortfall or tail loss, is an established risk measure quantifying the expected loss in bad cases [Ury00, AT02]. Given a probability value p, the *value-at-risk* of a random variable X is the worst p-quantile. In other words, it is defined such that an outcome is worse than the value-at-risk with probability p. The conditional value-at-risk is the expectation of X under the condition that the outcome is worse than the value-at-risk. The conditional value-at-risk quantifies where the outliers in the distribution are located by specifying the average of the outcomes above the value-at-risk, i.e., the average of the worst p outcomes. As a pure worst-case analysis is often inappropriate in a probabilistic setting, the conditional value-at-risk constitutes an important tool in risk management to quantify these worst outcomes, going beyond the value-at-risk that does not take unlikely outliers into consideration. For MDPs, the conditional value-at-risk has been studied for mean-payoffs and for weighted reachability where on each run only once a terminal weight is collected when a target state is reached [KM18]. We will consider the conditional value-at-risk for the more general accumulated weight before reaching the goal, i.e. for the classical stochastic shortest path problem.

1.2 Related problems

There is a variety of problems closely related to stochastic shortest path problems. We give a brief overview of several of these problems in the sequel. For the problems addressing the long-run satisfaction of path properties that we address at the end of this section, the connection to stochastic shortest path problems is not quite obvious on first sight. In this thesis, however, we will disclose parallels and show that techniques for the treatment of stochastic shortest path problems can also be applied to notions of long-run satisfaction.

Cost problems and quantiles. Besides the expected value, several further aspects of the distribution of the random variable assigning the accumulated weight can be of interest in the context of verification. Many problems surround decision problems of the form

$$\text{Is } \Pr^{\max}_{\mathcal{M}}(\text{accumulated weight } \leq w) \geq p?$$

That is, these problems ask whether the maximal (or minimal) probability that the accumulated weight stays below a bound w is at least (or at most) p in an MDP $\mathcal{M}$ for given values w and p. For the precise meaning of "accumulated weight $\leq w$", there are several options. It can refer to the total accumulated weight of a run, to the accumulated weight before reaching a goal state, or it requires that the accumulated weight never exceeds w during a run.

On the one hand, one might be interested in maximizing or minimizing the probability that the accumulated weight lies within the given bound w or exceeds the bound when a goal state is reached – a problem addressed in [HK15, HKL17, BBD+18] and called the *cost problem* in [HK15]. On the other hand, quantile queries ask for the minimal weight w such that the weight of a path stays below w with probability at least p for the given value p under some or all schedulers [UB13, BDD+14, RRS17]. Both of these problems have been addressed for MDPs with non-negative weights and are solvable in exponential time in this setting [UB13, HK15]. The decision version of the cost problem with non-negative weights is furthermore PSPACE-hard for a single inequality on the accumulated weight and EXPTIME-complete if a Boolean combination of inequality constraints on the accumulated weight is considered [HK15]. For the setting with arbitrary weights, [BBD+18] provides solutions to the qualitative question whether a constraint on the accumulated weight is satisfied with probability 1 (or > 0). Further, it is known that the quantitative problem is undecidable if multiple objectives with multiple weight functions have to be satisfied simultaneously [RRS17].

One-counter MDPs. The termination problem of one-counter MDPs introduced in [BBE$^+$10] can also be seen as an instance of the question with which probability a constraint on the accumulated weight is satisfied maximally or minimally. In a one-counter MDP, there is a counter that can be increased, decreased or left unchanged in each step. The process typically starts with counter value 1 and terminates if the counter value reaches 0. The termination problem asks for the maximal probability of termination that a scheduler can achieve. While it is decidable whether the maximal termination probability is 1 in polynomial time and in exponential time if termination is required to occur inside a specified set of states [BBE$^+$10], the computation of the optimal value and the quantitative decision problem whether the optimal value exceeds a threshold p are left open in the literature. Also the problem to compute the minimal or maximal expected termination time of a one-counter MDP that terminates almost surely under any scheduler is open. There are, however, approximation algorithms for the optimal termination probability [BBEK11] and for the expected termination time of almost surely terminating one-counter MDPs [BKNW12]. One-counter MDPs can be seen as a special case of recursive MDPs [EY15]. For general recursive MDPs, the qualitative decision problem whether the maximal termination probability is 1 is undecidable while for restricted forms, so-called 1-exit recursive MDPs, the qualitative and also the quantitative problem is decidable in polynomial space [EY15]. One-counter MDPs can be seen as a special case of 1-box recursive MDPs in the terminology of [EY15], a restriction orthogonal to 1-exit recursive MDPs.

Energy objectives. If the accumulated weight that can increase and decrease along a run models a resource like energy, a further natural objective is to keep the accumulated value above 0 at all times instead of trying to reach the value 0 as in one-counter MDPs. This objective is often called an *energy objective*. There has been work on combinations of the energy objective with further objectives such as parity objectives [CD11,MSTW17] and expected mean payoffs [BKN16]. Again, previous work on this objective focused on the possibility to satisfy the objective (or the combination of objectives) almost surely. The quantitative problem whether it is possible to satisfy an energy objective with probability greater than some threshold p is open. Note that the maximal probability to satisfy an energy objective corresponds directly to the minimal termination probability of the corresponding one-counter MDP.

Long-run satisfaction of path properties. Besides encoding quantitative features of a model into a weight-structure, a further branch of research addresses ways to quantify the degree to which a specification is satisfied by a model (see [Hen13] for an overview of the field). This includes work on the robust satisfaction of temporal specifications [KV99, TN16], vacuity and coverage semantics [KV03, CS09, CKV06, KLS08], robustness distances [CHR12], and the more general model-measurement semantics based on automatic distance functions of [HO13].

One line of research in this direction attempts to measure the degree to which a specification is satisfied when evolving over time. This includes, e.g., the work on frequency-LTL [BDL12, FK15, FKK15]. In frequency-LTL, temporal operators are relaxed by frequency constraints. A formula then does not have to hold on all suffixes, but the frequency of suffixes that satisfy a formula – defined as the limit of the fraction of the number of suffixes that satisfy the formula over the number of all suffixes – has to be at least some rational q for example. Alternatively, averaging LTL [BMM14] rather than truth values, assigns quantities to pairs of paths and formula. It is based on a quantitative labeling function for atomic propositions and inductively defines the semantics of temporal operators as the average of the value of φ at the respective positions. A notable similarity of these two quantitative extensions of LTL is the undecidability of the model checking problem of the full logics [BDL12, BMM14]. Decidable fragments of frequency-LTL can be obtained by restricting the nesting of temporal operators or the allowed frequency thresholds [BDL12, FK15, FKK15].

In this thesis, we study a series of optimization problems concerned with the degree of satisfaction of a property in a system in the long-run. In non-probabilistic systems, we address the optimization of *long-run frequencies* defined as in the definition of the quantitative globally operator of frequency-LTL. In probabilistic systems, we extend this notion to *long-run probabilities*, expressing the average probability that a system will satisfy a property when we start to observe it after many steps, i.e., the long-run average satisfaction probability. Finally, long-run expectations express the long-run average amount of weight that will be accumulated in expectation before the next visit to a goal state. This notion can be used to determine, for example, the expected time until the next message is processed when starting to observe a system after many steps. Long-run probabilities and long-run expectations can be useful for the analysis of the properties of systems in the long-run equilibrium after some initialization phase. This is helpful, e.g., to quantify the availability of system components. For a case study in this direction employing probabilistic model checking to analyze system availability, see [LPM$^+$15].

1.3 Contributions and structure of the thesis

In **Chapter 2**, we provide basic definitions and the notation we use throughout the thesis. Additionally, we briefly present some well-known, basic results for MDPs that are of importance in the subsequent chapters.

Chapter 3 addresses the non-classical stochastic shortest path problems. We first see that the partial and conditional expectation in Markov chains can be easily computed in polynomial time. For MDPs, however, we observe that these problems are considerably more complicated than the classical stochastic shortest path problem:

Result A. The optimal partial and conditional expectation in an MDP can be irrational. Infinite-memory schedulers can be necessary to achieve the optimal value.

This stands in strong contrast to the classical problem where the optimal value is rational, if finite, and can be achieved by a memoryless scheduler. Some techniques known from the classical setting to decide finiteness of the optimal values and to preprocess the input MDP can be transferred to the non-classical setting simplifying all subsequent investigations. Despite the irrationality of optimal values, we can show that optimal schedulers exist and that the memory requirement, although infinite, takes a rather simple form:

Result B. If the optimal partial (or conditional) expectation is finite in an MDP, the optimal value can be achieved by a weight-based deterministic scheduler, i.e., the optimal decisions depend only on the current state and the weight accumulated so far.

For MDPs with non-negative weights, it has been shown in [CFK$^+$13a,BKKW17] that the optimal partial and conditional expectations, respectively, can be computed in exponential time. The algorithms rely on the existence of a saturation point, a bound on the accumulated weight such that an optimal scheduler can behave memorylessly as soon as the accumulated weight exceeds that bound. We refine this result for partial expectations by providing the least possible saturation point. While the resulting algorithm still runs in exponential time, the considerably smaller saturation point we provide might lead to a considerably faster computation in practice. Finally, we show that the existence of a simple saturation point can be exploited to compute optimal conditional values-at-risk for accumulated weights before reaching a goal state in MDPs with non-negative weights as well.

In **Chapter 4**, we investigate the notions regarding the long-run behavior of a system: long-run frequencies and long-run probabilities of path properties and long-run expectations. In the non-probabilistic setting, we study the optimization of long-run frequencies in transition systems. After identifying easily solvable instances, we focus on regular co-safety properties given by non-deterministic finite automata (NFAs). By providing a product construction for a transition system and an NFA that keeps track of runs of the NFA on suffixes of a path in the transition system, we obtain the following result:

Result C. Given a transition system $\mathcal{T}$ and a regular co-safety property represented by an NFA $\mathcal{A}$, the optimal long-run frequency of the co-safety property can be computed in time polynomial in the size of $\mathcal{T}$ and exponential in the size of $\mathcal{A}$. The corresponding threshold problem is PSPACE-hard.

For the special case of constrained reachability properties $(a \, \mathrm{U} \, b)$, this implies that optimal long-run frequencies can be computed in polynomial time.

In probabilistic systems, the situation for long-run probabilities becomes more difficult. Again, for several types of path properties the optimal long-run probability can be computed efficiently. Turning the attention to regular co-safety properties, deterministic finite automata (DFA) are more suitable in the probabilistic setting as the product of an MDP and a DFA is well-behaved. For co-safety properties given as a DFA, we again provide a product construction starting from an MDP $\mathcal{M}$ and the DFA $\mathcal{D}$ that keeps track of the runs of the DFA on the suffixes of the path produced in the MDP. The constructed MDP $\mathcal{M}_{\mathcal{D}}$ is an infinite-state MDP equipped with a weight function. Using Fatou's lemma, we show that the optimal long-run probability of the property expressed by $\mathcal{D}$ in $\mathcal{M}$ and the optimal expected mean payoff in $\mathcal{M}_{\mathcal{D}}$ agree and can be approximated via finite-memory schedulers. While the infinite-state MDP $\mathcal{M}_{\mathcal{D}}$ does not allow the computation of the optimal expected mean payoff, we can show that for constrained reachability $(a \, \mathrm{U} \, b)$, the constructed MDP can be seen as the MDP $\mathcal{M}$ equipped with one counter. For this case, we can prove the existence of a saturation point similar to the saturation point for non-classical stochastic shortest path problems. This saturation point can then be employed to obtain the following result:

Result D. Given a labeled MDP $\mathcal{M}$, the optimal long-run probability of constrained reachability properties $(a \, \mathrm{U} \, b)$ can be computed in exponential time. The corresponding threshold problem is NP-hard.

For the analysis of the behavior of quantitative aspects of a system in the long-run, we introduce the notion of long-run expectation. Interestingly, even in MDPs with both positive and negative weights, we can prove the existence of a saturation point as a bound on the number of steps since the last visit to a goal state after which an optimal scheduler can switch to memoryless behavior. Again, this saturation point can be used for the computation of the optimal value:

Result E. The optimal long-run expectation in an MDP with arbitrary integer weights can be computed in exponential time. The corresponding threshold problem is NP-hard.

In **Chapter 5**, we prove that the non-classical stochastic shortest path problems, the computation of long-run probabilities, and several related problems studied in the literature exhibit an inherent mathematical difficulty that suggests that the problems are not solvable with known techniques. More precisely, we show that this series of problems is at least as hard as the *Positivity problem* for rational linear recurrence sequences.

This problem asks whether a linear recurrence sequence $(u_n)_{n\geq 0}$ given by k initial values $u_0, \ldots, u_{k-1}$ and a linear recurrence relation

$$u_{n+k} = \alpha_1 \cdot u_{n+k-1} + \alpha_2 \cdot u_{n+k-2} + \cdots + \alpha_k \cdot u_n$$

of depth k with rational coefficients $\alpha_1, \ldots, \alpha_k$ stays non-negative, i.e., whether $u_n \geq 0$ for all n. The famous *Skolem problem* asking whether such a sequence has a zero is reducible to the Positivity problem. Both problems have been open for many decades and defy all solution attempts with known number-theoretic techniques (see [HHHK05,OW14b]). The decidability of the Positivity problem would constitute a major breakthrough in analytic number theory, more precisely in the field of Diophantine approximation of transcendental numbers (see [OW14b]).

To obtain our Positivity-hardness results, we construct an MDP-gadget that ensures that the difference of the optimal values of various optimization problems in two states satisfies a linear recurrence relations when increasing the level of accumulated weight. This gadget forms the basis for all the Positivity-hardness proofs. For several optimization problems, we then construct gadgets that also encode the initial values of a linear recurrence sequence for low levels of accumulated weight. In this way, we can encode arbitrary linear recurrence sequences into the optimal values of the respective problems. This encoding allows us then to reduce the Positivity problem to the threshold problems of the respective optimization problems. Using reductions from further problems and adjusting the construction when necessary, we can transfer the Positivity-hardness to various other problems:

Result F. The Positivity problem is polynomial-time reducible to the threshold problems for the optimal values of the following quantities:

- partial and conditional expectations,
- a two-sided version of partial expectations in MDPs with non-negative weights,
- long-run probabilities of regular co-safety properties,
- conditional values-at-risk for accumulated weights before reaching a goal,
- termination probabilities and termination times of one-counter MDPs,
- the satisfaction probability of energy objectives, and
- the probability that the accumulated weight when reaching a target state satisfies an inequality constraint (cost problem).

Furthermore, algorithms for the following problems would imply the decidability of the Positivity problem:

- the model-checking problem of frequency-LTL (as defined in [FK15, FKK15]),
- the computation of quantiles for the accumulated weight (before reaching a goal).

An overview over the Positivity-hardness results and the decidable restricted versions of the problems is shown in Table 1.1. The dependencies between the Positivity-hardness results is also depicted in Figure 5.1 in Chapter 5.

As the Positivity-hardness results show that exact solutions to the non-classical stochastic shortest path problems are not in sight without overcoming major difficulties or might even be undecidable, we focus on the approximability of the optimal values in **Chapter 6**. Using the fact that saturation points still provide us with information on the behavior of optimal schedulers in the setting of arbitrary weights together with estimations on the possible growth of weights in end components with negative maximal expected mean-payoff, we provide a way to approximate optimal values with finite-memory schedulers. This allows us to conclude that the optimal values can be approximated with reasonable complexity:

Result G. Given an MDP $\mathcal{M}$ and $\varepsilon > 0$, the maximal partial and conditional expectations in $\mathcal{M}$ can be approximated up to an absolute error of ε in time exponential in the size of $\mathcal{M}$ and polynomial in $\log(1/\varepsilon)$.

As the approximation algorithms behave well with respect to the desired precision ε, these approximation algorithms have the potential to provide sufficiently accurate values for practical applications. To conclude the chapter, we show that there are no polynomial-time approximation algorithms if P $\neq$ PSPACE.

In **Chapter 7**, we conclude with remarks on the presented work and on possible future directions of research.

The results presented in this thesis have partially been published in [PB19], [BBPS19], and [PB20]. At the beginning of each chapter, we describe the publication status of the results more precisely and point out which results of this thesis constitute extensions of the contributions of these publications.

	Optimum computable in exponential time	Positivity-hard threshold problem
Partial expectations	weights in $\mathbb{N}$ [CFK+13a] **(PSPACE-h., Sec. 3.5)**	**weights in $\mathbb{Z}$, Sec. 5.2.1 (exponential-time approximation algorithm, Sec. 6.2)**
Conditional expectations	weights in $\mathbb{N}$ [BKKW17] (PSPACE-h. [BKKW17])	**weights in $\mathbb{Z}$, Sec. 5.2.1 (exponential-time approximation algorithm, Sec. 6.3)**
Conditional value-at-risk for accumulated weights	**weights in $\mathbb{N}$, Sec. 3.6**	**weights in $\mathbb{Z}$, Sec. 5.2.4**
Cost problems, quantiles	weights in $\mathbb{N}$ [HK15, UB13] (PSPACE-hard [HK15])	**weights in $\mathbb{Z}$, Sec. 5.2.3**
One-counter MDPs	qualitative termination problem with target state [BBE+10] (PSPACE-hard, in polynomial time without target state [BBE+10])	**termination probability (without target state), termination time, Sec. 5.2.3**
Energy objectives	almost-sure satisfaction [CD11]	**satisfaction probability, Sec. 5.2.3**
Long-run probability	**constrained reachability properties (NP-hard), Sec. 4.2.3**	**regular co-safety properties, Sec. 5.2.2**
Long-run expectation	**weights in $\mathbb{Z}$ (NP-hard), Sec. 4.3**	—

Table 1.1: Overview of the results. Contributions of this thesis are written in bold face. The hardness results in the middle column refer to the threshold problems.

CHAPTER TWO

PRELIMINARIES

This chapter contains basic definitions and our notation. Furthermore, we state preliminary results from the literature that are employed throughout the thesis.

2.1 Markov decision processes

Finite-state Markov decision processes (MDPs) are the main model we work with in this thesis. We assume some familiarity with MDPs and present the basic notions and preliminary results only briefly. More details can be found in textbooks, e.g., [Put94].

2.1.1 Definitions

We begin by defining MDPs, Markov chains, schedulers, and the resulting probability measure on maximal paths in an MDP. Further, we sketch how the quotient of an MDP by its maximal end components is constructed.

Markov decision process. A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (S, Act, P, s_{init})$ where

- S is a finite set of states,
- Act is a finite set of actions,
- $P\colon S \times Act \times S \to [0,1] \cap \mathbb{Q}$ is the transition probability function for which we require that $\sum_{t\in S} P(s,\alpha,t) \in \{0,1\}$ for all $(s,\alpha) \in S \times Act$, and
- $s_{init} \in S$ is the initial state.

Depending on the context, we enrich MDPs with

- a weight function $\mathit{wgt}\colon S \times \mathit{Act} \to \mathbb{Z}$,
- a finite set of atomic propositions AP and a labeling function $L\colon S \to 2^{\mathsf{AP}}$, or
- a designated set of goal states Goal.

The *size* of an MDP $\mathcal{M}$, denoted by $\mathit{size}(\mathcal{M})$, is the sum of the number of states plus the total sum of the logarithmic lengths of the non-zero probability values $P(s,\alpha,s')$ as fractions of co-prime integers and the logarithmic lengths of the weight values $\mathit{wgt}(s,\alpha)$.

We write $\mathit{Act}(s)$ for the set of actions that are enabled in a state s, i.e., $\alpha \in \mathit{Act}(s)$ iff $\sum_{t\in S} P(s,\alpha,t) = 1$. Whenever the process is in a state s, a non-deterministic choice between the enabled actions $\mathit{Act}(s)$ has to be made. We call a state *absorbing* if the only enabled actions lead to the state itself with probability 1 and weight 0. If there are no enabled actions, we call a state a *trap*.

The paths of $\mathcal{M}$ are finite or infinite sequences $s_0\,\alpha_0\,s_1\,\alpha_1\,s_2\,\alpha_2\dots$ where states and actions alternate such that $P(s_i,\alpha_i,s_{i+1}) > 0$ for all $i \geq 0$. Throughout this section, we assume that all states are reachable from the initial state in any MDP, i.e., that there is a finite path from s_{init} to each state s. We extend the weight function to finite paths. For a finite path $\pi = s_0\,\alpha_0\,s_1\,\alpha_1\,\dots\alpha_{k-1}\,s_k$, we denote its accumulated weight by

$$\mathit{wgt}(\pi) = \mathit{wgt}(s_0,\alpha_0) + \ldots + \mathit{wgt}(s_{k-1},\alpha_{k-1}).$$

Similarly, we extend the transition probability function to finite paths and write

$$P(\pi) = P(s_0,\alpha_0,s_1) \cdot \ldots \cdot P(s_{k-1},\alpha_{k-1},s_k).$$

A *Markov chain* is an MDP in which the set of actions is a singleton. There are no non-deterministic choices in a Markov chain and hence we drop the set of actions. Consequently, a Markov chain is a tuple $\mathcal{M} = (S,P,s_{\mathit{init}})$, possibly extended with a weight function, a labeling, or a designated set of goal states. The transition probability function P is a function from $S \times S$ to $[0,1]\cap\mathbb{Q}$ such that $\sum_{t\in S} P(s,t) \in \{0,1\}$ for all $s \in S$.

Remark 2.1 (Rational weights)**.** We could as well allow rational weights instead of integer weights in the definition of MDPs. An MDP with rational weights can easily be transformed to an integer weighted MDP by multiplying all weights with the least common multiple of all denominators of the weights appearing in the MDP. All quantities of interest, such as expected accumulated weights, scale accordingly. $\triangleleft$

Remark 2.2 (Infinite state space)**.** The focus of this thesis lies on optimization and decision problems in finite-state MDPs. Hence, we included the requirement that the state space is finite in the definition. Nevertheless, we will encounter infinite-state MDPs on a few occasions. In these situations, it will be made clear that we drop the requirement of a finite state space. ◁

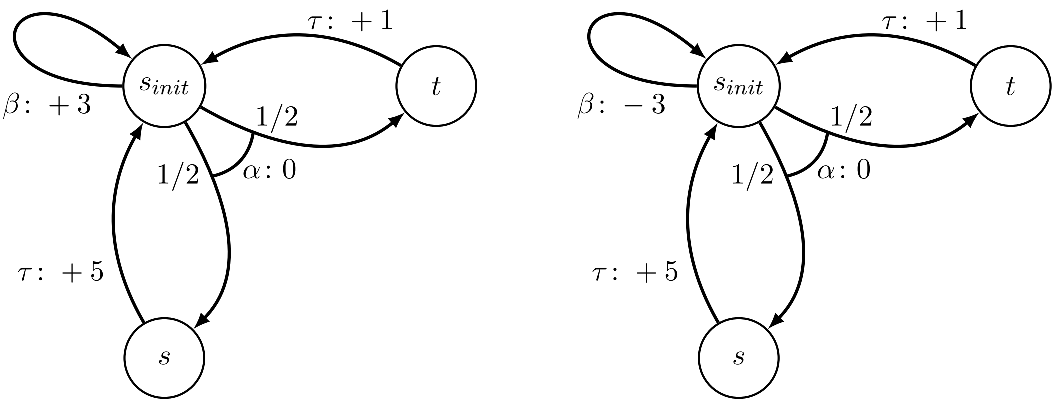
Scheduler. A *scheduler* for an MDP $\mathcal{M} = (S, Act, P, s_{init})$ is a function $\mathfrak{S}$ that assigns to each finite path π a probability distribution over $Act(last(\pi))$ where $last(\pi)$ denotes the last state of π. This probability distribution indicates which of the enabled actions is chosen with which probability under $\mathfrak{S}$ after the process has followed the finite path π. Given a scheduler $\mathfrak{S}$, a path $\zeta = s_0\,\alpha_0\,s_1\,\alpha_1\ldots$ is a $\mathfrak{S}$-*path* iff ζ is a path and $\mathfrak{S}(s_0\,\alpha_0\,s_1\,\alpha_1\ldots\alpha_{k-1}\,s_k)(\alpha_k) > 0$ for all $k \geq 0$.

We allow schedulers to be *randomized* and *history-dependent*. By restricting the possibility to randomize over actions or by restricting the amount of information from the history of a run that can affect the choice of a scheduler, we obtain the following types of schedulers: A scheduler $\mathfrak{S}$ is called *deterministic* if it does not make use of the possibility to randomize over actions, i.e., if $\mathfrak{S}(\pi)$ is a Dirac distribution for each path π. Such a scheduler $\mathfrak{S}$ can be viewed as a function that assigns an action to each finite path π. A scheduler $\mathfrak{S}$ is called *memoryless* if $\mathfrak{S}(\pi) = \mathfrak{S}(\pi')$ for all finite paths π, π' with $last(\pi) = last(\pi')$. In this case, $\mathfrak{S}$ can be viewed as a function that assigns to each state s a distribution over $Act(s)$. A memoryless deterministic scheduler hence can be seen as a function from states to actions. In an MDP with a weight function, a scheduler $\mathfrak{S}$ is said to be *weight-based* if $\mathfrak{S}(\pi) = \mathfrak{S}(\pi')$ for all finite paths π, π' with $wgt(\pi) = wgt(\pi')$ and $last(\pi) = last(\pi')$. Such a scheduler assigns distributions over actions to state-weight pairs from $S \times Act$. Finally, let X be a finite set of memory modes with initial mode x_{init} and $U : X \times S \times Act \times S \to X$ a memory update function. From a finite path $\pi = s_0\,\alpha_0\,s_1\,\alpha_1\,\ldots\alpha_{k-1}\,s_k$ we can extract a sequence of memory modes $x_0\,\ldots\,x_k$. We let $x_0 = x_{init}$, and $x_{i+1} = U(x_i, s_i, \alpha_i, s_{i+1})$ for all $i < k$. Let us denote the last memory mode x_k after the finite path π by $U(x_{init}, \pi)$. A scheduler $\mathfrak{S}$ is a *finite-memory scheduler* if there is such a finite set of memory modes X with an initial mode x_{init} and an update function U such that $\mathfrak{S}(\pi) = \mathfrak{S}(\pi')$ for all finite paths π, π' with $U(x_{init}, \pi) = U(x_{init}, \pi')$ and $last(\pi) = last(\pi')$.

Example 2.3 (Finite- vs infinite-memory schedulers)**.** Consider the example MDPs depicted in Figure 2.1. We use arrows connected by an arc to depict transitions belonging to the same action. All non-trivial probability values are denoted next to the arrows. In the two example MDPs, there is a non-deterministic choice between actions α and β in the initial state s_{init}. Except for the weight of the state-action pair (s_{init}, β), the two MDPs are identical. Let $\mathfrak{S}$ be the scheduler for $\mathcal{M}$ given by

$$\mathfrak{S}(\pi) = \begin{cases} \beta & \text{if } wgt(\pi) > 3 \text{ and } wgt(\pi) \text{ is even,} \\ \alpha & \text{otherwise.} \end{cases}$$

for all finite paths π ending in s_{init}. Further, let $\mathfrak{S}'$ be the scheduler for $\mathcal{N}$ given by the same definition. Both schedulers, $\mathfrak{S}$ for $\mathcal{M}$ and $\mathfrak{S}'$ for $\mathcal{N}$, are weight-based and deterministic. Note that the functions from finite paths to probability distributions over actions given by $\mathfrak{S}$ and $\mathfrak{S}'$ are different in the MDPs $\mathcal{M}$ and $\mathcal{N}$ as the definition of the schedulers depends on the weight functions.



(a) Example MDP $\mathcal{M}$. **(b)** Example MDP $\mathcal{N}$.

Figure 2.1: Example MDPs to illustrate finite- and infinite-memory weight-based schedulers. Note that the weight of action β in state s_{init} is positive in $\mathcal{M}$, while it is negative in $\mathcal{N}$.

In the MDP $\mathcal{M}$, the scheduler $\mathfrak{S}$ only requires finite memory: as memory modes we can use the set $\{0, 1, 2, 3, \mathit{even}, \mathit{odd}\}$. The update function now updates the memory mode as follows: As long as the accumulated weight is 3 or less, the memory mode equals that weight. As soon as the accumulated weight exceeds 3, the memory only keeps track of the parity of the accumulated weight. As all weights are non-negative, the accumulated weight along a path stays above 3 once it exceeded this bound. Thus, the parity is sufficient to determine which action the scheduler chooses.

In the MDP $\mathcal{N}$, however, the scheduler $\mathfrak{S}'$ is not a finite-memory scheduler. No matter how much weight has been accumulated on a finite path, there is always a positive probability that this weight drops below 3 again. Namely, starting in s_{init} with even weight higher than 5, the path $s_{init}\,\beta\,s_{init}\,\alpha\,t\,\tau\,s_{init}$ decreases the weight by 2 and might be repeated until the accumulated weight is 2. Then, the scheduler does not choose β again but α although the weight is even. Therefore, the scheduler has to keep track of the accumulated weight for arbitrarily high values and finite memory is not sufficient. $\lhd$

Once a scheduler $\mathfrak{S}$ is specified for an MDP $\mathcal{M} = (S, Act, P, s_{init})$, the behavior under this scheduler is purely probabilistic. The scheduler induces a natural infinite-state Markov chain: The states of the Markov chain are the finite paths of $\mathcal{M}$. The path s_{init} only consisting of the initial state of $\mathcal{M}$ takes the role of the initial state in the Markov chain. The transitions probability from a finite paths π to an extension of the form $\pi\,\alpha\,s$ with $s \in S$ and $\alpha \in \mathit{Act}$ is given by $\mathfrak{S}(\pi)(\alpha) \cdot P(\mathit{last}(\pi), \alpha, s)$. This is simply the probability that $\mathfrak{S}$ chooses α after π and that the next state is s. If $\mathfrak{S}$ is a finite-memory scheduler with the finite set of memory modes X, initial mode x_{init} and memory update function U, the induced Markov chain can be seen as a finite-state Markov chain with state space $S \times X$. The initial state is $(s_{\mathit{init}}, x_{\mathit{init}})$. The probability to move from (s,x) to (t,y) is given by

$$\sum_{\alpha \in \mathit{Act}(s), y = U(x,s,\alpha,t)} \mathfrak{S}(s,x)(\alpha) \cdot P(s,\alpha,t).$$

Probability measure. Given an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}})$ and a scheduler $\mathfrak{S}$, we obtain a probability measure $\Pr^{\mathfrak{S}}_{\mathcal{M},s}$ on the set of maximal paths of $\mathcal{M}$ that start in s: For each finite paths $\pi = s_0\,\alpha_0\,s_1\,\alpha_1\,\ldots\alpha_{k-1}\,s_k$ with $s_0 = s$, we denote the cylinder set of all its maximal extensions by $\mathit{Cyl}(\pi)$. The probability mass of this cylinder set is then given by

$$\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\mathit{Cyl}(\pi)) = P(\pi) \cdot \Pi_{i=0}^{k-1} \mathfrak{S}(s_0\,\ldots\,s_i)(\alpha_i).$$

Recall that $\mathfrak{S}(s_0\,\ldots\,s_i)$ is a probability distribution over actions and that $\mathfrak{S}(s_0\,\ldots\,s_i)(\alpha_i)$ denotes the probability that the scheduler $\mathfrak{S}$ chooses action α after the prefix $s_0\,\ldots\,s_i$ of π. In particular, that means that the cylinder set $\mathit{Cyl}(\pi)$ has positive probability under $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\mathit{Cyl}(\pi))$ iff π is a $\mathfrak{S}$-path. The set of cylinder sets forms the basis of the standard tree topology on the set of maximal paths. By Carathéodory's extension theorem, we can extend the pre-measure $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\mathit{Cyl}(\pi))$ defined on the cylinder sets to a probability measure on the Borel σ-algebra of the space of maximal paths with the standard tree topology. We sometimes drop the subscript s if s is the initial state s_{init} of $\mathcal{M}$. In a Markov chain $\mathcal{N}$, we drop the reference to a scheduler and write $\Pr_{\mathcal{N},s}$.

Let X be a random variable on the set of maximal paths of $\mathcal{M}$ starting in s, i.e., X is a function assigning values from $\mathbb{R} \cup \{-\infty, +\infty\}$ to maximal paths. We denote the expected value of X under the probability measure $\Pr^{\mathfrak{S}}_{\mathcal{M},s}$ by $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(X)$.

The values we are typically interested in are the worst- or best-case probabilities of an event or the worst- or best-case expected values of a random variable. Worst or best case refers to the possible ways to resolve the non-deterministic choices. Hence, these values are formally expressed by taking the supremum or infimum over all schedulers. Given an MDP $\mathcal{M}$, a state s, an event, i.e., a path set, E, and a random variable X on the maximal paths of $\mathcal{M}$, we define

$$\Pr^{\max}_{\mathcal{M},s}(E) = \sup_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s}(E), \qquad \Pr^{\min}_{\mathcal{M},s}(E) = \inf_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M},s}(E),$$

$$\mathbb{E}^{\max}_{\mathcal{M},s}(X) = \sup_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(X), \text{ and} \qquad \mathbb{E}^{\min}_{\mathcal{M},s}(X) = \inf_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(X),$$

where inf and sup range over all schedulers $\mathfrak{S}$ for $\mathcal{M}$.

Remark 2.4 (Measurability)**.** For all events and for all random variables investigated in this thesis, measurability is easily established. In particular, all ω-regular sets of maximal paths and hence also all sets of maximal paths specified by common temporal logics such as linear temporal logic (LTL) are measurable. Therefore, measurability questions will be no concern throughout the thesis and we will not explicitly address these questions. For more details, see [Put94]. ◁

End components, MEC-quotient. Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}})$ be an MDP. An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP. End components can be formalized as pairs $\mathcal{E} = (E, \mathfrak{A})$ where E is a nonempty subset of S and $\mathfrak{A}$ a function that assigns to each state $s \in E$ a nonempty subset of $\mathit{Act}(s)$ such that the graph induced by the transitions with non-zero probability in $\mathcal{E}$ is strongly connected. In other words, there has to be a path inside $\mathcal{E}$ between any two states s and t in E that only uses actions belonging to the end-component $\mathcal{E}$.

An end component $\mathcal{E}$ is called *maximal* if there is no end component $\mathcal{E}' = (E', \mathfrak{A}')$ with $\mathcal{E} \neq \mathcal{E}'$, $E \subseteq E'$ and $\mathfrak{A}(s) \subseteq \mathfrak{A}'(s)$ for all $s \in E$. For each state s there is a unique maximal end component $\mathcal{E}$ containing s. Note that a single state without any actions is also an end component. We call such end components trivial end components. The *quotient by maximal end components (MEC-quotient)* of an MDP $\mathcal{M}$ is the following MDP $\mathit{MEC}(\mathcal{M})$: The set of states S_{MEC} is the set of maximal end components of $\mathcal{M}$. All actions α that are enabled in some state s of $\mathcal{E} = (E, \mathfrak{A})$ and do not themselves belong to $\mathcal{E}$, i.e., $\alpha \notin \mathfrak{A}(s)$, are enabled in the state $\mathcal{E}$ of $\mathit{MEC}(\mathcal{M})$. Let us assume that the sets of enabled actions at different states in $\mathcal{M}$ are disjoint. Then, for each state $s \in E$ and each action $\alpha \in \mathit{Act}(s) \setminus \mathfrak{A}(s)$, the action α is enabled in the state $\mathcal{E}$. The transition probability function P_{MEC} in $\mathit{MEC}(\mathcal{M})$ is given by $P_{\mathit{MEC}}(\mathcal{E}, \alpha, \mathcal{F}) = \sum_{t \in F} P(s, \alpha, t)$ for any maximal end component $\mathcal{F} = (F, \mathfrak{B})$. The initial state of the MDP $\mathit{MEC}(\mathcal{M})$ is $\mathcal{E}_{\mathit{init}}$, the maximal end component containing the initial state s_{init}. Maximal end-components and the MEC-quotient are computable in polynomial time [dA97, CH11]. For more details and the formal construction of the MEC-quotient, we refer to [dA97, CBGK08].

Non-trivial end components of Markov chains are called *bottom strongly connected components* (BSCC). In other words, a BSCC is a subset of states T such that any state inside T is reachable from any other state in T and such that there are no transitions from a T-state to a state not in T. It is well-known that the states occurring infinitely often on a path form a BSCC for almost all paths in a Markov chain (see, e.g., [dA97]).

2.1.2 Preliminary results

Several optimization problems on MDPs addressing optimal probabilities of simple events or optimal expected values are solvable in polynomial time via a linear program. We briefly state a few of these results that are of importance to the contents of this thesis.

Optimal reachability probabilities. Computing the minimal or maximal probability of the event $\Diamond T$ that a given set of states T is eventually reached is a recurrent task in the formal verification of probabilistic systems. A linear programming approach makes it possible to solve this important problem in polynomial time.

Let us briefly sketch how maximal reachability probabilities can be computed following the procedure given in [dA97]: Let $\mathcal{M} = (S, Act, P, s_{init})$ be an MDP. Let $T \subseteq S$ be a set of target states. Let $\mathcal{T}$ be the set of states in $MEC(\mathcal{M}) = (S_{MEC}, Act_{MEC}, P_{MEC}, \mathcal{E}_{init})$ corresponding to maximal end components containing a state from T. As a scheduler can reach any state within a maximal end component from any other state in that end component with probability 1, we have

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\Diamond T) = \Pr^{\max}_{MEC(\mathcal{M}),\mathcal{E}_{init}}(\Diamond \mathcal{T}).$$

In the MEC-quotient there are no end components except for trivial one-state end components. Hence, with probability 1 under any scheduler for $MEC(\mathcal{M})$, a trap state or a state in $\mathcal{T}$ is reached. Let us denote the set of states in S_{MEC} from which $\mathcal{T}$ is unreachable by $\mathcal{F}$. The maximal reachability probability can now be computed via the following linear program with one variable x_s per state $s \in S_{MEC}$: Minimize $\sum_{s \in S_{MEC}} x_s$ under the conditions

$$\begin{aligned}
&x_s = 1 && \text{for } s \in \mathcal{T},\\
&x_s = 0 && \text{for } s \in \mathcal{F},\\
&x_s \geq \sum_{t \in S'} P_{MEC}(s,\alpha,t) \cdot x_t && \text{for } s \in S_{MEC} \setminus (\mathcal{T} \cup \mathcal{F}) \text{ and } \alpha \in Act_{MEC}(s).
\end{aligned}$$

By assigning 0 to the states in $\mathcal{F}$, we ensure that the linear program has a unique solution. The pre-processing step to move to the MEC-quotient might considerably decrease the size of the linear program. As it only requires graph algorithms and can disregard the probability values, this can lead to a notable speed-up.

From the solution, we can extract a memoryless deterministic scheduler that realizes the optimal reachability probability from each state. In the MEC-quotient, we can simply check for which actions equality is obtained when plugging the optimal values into the respective inequalities in the linear program. The traversal of the maximal end components in the original MDP can easily be done in a memoryless fashion as well. Minimal reachability probabilities can be computed via a similar linear program. In Markov chains, we can formulate a system of linear equations instead of a linear program to compute reachability probabilities.

Weighted reachability. A well-known generalization of the optimization of reachability probability is the optimization of expected terminal weights in a *weighted reachability* problem to which the approach of [dA97] can easily be extended: Given an MDP $\mathcal{M} = (S, Act, P, s_{init})$, we not only are given a set of terminal target states T, but also a terminal weight $w_t \in \mathbb{Q}$ for each state in T. The expected terminal weight under scheduler $\mathfrak{S}$ is $\sum_{t \in T} \mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(\lozenge t) \cdot w_t$. We want to maximize this expected value among all schedulers that reach T with probability 1. We can use the MEC-quotient as above to make sure that T is reached with probability 1 under any scheduler. The maximal expected terminal weight can now be computed via a linear program very similar to the program above: Minimize $\sum_{s \in S} x_s$ under the conditions

$$\begin{aligned} x_t &= w_t && \text{for } t \in T, \\ x_s &\geq \sum_{t \in S} P(s, \alpha, t) \cdot x_t && \text{for } s \in S \setminus T \text{ and } \alpha \in Act(s). \end{aligned}$$

The value x_s in the optimal solution equals the maximal expected terminal weight when starting in s.

Mean payoff. A well-known measure for the long-run behavior of a scheduler $\mathfrak{S}$ in an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt)$ is the expected *mean payoff*. Intuitively, the mean payoff is the amount of weight accumulated per step on average in the long run. Formally, we define the mean payoff as the random variable MP on infinite paths $\zeta = s_0 \alpha_0 s_1 \alpha_1 \dots$ by

$$MP(\zeta) = \liminf_{k \to \infty} \frac{\sum_{i=0}^{k} wgt(s_i, \alpha_i)}{k+1}.$$

We assume that there are no trap states in $\mathcal{M}$, so all maximal paths are infinite. The expected mean payoff of the scheduler $\mathfrak{S}$ is defined as the expected value $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(MP)$. The maximal expected mean payoff is the supremum over all schedulers. It is well-known that this supremum is equal to the maximum over all memoryless deterministic schedulers (see, e.g., [Put94]). In strongly connected MDPs, the maximal expected mean payoff does not depend on the initial state. In order to compute the maximal expected mean payoff, it is easiest to consider the non-trivial maximal end components separately first. Inside a maximal end component, we can employ a linear programming approach as presented in [HK79] and [dA97]. Let $\mathcal{E} = (E, \mathfrak{A})$ be a maximal end component. The maximal expected mean payoff in $\mathcal{E}$ is given by the value of g in an optimal solution to the following linear program with one variable u_s per state $s \in E$ as well as the variable

g: Minimize g under the conditions

$$u_s + g \geq wgt(s,\alpha) + \sum_{t\in E} P(s,\alpha,t)\cdot u_t \qquad \text{for } s\in E, \alpha\in\mathfrak{A}(s).$$

The values of the variables $(u_s)_{s\in E}$ can be interpreted as a *potential* (as the term is used in physics for example). A high value of the variable u_s indicates that the maximal expected accumulated weight after a large number of steps when starting from state s is comparably high. Intuitively that means that in the initial part of a run, starting in s makes it possible to collect high amounts of weights before the expected collected weight per step gets closer to the expected mean payoff. We can make this intuition of a potential precise: Let $(u_s)_{s\in E}$ and g be an optimal solution to the linear program. For each pair of states s and t form E, we have that

$$\lim_{n\to\infty}\Big(\mathbb{E}^{\max}_{\mathcal{E},s}(\text{weight after } n \text{ steps}) - \mathbb{E}^{\max}_{\mathcal{E},t}(\text{weight after } n \text{ steps})\Big) = u_s - u_t.$$

For details, consult [Kal11]. Note that the value of a single variable u_s does not have a meaningful interpretation. In particular, we can obtain another optimal solution by leaving g unchanged and adding the same number $u\in\mathbb{R}$ to all values u_s with $s\in E$. This is exactly what we expect from a potential; only the pairwise differences of the values are important. Once, the optimal values in each maximal end components are known, the optimal expected mean payoff can be computed by solving a weighted reachability problem in the MEC-quotient.

Long-run distribution of Markov chains. Let $\mathcal{M} = (S, P, s_{\mathit{init}})$ be a Markov chain. The *long-run distribution* of $\mathcal{M}$ is given by the values

$$\theta^{\mathcal{M}}_t = \lim_{n\to\infty}\frac{1}{n+1}\sum_{i=0}^{n}\Pr_{\mathcal{M},s_{\mathit{init}}}(\text{state after } i \text{ steps is } t).$$

We also call these values *steady state probabilities.* These values exist for all (finite-state) Markov chains. If $\mathcal{M}$ is strongly connected, the values $\theta^{\mathcal{M}}_t$ are the values for the variables x_t with $t\in T$ in the unique solution to the set of equations

$$\begin{aligned}\sum_{t\in S} x_t &= 1,\\ x_s &= \sum_{t\in S} x_t\cdot P(t,s), \qquad \text{for all } s\in S.\end{aligned}$$

In arbitrary Markov chains $\mathcal{M} = (S, P, s_{\mathit{init}})$, the values for states t inside a BSCC $\mathcal{B}$ with states B are the product of the values $\theta^{\mathcal{B}}_t$ and the probability $\Pr_{\mathcal{M},s_{\mathit{init}}}(\Diamond B)$ that the BSCC is reached in $\mathcal{M}$. Hence, the long-run distribution can be computed in polynomial time. For more details, see [BK08, Kul16].

The long-run distribution is a valuable tool when analyzing the limiting behavior of a Markov chain. In particular, the expected mean-payoff in a Markov chain $\mathcal{M} = (S, P, s_{\mathit{init}}, \mathit{wgt})$ with a weight function $\mathit{wgt} : S \to \mathbb{Z}$ is given by

$$\mathbb{E}_{\mathcal{M},s_{\mathit{init}}}(MP) = \sum_{t \in S} \theta_t^{\mathcal{M}} \cdot \mathit{wgt}(t).$$

2.2 Linear temporal logic and automata

We assume familiarity with linear temporal logic and finite automata. We briefly present our notation here.

Linear temporal logic (LTL). Let AP be a finite set of atomic propositions. The syntax of linear temporal logic (LTL) over AP is given by

$$\varphi ::= a | \varphi \wedge \varphi | \neg \varphi | \bigcirc \varphi | \varphi \, \mathrm{U} \, \varphi$$

where $a \in \mathsf{AP}$. The semantics of LTL are given on words in $(2^{\mathsf{AP}})^{\omega}$. For a word $w = w_0 w_1 w_2 \ldots$, the semantics are recursively defined as follows:

$$\begin{array}{lll}
w \vDash a & \text{iff} & a \in w_0, \\
w \vDash \varphi \wedge \psi & \text{iff} & w \vDash \varphi \text{ and } w \vDash \psi, \\
w \vDash \neg \varphi & \text{iff} & w \nvDash \varphi, \\
w \vDash \bigcirc \varphi & \text{iff} & w_1 w_2 w_3 \cdots \vDash \varphi, \\
w \vDash \varphi \, \mathrm{U} \, \psi & \text{iff} & \text{there is } j \in \mathbb{N} \text{ with } w_j w_{j+1} w_{j+2} \cdots \vDash \psi \text{ and} \\
 & & w_i w_{i+1} w_{i+2} \cdots \vDash \varphi \text{ for all } i < j.
\end{array}$$

We use the usual Boolean abbreviations and the common abbreviations $\Diamond \varphi$ for $\mathit{true} \, \mathrm{U} \, \varphi$ stating that φ holds eventually (on some suffix) and $\Box \varphi$ for $\neg \Diamond \neg \varphi$ stating that φ holds globally (on all suffixes). For more details, consult, e.g., [BK08]. Furthermore, we use LTL-like notation to denote events such as "$\Diamond$(accumulated weight < 0)" expressing the set of paths in a weighted structure with a prefix π such that the accumulated weight of π is less than 0.

Nondeterministic finite automata (NFA). An NFA is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, Q_0, F)$ where Q is a finite set of states, Σ a finite alphabet, $\Delta \subseteq Q \times \Sigma \times Q$ the transition relation, $Q_0 \subseteq Q$ the set of initial states and $F \subseteq Q$ the set of final states. Let $w = w_0 \ldots w_n \in \Sigma^*$ be a finite word over Σ. A run of $\mathcal{A}$ on w is a sequence of states $q_0 \ldots q_{n+1}$ from Q such that $q_0 \in Q_0$ and such that $(q_i, w_i, q_{i+1}) \in \Delta$ for all $i \leq n$. The run is accepting if

$q_{n+1} \in F$. The word w is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w. The language $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of accepted words from Σ^*.

Deterministic finite automata (DFA). An DFA is a tuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ a finite alphabet, $\delta\colon Q \times \Sigma \to Q$ the transition function, $q_0 \in Q$ the initial state and $F \subseteq Q$ the set of final states. For a word $w = w_0 \dots w_n \in \Sigma^*$, the unique run of $\mathcal{D}$ on w is the sequence $q_0 \dots q_{n+1}$ starting with the initial state q_0 and satisfying $q_{i+1} = \delta(q_i, w_i)$ for all $i \leq n$. The word w is accepted by $\mathcal{D}$ if the unique run ends in an accepting state, i.e., if $q_{n+1} \in F$. Again, the language $\mathcal{L}(\mathcal{D})$ is the set of words from Σ^* accepted by $\mathcal{D}$.

Regular co-safety property. Let AP be a finite set of atomic propositions. A co-safety property is a set of words $\Pi \subseteq (2^{\mathsf{AP}})^\omega$ with the following property: for each $w \in \Pi$, there is a prefix π such that $\mathit{Cyl}(\pi) \subseteq \Pi$, i.e., all extensions of π are in Π. In terms of the usual tree-topology induced by the cylinder sets, co-safety properties are precisely the open sets. An (ω-)regular co-safety property can be given in terms of an NFA or DFA $\mathcal{A}$: The automaton $\mathcal{A}$ accepts all finite words π such that $\mathit{Cyl}(\pi)$ belongs to the co-safety property. So, a word w satisfies to the co-safety property given by $\mathcal{A}$ iff it has a prefix accepted by $\mathcal{A}$.

CHAPTER THREE

NON-CLASSICAL STOCHASTIC SHORTEST PATH PROBLEMS

Stochastic shortest path problems lie at the heart of many verification problems for systems that exhibit probabilistic and non-deterministic behavior. In a finite-state MDP that can be used to model such systems, these problems ask for the maximal or minimal expected accumulated weight before reaching a target state. In the classical setting, it is required that a target state is reached almost surely. As discussed in Section 1.1, there are various reasons why the classical stochastic shortest path problem might not be applicable or does not cover all relevant executions of the system due to this restriction to schedulers that reach a target state with probability 1. Typical situations in which this is the case include, e.g., the analysis of probabilistic programs that do not necessarily terminate almost surely or the analysis of the costs caused in case of an error scenario that occurs with small but positive probability. In this chapter, we turn our hands to two non-classical variants of the stochastic shortest path problem in which it is not required that the target is reached almost surely. For the switch to the non-classical variants, we have to specify a way to treat executions that do not reach the target. Depending on the quantitative aspect of a system that is modeled and the verification question under consideration, different treatments of these executions are appropriate. The two variants we consider are partial expectations and conditional expectations.

For partial expectations, we assign weight 0 to runs missing the target. An example situation in which this treatment is appropriate is the following: A system component processes data and tries to transmit the results to other components. The transmission might, however, fail with some positive probability. If we are interested in the best- or worst-case expected amount of data that is successfully transmitted after an execution of the component, we naturally arrive at a problem that asks to maximize or minimize the expected accumulated weight (amount of processed data) while assigning weight 0 to executions that do not reach the target (successful transmission of the results). Note that implicitly the optimization of partial expectations requires an analysis of the trade-off between the accumulation of high amounts of weights and a high probability to reach the target.

For conditional expectations, we consider the expected accumulated weight before reaching a target state under the condition that a target state is reached. In other words, the conditional expectation is the average accumulated weight of all paths reaching a target state. In particular in situations where the target is reached with a small probability, the conditional expectation can be a useful measure. Consider for example a repair mechanism that is activated if a certain unlikely error scenario occurs. If we are interested in the expected energy consumption of the repair mechanism, the conditional expectation under the condition that the error scenario occurs provides information on the amount of energy that should be available for the repair mechanism for example. The conditional expectation provides bounds on the expected energy consumption during one execution of the repair mechanism. If we used the partial expectation assigning weight 0 to runs in which the error scenario does not occur, we would obtain the expected energy consumption by the repair mechanism during one execution of the whole system. For errors with small probabilities, the latter value might still be interesting, but the conditional expectation seems to provide the more relevant information. It is worth mentioning that the classical stochastic shortest path problem and these two non-classical stochastic shortest path problems all coincide in MDPs in which a target state is reached almost surely under all schedulers.

Besides partial and conditional expectations, we also consider the conditional value-at-risk, a risk measure that is defined in terms of a conditional expectation (see also Section 1.1). Given a probability value p, the conditional value-at-risk quantifies the average of the p worst outcomes of a random variable. We investigate the conditional value-at-risk of the random variable of stochastic shortest path problems, the accumulated weight before reaching a target state. The optimization of the conditional value-at-risk then constitutes a problem that is closely related to our two non-classical stochastic shortest path problems.

Outline. After summarizing the main techniques used for the solution of the classical stochastic shortest path problem (Section 3.1), we illustrate the non-classical variants with examples and show that these problems on MDPs with arbitrary integer weights impose new challenges compared to the classical setting and also compared to the setting with non-negative weights: The optimal values can be irrational and infinite-memory schedulers can be necessary to obtain the optimal value (Section 3.2). Nevertheless, we are able to employ techniques from the classical setting to prove that we can decide finiteness of optimal values in polynomial time. Furthermore, the threshold problems for partial and conditional expectations turn out to be easily inter-reducible demonstrating the close relationship between the two problems (Section 3.3). Despite the complications in the setting of integer weights, we can show that there are always optimal schedulers if the optimal values are finite and that the optimal scheduler can be chosen to be weight-based deterministic (Section 3.4). In the setting with non-negative weights, we can refine the results for stochastic multi-player games of [CFK$^+$13a] for MDPs. We show that the least possible saturation point that in particular does not rely on estimations of upper bounds as the saturation point in [CFK$^+$13a] can be computed efficiently (Section 3.5). Finally, we show that also the optimal conditional value-at-risk for the accumulated weight before reaching the goal can be computed in exponential time in MDPs with non-negative weights via a simple saturation point (Section 3.6).

Related work. We briefly discuss closely related work: Previous work on partial or conditional expected accumulated weights was restricted to the case of non-negative weights. More precisely, partial expectations have been studied in the setting of stochastic multiplayer games with non-negative weights [CFK$^+$13a]. Conditional expectations in MDPs with non-negative weights have been addressed in [BKKW17]. In both cases, optimal values are achieved by weight-based deterministic schedulers that depend on the current state and the weight that has been accumulated so far, while memoryless schedulers are not sufficient. Both [CFK$^+$13a] and [BKKW17] prove the existence of a saturation point for the accumulated weight from which on optimal schedulers behave memorylessly and maximize the probability to reach a goal state. This yields exponential-time algorithms for computing optimal schedulers. Moreover, [BKKW17] proves that the threshold problem for conditional expectations ("does there exist a scheduler $\mathfrak{S}$ such that the conditional expectation under $\mathfrak{S}$ exceeds a given threshold?") is PSPACE-hard even for acyclic MDPs.

The optimization of the conditional value-at-risk in MDPs has been investigated in [KM18] for weighted reachability and mean payoffs. For weighted reachability, which can be seen as a special case of accumulated weight before reaching the goal, the optimal conditional value-at-risk is shown to be computable in polynomial time. Furthermore, [KM18] analyzes the simultaneous satisfaction of constraints on the conditional value-at-risk, the value-at-risk, and the expected value.

Note on the publication of the results. The results presented in this chapter have been published in joint work with Christel Baier. Most results appear in [PB19] published at FoSSaCS 2019. The lower bound for the threshold problem for partial expectations and the results on the conditional value-at-risk are part of the publication [PB20] at ICALP 2020.

3.1 Classical stochastic shortest path problem

Before we move to the non-classical setting, we take a closer look at the classical stochastic shortest path problem and its solution. Some concepts and techniques used for this solution will be important to the treatment of the non-classical problems.

Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with a weight function $wgt\colon S \times Act \to \mathbb{Z}$ and a set $Goal \subseteq S$ of designated trap states. As we are interested in the accumulated weight before the set of goal states $Goal$ is reached, it is not important how a run continues after $Goal$ is reached. Hence, we always assume all states in $Goal$ to be trap states and further that all states are reachable from the initial state. We define the following random variable $\oplus goal$ on maximal paths ζ of $\mathcal{M}$ as follows:

$$\oplus Goal(\zeta) = \begin{cases} wgt(\zeta) & \text{if } \zeta \vDash \Diamond Goal, \\ undefined & \text{otherwise.} \end{cases}$$

The expected accumulated weight before reaching $Goal$ under a scheduler $\mathfrak{S}$ is given by the expected value $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal)$. It is evident that this expected value is only defined if $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal) = 1$. The *classical stochastic shortest path problem* asks for the optimal value

$$\mathbb{E}^{\max}_{\mathcal{M},s_{init}}(\oplus Goal) = \sup_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal)$$

where the supremum ranges over all schedulers $\mathfrak{S}$ with $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal) = 1$. We will sketch the treatment of the maximization problem here. In the setting of integer weights, the minimization problem can be turned into a maximization problem by multiplying all weights with -1.

In [BT91], a sufficient condition is given under which the problem can be solved by a linear program. For the restriction to non-negative or non-positive weights, [dA99] provides solutions. The solution without further restrictions of the problem requires a classification of end components that is provided in [BBD$^+$18]. We give a brief overview over this solution in the sequel and start with a central definition.

Definition 3.1 (see [BBD$^+$18])**.** We call an end component $\mathcal{E}$ *positively weight-divergent* if there is a scheduler $\mathfrak{S}$ for $\mathcal{E}$ such that $\Pr^{\mathfrak{S}}_{\mathcal{E},s}(\Diamond(\text{accumulated weight} \geq n)) = 1$ for all $s \in \mathcal{E}$ and $n \in \mathbb{N}$ where $\Diamond(\text{accumulated weight} \geq n)$ denotes the event that the accumulated weight of a prefix of a path is at least n. ◁

In [BBD$^+$18], it is shown that the existence of positively weight-divergent end components can be decided in polynomial time. The decision procedure distinguishes two types of positively weight-divergent end components: If the maximal expected mean payoff inside an end component $\mathcal{E}$ is positive, of course arbitrarily high weights can be reached almost surely. If the maximal expected mean payoff in $\mathcal{E}$ is 0, however, an additional analysis is necessary. We first move to a smaller end component $\mathcal{E}'$ in which only the state-action pairs that enable us to obtain expected mean payoff 0 are included. In other words, for each of these state-action pairs (s, α), there is a memoryless deterministic scheduler choosing α in s while achieving expected mean payoff 0. These are exactly the actions for which an optimal solution to the linear program for the optimal expected mean payoff presented in Section 2.1.2 leads to an equality in the corresponding constraint. In $\mathcal{E}'$, the expected mean payoff is 0 under all schedulers. For this case, the results of [BBD+18] state that the end component $\mathcal{E}$ is positively weight-divergent iff there is a cycle with positive weight in $\mathcal{E}'$. The procedure leads to the following result:

Theorem 3.2 (see [BBD+18])**.** *Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$, the existence of a positively weight-divergent end component can be detected in polynomial time.*

In [BBD+18], it is shown that the value $\mathbb{E}^{\max}_{\mathcal{M},s_{init}}(\oplus Goal)$ is finite if and only if there is no positively weight-divergent end component. So, finiteness of $\mathbb{E}^{\max}_{\mathcal{M},s_{init}}(\oplus Goal)$ can be decided in polynomial time via the analysis of the end components.

The analysis further shows that in an MDP $\mathcal{M}$ without positively weight-divergent end components, all end components either have negative maximal expected mean payoff or contain an end component in which all cycles have length 0, called a 0*-end component.* The *spider construction* provided in [BBD+18] allows us to remove all 0-end components. We add a small modification to the construction by allowing a scheduler to move to a trap state *fail* from all states in a 0-end component. This will be important later on as it allows us to mimic schedulers that stay in a 0-end component with positive probability by moving to *fail* with the same probability.

Modified spider construction. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP and let $\mathcal{E} = (E, \mathfrak{A})$ be an end component in which all cycles in $\mathcal{E}$ have weight 0. The construction proceeds as follows: Pick a state $e \in E$. For each state $s \in E$, all paths from s to e inside $\mathcal{E}$ have the same weight w_s. This follows from the condition that all cycles have weight 0. For each state $s \in E$, disable all actions in state s. Instead enable one action β_s in each state $s \in E \setminus \{e\}$ leading to state e with probability 1 and weight w_s. For each state $s \in E$ and each action $\alpha \in Act(s) \setminus \mathfrak{A}(s)$, enable a new action $\beta_{s,\alpha}$ in state e. Let the weight of this new action in state e be $wgt(s, \alpha) - w_s$. For each state $t \in S$, let $P(e, \beta_{s,\alpha}, t) = P(s, \alpha, t)$. In this way, taking action α in state s can be mimicked by first moving to e via β_s and then taking $\beta_{s,\alpha}$ in state e. Finally, for each state in E, we add a transition with probability 1 and weight 0 to a new trap state *fail* (this is the modification in contrast to the construction from [BBD+18]). A simple instance of the modified spider construction is illustrated in Figure 3.1.

The spider construction can be applied repeatedly to all 0-end components. Starting from an MDP $\mathcal{M}$ without positively weight divergent end components, we obtain an MDP

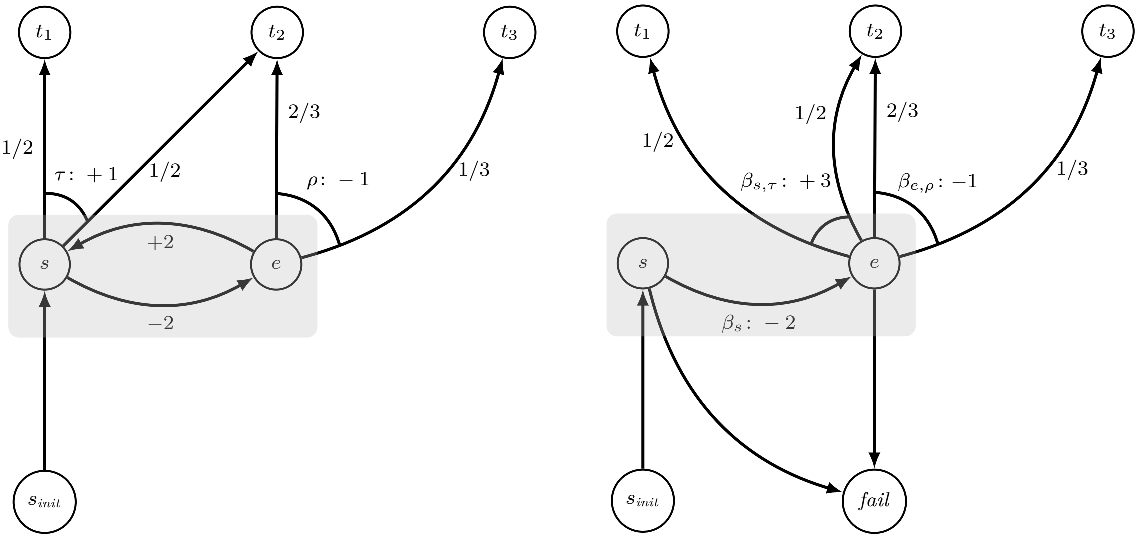


Figure 3.1: Illustration of the spider construction. The gray 0-end component in the left MDP is removed on the right. All transitions leaving the end component are moved to start from state e after the construction. The weights are adjusted accordingly. Additionally, one can move to the new state *fail* from all states of the former 0-end component.

spider($\mathcal{M}$) in which all end components have negative maximal expected mean payoff in polynomial time: Inside each MEC, we identify the existence of 0-end components and identify all state-action pairs that can belong to a 0-end component. By eliminating the end components that are maximal among end components using only these state-action pairs, we remove all 0-end components.

Schedulers for $\mathcal{M}$ can be transferred to schedulers for *spider*($\mathcal{M}$) and vice versa. Namely, moving to *fail* in *spider*($\mathcal{M}$) corresponds to staying in a 0-end component of $\mathcal{M}$ forever. If a 0-end component is left in $\mathcal{M}$ via state-action pair (s, α), the spider construction allows to mimic this by moving to the pivotal point e of the removed 0-end component by the action β_s if s is not already e and by then taking action $\beta_{s,\alpha}$. In the other direction, taking action $\beta_{s,\alpha}$ can be mimicked by moving through the 0-end component in a memoryless fashion until state s is reached and afterwards taking action α there. The adjustment of the weights in the spider construction makes sure that the accumulated weight when leaving the (former) 0-end component is the same in $\mathcal{M}$ and *spider*($\mathcal{M}$). For path properties or random variables that are not affected by the insertion of 0-weight cycles and for which it makes no difference whether a run stays in a 0-end component forever or moves to an absorbing state *fail*, the switch from $\mathcal{M}$ to *spider*($\mathcal{M}$) and the transformation of schedulers as described above does not influence the the satisfaction probability or the expected value, respectively. In particular, this applies to expected accumulated weights before reaching a target state as in the stochastic shortest path problems.

In conclusion, the classification of end components and the spider construction from [BBD$^+$18] combined with the results from [BT91] allow us to solve the classical stochastic shortest path problem in polynomial time: The MDP *spider*$(\mathcal{M})$ obtained by applying the spider construction to all 0-end components of an MDP $\mathcal{M}$ without positively weight-divergent end components satisfies the conditions provided in [BT91]. The classical stochastic shortest path problem can then be solved via a linear program. From a solution to the linear program, an optimal memoryless deterministic scheduler can be derived. This leads to the following result.

Theorem 3.3 (see [BT91, dA99, BBD$^+$18])**.** *The classical stochastic shortest path problem is solvable in polynomial time. If the optimal value is finite, there is an optimal memoryless deterministic scheduler.*

3.2 Two non-classical stochastic shortest path problems

In order to compute expected accumulated weights before reaching a goal state under schedulers that do not reach the goal almost surely, we now have to specify how to treat paths not reaching the goal. In this section, we formally define the two variants we investigate, partial and conditional expectations. Afterwards, we make first observations showing that the situation becomes much more complicated than in the classical setting.

3.2.1 Definition of the partial stochastic shortest path problem

First, consider the following illustrating example.

Example 3.4. The folk dice game “Dice 10,000”, also called “Farkle” or “Zilch” among others, employs the following basic principle: The active player starts a turn by rolling 6 dice. After each roll of dice, she has to put aside at least one scoring die and collects points by doing so. Scoring dice are 1s, scoring 100 points, and 5s, scoring 50 points. We disregard scoring combinations of several dice used in these games for the sake of simplicity here. After putting away some or all of the scoring dice, the player can roll the remaining dice again; or if all dice have been put aside, she can roll all six dice again. Instead of rolling again, the player can also end the turn and receive all points collected during the move. If the player rolls the dice and no scoring die comes up, the turn ends automatically and the player receives no points.

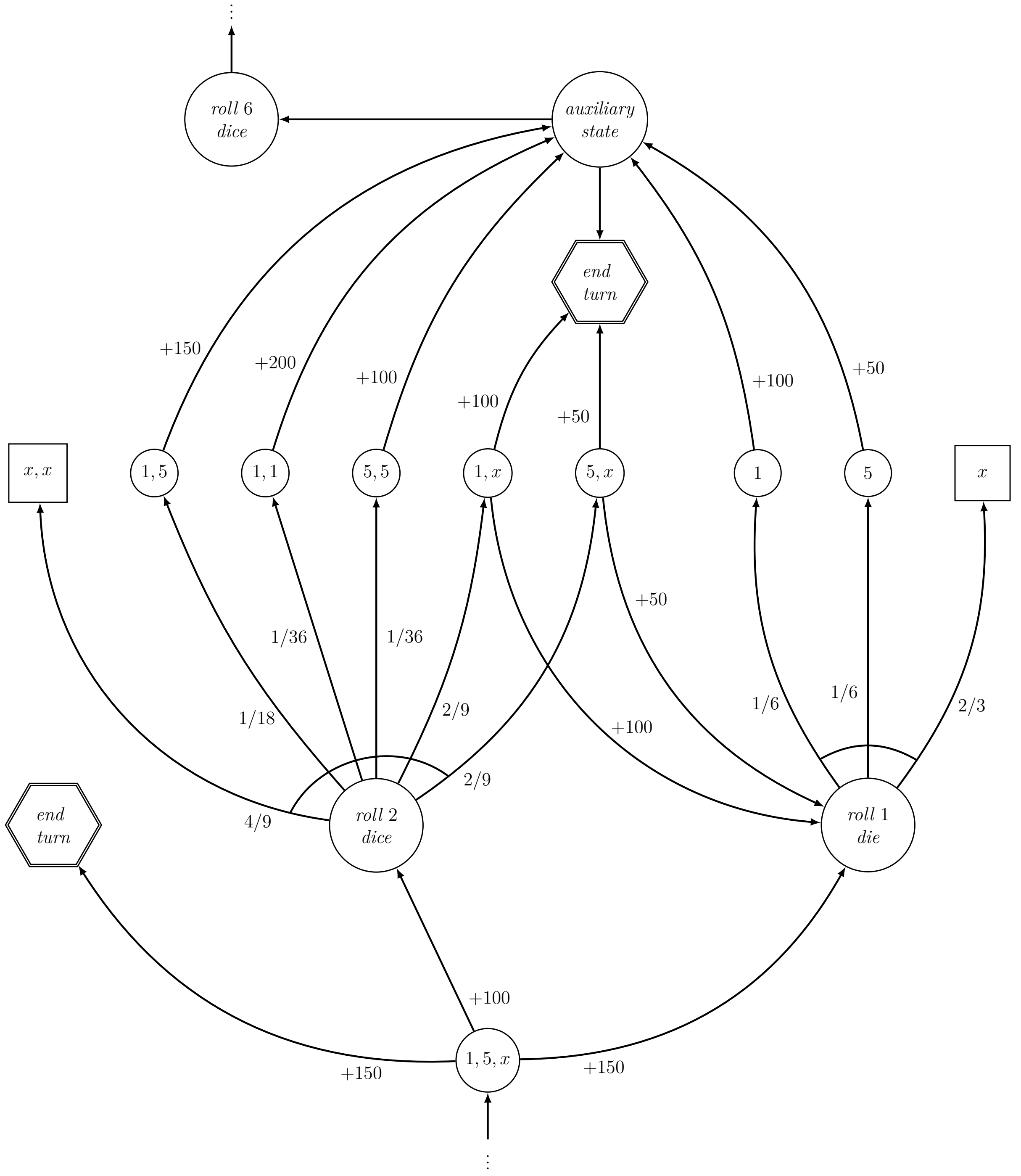


Figure 3.2: An excerpt of an MDP $\mathcal{D}$ modelling one turn in the dice game “Dice 10,000”.

We can construct an MDP modelling one turn of this game. An excerpt of this MDP $\mathcal{D}$ with a weight function *wgt* representing the points that are accumulated after each decision is depicted in Figure 3.2. This excerpt represents the situation in which the player rolled three dice and the outcome was $1, 5, x$ where x stands for any outcome other than 1 and 5. We do not include moves that are obviously suboptimal such as putting aside only the 5 instead of the 1 before rolling two dice again. The hexagonal states are reached when the player decides to end the turn and to collect points, whereas the square states are reached when no scoring die comes up in a roll and the turn hence ends with no points for the player. The initial state of the process is the state "*roll* 6 *dice*" which can be reached again in the excerpt depicted when all remaining dice are scoring in a roll and the player decides to continue the turn by rolling all 6 dice again.

If we want to investigate how to play optimally in this game, we might in particular be interested in how we can maximize the expected number of points collected in one turn and how we can obtain that expectation. In our model $\mathcal{D}$, we hence want to find a scheduler maximizing the expected accumulated weight before reaching "*end turn*" while unsuccessful moves not reaching this target get weight 0. More formally, we can define the random variable *points* on maximal runs ζ in $\mathcal{D}$ as follows:

$$\mathit{points}(\zeta) = \begin{cases} \mathit{wgt}(\rho) & \text{if } \rho \vDash \Diamond \mathit{end\ turn}, \\ 0 & \text{otherwise.} \end{cases}$$

The value of interest is the maximal expected value $\mathbb{E}^{\max}_{\mathcal{D},\mathit{roll}\ 6\ \mathit{dice}}(\mathit{points})$ of this random variable. Note that it is impossible to reach a target state with probability 1 as already the first roll of 6 dice can fail to score any points. For this reason, this optimization problem cannot be solved by employing the classical stochastic shortest path problem.

Further, on an intuitive level, we can already observe that the weight collected so far is important for the decision to be made: If we arrive at the state "$1, 5, x$" after having put aside only three 5s so far for 150 points, it seems reasonable to continue to roll one or two dice for the chance to collect additional points and to possibly be allowed to continue with all 6 dice again. If, on the other hand, we have already cleared all 6 dice several times and arrive with 2000 collected points at the state "$1, 5, x$", the risk of loosing all points when rolling again, which is at least 4/9, is most likely not worth the chance to collect more points in the future. In this situation, we would expect also a precise analysis to conclude that ending the turn is the best choice. ◁

Formally, we define the partial stochastic shortest path problem as follows.

Definition 3.5 (Partial stochastic shortest path problem)**.** Consider an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$. We define the random variable $\oplus \mathit{Goal}$ on maximal paths ζ of

$\mathcal{M}$ by

$$\oplus Goal(\zeta) = \begin{cases} wgt(\zeta) & \text{if } \zeta \vDash \Diamond Goal, \\ 0 & \text{otherwise.} \end{cases}$$

We call the expectation $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal)$ of this random variable under a scheduler $\mathfrak{S}$ the *partial expectation* of $\mathfrak{S}$ and denote it also by $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$. The set *Goal* is specified in the signature of the MDP and hence not included in the notation. The maximal partial expectation $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ is the supremum $\sup_{\mathfrak{S}} \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$ over all schedulers. The minimal partial expectation is defined analogously. We refer to the task to maximize or minimize the partial expectation as the *partial stochastic shortest path problem.* ◁

3.2.2 Definition of the conditional stochastic shortest path problem

Again, we illustrate the problem with an example.

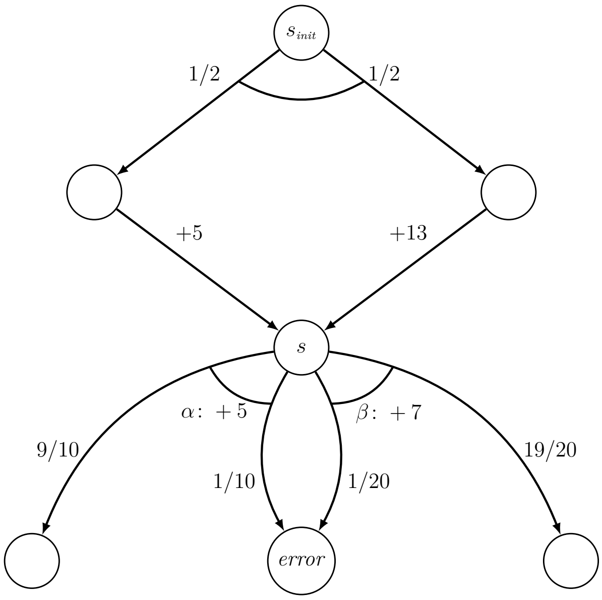


Figure 3.3: A model of a probabilistic system in which an error can occur.

Example 3.6. Consider the MDP depicted in Figure 3.3 modeling the behavior of a probabilistic system. In this system, a certain error scenario occurs with positive probability. If the error occurs a repair mechanism has to reset different components that have been in use. The cost of this reset depends on the run of the system so far and is modeled by the weights that are accumulated. The weights could express the number of components that have been in use and have to be reset in case of the error for instance.

The system arrives at state s with accumulated weight $+5$ or $+13$ with probability $1/2$ each. Then, a decision has to be made whether to choose action α or β. Action α has weight $+5$ and leads to the error with probability $1/10$. Action β leads to the error with probability $1/20$ and weight $+7$. We want to determine how the choice in state s should be made such that the expected costs of the repair mechanism is as low as possible in case the error occurs.

There are four deterministic schedulers. We call these schedulers $\mathfrak{S}_{\alpha\alpha}$, $\mathfrak{S}_{\alpha\beta}$, $\mathfrak{S}_{\beta\alpha}$, and $\mathfrak{S}_{\beta\beta}$. The first subscript indicates which action the schedulers choose when arriving in state s with weight $+5$ and the second index indicates the action for weight $+13$. Under each of the schedulers, there are two paths to the error state. We compute the average cost of these paths for each scheduler weighted with the respective probabilities in the following table:

Scheduler	*wgt* in s	probability of error	cost of error	average cost of error
$\mathfrak{S}_{\alpha\alpha}$	5	$1/2 \cdot 1/10$	10	
	13	$1/2 \cdot 1/10$	18	14
$\mathfrak{S}_{\alpha\beta}$	5	$1/2 \cdot 1/10$	10	
	13	$1/2 \cdot 1/20$	20	$40/3 \approx 13.3$
$\mathfrak{S}_{\beta\alpha}$	5	$1/2 \cdot 1/20$	12	
	13	$1/2 \cdot 1/10$	18	16
$\mathfrak{S}_{\beta\beta}$	5	$1/2 \cdot 1/20$	12	
	13	$1/2 \cdot 1/20$	20	16

The average costs of the paths leading to the error state weighted by their respective probabilities is precisely the conditional expected cost of the repair mechanism under the condition that an error occurs. We observe that memoryless schedulers are not sufficient to obtain the optimal conditional expectation. The interplay between the probability to reach the error state and the weight that is accumulated make more complicated schedulers necessary for the optimization of conditional expectations. Again, we already see that the weight accumulated when reaching state s is important for the decision to be made. ◁

The formal definition of the conditional stochastic shortest path problem now goes as follows:

Definition 3.7 (Conditional stochastic shortest path problem)**.** Consider an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$. By *conditional expectation* of a scheduler $\mathfrak{S}$, we refer to the conditional expectation $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\oplus Goal|\lozenge Goal)$ of the random variable $\oplus Goal$ (defined in Definition 3.5) under the condition that *Goal* is reached. We denote the conditional expectation of a scheduler $\mathfrak{S}$ by $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$. The conditional expectation is well-defined for all schedulers reaching *Goal* with positive probability. The maximal conditional expectation $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ is the supremum $\sup_{\mathfrak{S}} \mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$ over all schedulers that reach *Goal* with positive probability. The minimal conditional expectation is defined analogously. We call the task to maximize or minimize the conditional expectation the *conditional stochastic shortest path problem.* ◁

3.2.3 First observations in the non-classical setting

To start our analysis, we observe that the computation of partial and conditional expectations in Markov chains is easy.

Proposition 3.8. *Let $\mathcal{M} = (S, P, s_{init}, wgt, Goal)$ be a Markov chain. The partial expectation $\mathbb{PE}_{\mathcal{M},s_{init}}$ and the conditional expectation $\mathbb{CE}_{\mathcal{M},s_{init}}$ can be computed in polynomial time.*

Proof. In Markov chains, there is no choice between different actions in any state and hence we assume that the weight function is a map from S to $\mathbb{Z}$. We first collapse the set *Goal* to one goal state *goal* and all states from which *goal* is not reachable to one state *fail*. Note that *goal* or *fail* is reached with probability 1. For each state s of $\mathcal{M}$, we can compute the probability $p_s \overset{\text{def}}{=} \Pr_{\mathcal{M},s}(\lozenge goal)$ in polynomial time. Now, we define a new weight function wgt' on all states $s \in S \setminus \{goal, fail\}$ by

$$wgt'(s) \overset{\text{def}}{=} wgt(s) \cdot p_s.$$

Let $\mathcal{M}'$ be the Markov chain equipped with the new weight function wgt'. Let f_s be the expected number of visits in $\mathcal{M}$ to the state s for all $s \in S \setminus \{goal, fail\}$ Recall that $\oplus\{goal, fail\}$ is the random variable from the classical stochastic shortest path problem that assigns the accumulated weight before reaching *goal* or *fail* to a path. For each visit to a state s, the weight $wgt(s)$ contributes to $\mathbb{PE}_{\mathcal{M},s_{init}}$ with probability p_s and there are f_s visits to the state in expectation. So, the partial expectation can be expressed as follows:

$$\mathbb{PE}_{\mathcal{M},s_{init}} = \sum_{s \in S \setminus \{goal, fail\}} f_s \cdot wgt(s) \cdot p_s = \mathbb{E}_{\mathcal{M}',s_{init}}(\oplus\{goal, fail\}).$$

Hence, the partial expectation $\mathbb{PE}_{\mathcal{M},s_{init}}$ can be computed in polynomial time.

The conditional expectation $\mathbb{CE}_{\mathcal{M},s_{init}}$ is given by $\mathbb{PE}_{\mathcal{M},s_{init}}/p_{s_{init}}$. If $p_{s_{init}} = 0$, the conditional expectation is undefined and otherwise, it is computable in polynomial time as well. □

The optimization of partial and conditional expectations in MDPs, however, is considerably more difficult than the classical stochastic shortest path problem. While optimal schedulers can be chosen memoryless and the optimal values are rational for the classical problem, these results do not hold for the partial and conditional stochastic shortest path problem as we will see in this section by virtue of several examples used in the proofs.

Theorem 3.9. *The optimal partial expectation and the optimal conditional expectation can be irrational.*

Proof. Consider the MDP $\mathcal{M}$ depicted on the left in Figure 3.4. In the initial state s_{init}, two actions are enabled. Action τ leads to *Goal* with probability 1 and weight 0. Action σ leads to the states s and t with probability 1/2 from where we will return to s_{init} with weight -2 or $+1$, respectively. The scheduler choosing τ immediately leads to an expected weight of 0 and is optimal among schedulers reaching the goal almost surely. As long as we choose σ in s_{init}, the accumulated weight follows an asymmetric random walk increasing by 1 or decreasing by 2 with probability 1/2 before we return to s_{init}. The probability p to ever reach weight $+1$ in this asymmetric random walk satisfies

$$p = \frac{1}{2} + \frac{1}{2}p^3.$$

The reason is that weight $+1$ is reached either with probability 1/2 directly or if it is reached three times with probability p each after weight -2 has been collected with probability 1/2. The only solution of this equation in the open interval $(0,1)$ is $1/\Phi$ where $\Phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Likewise, ever reaching accumulated weight n has probability $1/\Phi^n$ for all $n \in \mathbb{N}$. Consider the scheduler $\mathfrak{S}_k$ choosing τ as soon as the accumulated weight reaches k in s_{init}. Its partial expectation is k/Φ^k as the paths which never reach weight k are assigned weight 0. The maximum is reached at $k = 2$. In Section 3.4, we prove that there are optimal schedulers that are deterministic and weight-based. With this result we can conclude that the maximal partial expectation is indeed $2/\Phi^2$, an irrational number.

The conditional expectation of $\mathfrak{S}_k$ in $\mathcal{M}$ is k as $\mathfrak{S}_k$ reaches the goal with accumulated weight k if it reaches the goal. So, the conditional expectation is not bounded. If we add a new initial state making sure that the goal is reached with positive probability as in the MDP $\mathcal{N}$, we can obtain an irrational maximal conditional expectation as well: The scheduler $\mathfrak{T}_k$ choosing τ in c as soon as the weight reaches k has conditional expectation

$$\frac{k/2\Phi^k}{1/2 + 1/2\Phi^k}.$$

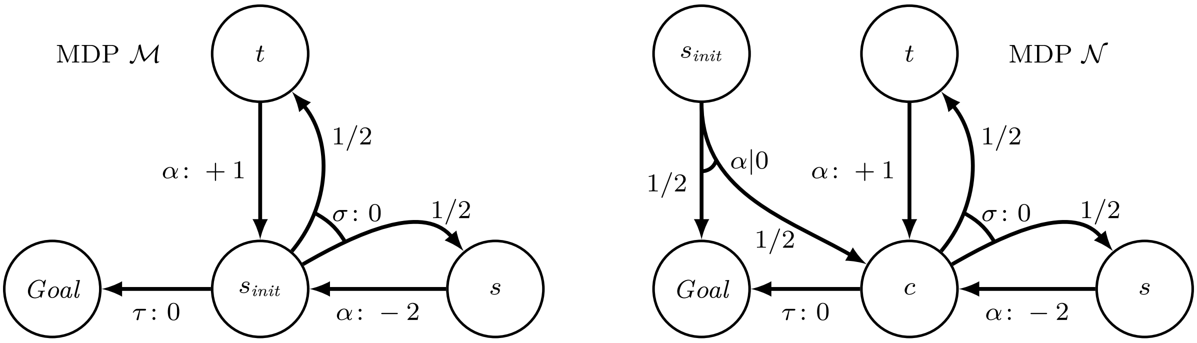


Figure 3.4: Example MDPs with irrational maximal partial and conditional expectation, respectively.

The maximum is obtained for $k = 3$; the maximal conditional expectation is

$$\frac{3/\Phi^3}{1+1/\Phi^3} = \frac{3}{3+\sqrt{5}}. \qquad \square$$

In the context of solvency games, a restricted form of weighted MDPs, Berger et al. call a scheduler that makes the same decisions in each state if the state is reached with an accumulated weight above a certain bound a "rich person's strategy" [BKSV08]. Such a scheduler has a rather simple structure although it requires infinite memory in general. For the non-classical stochastic shortest path problem, optimal schedulers might not only require infinite memory but also a more complicated structure:

Theorem 3.10. *There is an MDP $\mathcal{M}$ in which any scheduler maximizing the partial expectation requires infinite memory. Furthermore, there is no optimal "rich person's strategy" in $\mathcal{M}$.*

Proof. Let us first consider the MDP $\mathcal{N}$ depicted in Figure 3.5. Let π be a path reaching t for the first time with accumulated weight r. Consider a scheduler which chooses β for the first k times and then α. In this situation, the partial expectation from this point on is:

$$\frac{1}{2^{k+1}}(r-k)+\sum_{i=1}^{k}\frac{1}{2^i}(r-i) = \frac{1}{2^{k+1}}+\sum_{i=1}^{k+1}\frac{1}{2^i}(r-i) = \frac{k-r+4}{2^{k+1}}+r-2.$$

For $r \geq 2$, this partial expectation has its unique maximum for the choice $k = r-2$. This already shows that an optimal scheduler needs infinite memory. No matter how much weight r has been accumulated when reaching t, the optimal scheduler has to count the $r-2$ times it chooses β.

Furthermore, we can transfer the optimal scheduler for the MDP $\mathcal{N}$ to the MDP $\mathcal{M}$. In state t, we have to make a nondeterministic choice between two action leading to the states q_0 and q_1, respectively. In both of these states, action β is enabled which behaves

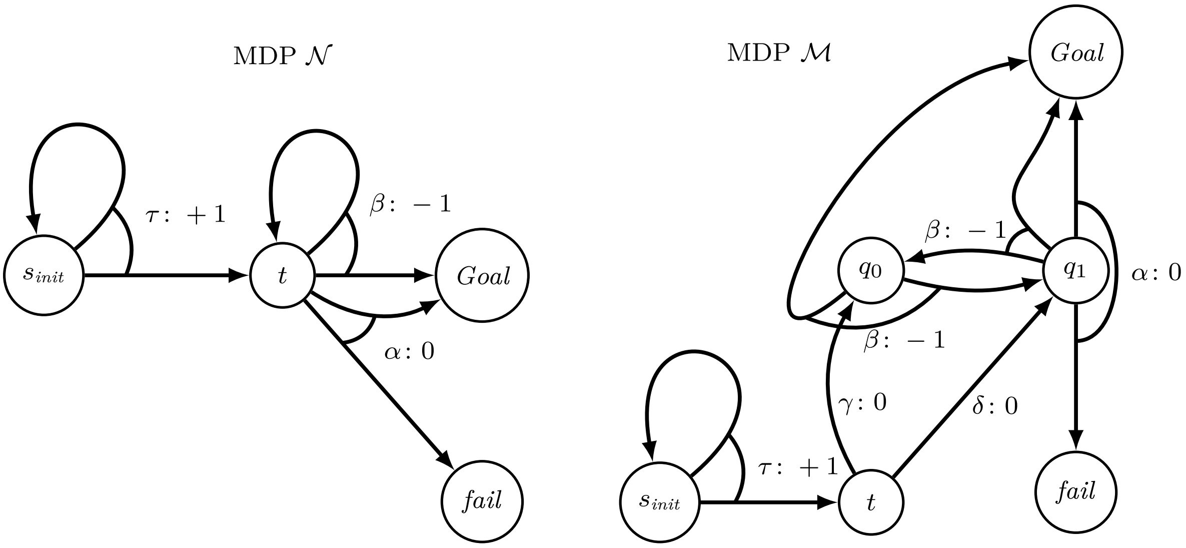


Figure 3.5: An MDP $\mathcal{M}$ in which the optimal choice to maximize the partial expectation in state t depends on the parity of the accumulated weight. All non-trivial transition probabilities are $1/2$. The MDP $\mathcal{N}$ serves as an auxiliary step in the argument.

like the same action in the MDP $\mathcal{N}$ except that it moves between the two states if *Goal* is not reached. So, the action α is only enabled every other step. As in $\mathcal{N}$, we want to choose α after choosing β exactly $r-2$ times if we arrived in t with accumulated weight $r \geq 2$. So, the choice in t depends on the parity of r: For $r = 1$ or r even, we choose δ. For odd $r \geq 3$, we choose γ. This shows that the optimal scheduler in the MDP $\mathcal{M}$ needs specific information about the accumulated weight, in this case the parity, no matter how much weight has been accumulated. □

In the proof, the optimal scheduler has a periodic behavior when fixing a state and looking at optimal decisions for increasing values of accumulated weight. The question whether an optimal scheduler always has such a periodic behavior remains open. In Chapter 5, we will see that this question is related to deep questions about the behavior of linear recurrence sequences.

3.3 Basic results

While we have seen that the non-classical problems pose difficult challenges, we will rely on techniques known from the classical stochastic shortest path problem to decide finiteness and simplify the MDPs under investigation in the sequel. Furthermore, we will uncover the close relationship between partial and conditional expectation by showing that the associated decision problems, the threshold problems, are inter-reducible.

3.3.1 Deciding finiteness and preprocessing

Finiteness can be decided in a fashion similar to the classical setting:

Proposition 3.11. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP. Collapse all states from which Goal is not reachable to a trap state fail. The optimal partial expectation $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ is finite if and only if there are no positively weight-divergent end components in $\mathcal{M}$.*

Proof. Suppose there is a positively weight-divergent end component $\mathcal{E}$. Since $\mathcal{E}$ is reachable and we can accumulated arbitrarily high weights inside $\mathcal{E}$ with probability 1, we can easily construct a sequence of schedulers whose partial expectation diverges to $+\infty$ by letting the schedulers stay in a positively weight divergent end component until an arbitrarily high weight has been accumulated, before they leave the end component and reach *Goal* with positive probability.

Now, suppose that there are no positively weight-divergent end components. So, for each end component $\mathcal{E}$, there is a number $W_{\mathcal{E}}$ and a probability $p_{\mathcal{E}}$ such that in $\mathcal{E}$ we have $\max_{s\in\mathcal{E}} \Pr^{\max}_{\mathcal{E},s}(\Diamond(wgt \geq W_{\mathcal{E}})) < 1$. On the other hand, in the MEC-quotient $MEC(\mathcal{M})$ of $\mathcal{M}$ the probability to reach *Goal* or $fail$ in $|S|$ steps is at least $\delta^{|S|}$ where δ is the minimal transition probability. Let $M \overset{\text{def}}{=} \max_{s,\alpha} |wgt(s,\alpha)|$. Then we can conclude that

$$\max_{s} \Pr^{\max}_{MEC(\mathcal{M}),s}(\Diamond(wgt > M \cdot |S|)) \leq 1 - \delta^{|S|}.$$

All in all, it is impossible for a scheduler to almost surely reach an accumulated weight above $M \cdot |S| + \sum_{\mathcal{E} \text{ is an end component}} W_{\mathcal{E}}$. Therefore, there is a natural number W such that $\max_s \Pr^{\max}_{s}(\Diamond(wgt \geq W)) < 1$. Call this probability p. For all $n \in \mathbb{N}$ we get that $\max_{s\in S} \Pr^{\max}_{\mathcal{M},s}(\Diamond(wgt \geq n\cdot W + M)) \leq p^n$. Hence, the partial expectation of any scheduler is bounded by the following upper bound on the expected accumulated weight

$$\sum_{n=0}^{\infty} (n+1) \cdot W \cdot p^n = \frac{W}{(1-p)^2}. \qquad \square$$

For the finiteness of the maximal conditional expectation, we obtain the following immediate consequence.

Corollary 3.12. *Let $\mathcal{M}$ be as in Proposition 3.11 and assume that $\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond Goal) > 0$. Then, $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ is finite if there are no positively weight-divergent end components in $\mathcal{M}$.*

Proof. The maximal value $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ is at most $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}/\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond Goal)$. □

To check the finiteness of the maximal conditional expectation, we need an additional condition if the minimal probability to reach *Goal* is 0. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$

be as in Proposition 3.11 and assume that $\Pr^{\min}_{\mathcal{M},s_{\mathit{init}}}(\Diamond \mathit{Goal}) = 0$. For each state s, we define the subset $\mathit{Act}^{\min}(s)$ as the set of actions $\alpha \in \mathit{Act}(s)$ with

$$\Pr^{\min}_{\mathcal{M},s}(\Diamond \mathit{Goal}) = \sum_{t \in S} P(s,\alpha,t) \cdot \Pr^{\min}_{\mathcal{M},t}(\Diamond \mathit{Goal}).$$

An action in $\mathit{Act}^{\min}(s)$ hence allows us to reach *Goal* with the minimal probability that is possible from s. Further, let $S_0 \subseteq S$ be the set of states that are reachable from s_{init} within $(S, \mathit{Act}^{\min})$. These states are reachable by a scheduler that avoids *Goal* with probability 1. The condition to check finiteness is given in the following proposition. This condition is equivalent to the condition given in [BKKW17].

Proposition 3.13 (see also [BKKW17]). *Let $\mathcal{M}$, $\mathit{Act}^{\min}$ and S_0 be as described above. The maximal conditional expectation in $\mathcal{M}$ is finite if and only if there are no positively weight-divergent end components in $\mathcal{M}$ and no positive cycles in $(S_0, \mathit{Act}^{\min})$.*

Proof. If there are positively weight-divergent end components, the maximal partial expectation and hence the maximal conditional expectation are unbounded. If there is a positive cycle in $(S_0, \mathit{Act}^{\min})$, let s be a state in this cycle and fix a memoryless scheduler $\mathfrak{F}$ that reaches *Goal* with positive probability $p > 0$ from s. Consider the following sequence of schedulers $\mathfrak{S}_n$: The scheduler $\mathfrak{S}_n$ tries to reach s inside $(S_0, \mathit{Act}^{\min})$ and take the positive cycle until the accumulated weight exceeds n in state s. This happens with positive probability q_n. If it never happens, $\mathfrak{S}_n$ stays inside $(S_0, \mathit{Act}^{\min})$ and never reaches *Goal*. Otherwise, it switches to the behavior of $\mathfrak{F}$. We can see that the conditional expectation of $\mathfrak{S}_n$ satisfies

$$\mathbb{CE}^{\mathfrak{S}_n}_{\mathcal{M},s_{\mathit{init}}} \geq \frac{q_n \cdot p \cdot n + q_n \cdot \mathbb{PE}^{\mathfrak{F}}_{\mathcal{M},s}}{q_n \cdot p}.$$

This expression diverges to ∞ for $n \to \infty$. Hence, the conditional expectation in $\mathcal{M}$ is not bounded from above.

Now assume that $(S_0, \mathit{Act}^{\min})$ does not contain positive cycles and that $\mathcal{M}$ does not contain positively weight-divergent end components. So, there is a unique maximal weight w_s of paths leading from s_{init} to s in $(S_0, \mathit{Act}^{\min})$ for each state $s \in S_0$. Consider the following MDP $\mathcal{N}$: It contains the MDP $\mathcal{M}$ and a new initial state t_{init}. For each $s \in S_0$ and each $\alpha \in \mathit{Act}(s) \setminus \mathit{Act}^{\min}(s)$, $\mathcal{N}$, a new action $\beta_{s,\alpha}$ is enabled in t_{init}. The action $\beta_{s,\alpha}$ has weight $w_s + \mathit{wgt}(s,\alpha)$ and the same probability distribution over successors as action α in state s. In this way, we ensure that $\Pr^{\min}_{\mathcal{N},t_{\mathit{init}}}(\Diamond \mathit{Goal}) > 0$ as, intuitively speaking, a scheduler has to choose immediately how to leave $(S_0, \mathit{Act}^{\min})$. We claim that the maximal conditional expectations in $\mathcal{M}$ and $\mathcal{N}$ are equal.

For each pair (s,α) with $s \in S_0$ and $\alpha \in \mathit{Act}(s) \setminus \mathit{Act}^{\min}(s)$, let $c_{s,\alpha} \overset{\text{def}}{=} \sup_{\mathfrak{T}} \mathbb{CE}^{\mathfrak{T}}_{\mathcal{N},t_{\mathit{init}}}$ where the supremum ranges over all schedulers choosing $\beta_{s,\alpha}$ with probability 1 in the first step. By Corollary 3.12, this value is finite. We claim that

$$\mathbb{CE}^{\max}_{\mathcal{N},t_{init}} = \max\{c_{s,\alpha} | s \in S_0 \text{ and } \alpha \in Act(s) \setminus Act^{\min}(s)\} \overset{\text{def}}{=} c.$$

Clearly, $\mathbb{CE}^{\max}_{\mathcal{N},t_{init}} \geq c$. On the other hand, consider a scheduler $\mathfrak{S}$ that chooses actions $\beta_{s,\alpha}$ with probability $p_{s,\alpha}$ and reaches the goal afterwards with probability $q_{s,\alpha}$ while obtaining a partial expectation of $E_{s,\alpha}$ if action $\beta_{s,\alpha}$ is taken first. As the conditional expectation when $\beta_{s,\alpha}$ is chosen is at most $c_{s,\alpha}$, we get that $E_{s,\alpha} \leq q_{s,\alpha} \cdot c_{s,\alpha}$ for all suitable pairs (s,α). We get that

$$\mathbb{CE}^{\mathfrak{S}}_{\mathcal{N},t_{init}} = \frac{\sum_{s,\alpha} p_{s,\alpha} \cdot E_{s,\alpha}}{\sum_{s,\alpha} p_{s,\alpha} \cdot q_{s,\alpha}} \leq \frac{\sum_{s,\alpha} p_{s,\alpha} \cdot (q_{s,\alpha} \cdot c_{s,\alpha})}{\sum_{s,\alpha} p_{s,\alpha} \cdot q_{s,\alpha}} \leq c.$$

We now prove that $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} = c$. A scheduler reaching *Goal* with positive probability in $\mathcal{M}$ has to choose an action not in $Act^{\min}$ after at least one path. Let $s \in S_0$ and $\alpha \in Act(t) \setminus Act^{\min}(s)$ be such that $c = c_{s,\alpha}$. For any scheduler $\mathfrak{T}$ for $\mathcal{N}$ starting with $\beta_{s,\alpha}$, we define the following scheduler $\mathfrak{T}'$ for $\mathcal{M}$: The scheduler $\mathfrak{T}'$ starts by following a path with maximal accumulated weight from s_{init} to s. If it reaches s with accumulated weight w_s it chooses α and follows the choices of $\mathfrak{T}$ from then on. If it does not reach s with accumulated weight w_s, $\mathfrak{T}'$ only picks actions in $Act^{\min}$ making sure that *Goal* will not be reached. In this way, $\mathbb{CE}^{\mathfrak{T}'}_{\mathcal{M},s_{init}} = \mathbb{CE}^{\mathfrak{T}}_{\mathcal{N},t_{init}}$. So, $\mathbb{CE}^{\sup}_{\mathcal{M},s_{init}} \geq c$.

Before we show the other direction, we define, given a finite path π, a finite path ρ starting in $last(\pi)$, and a scheduler $\mathfrak{Q}$, the scheduler $\mathfrak{Q} \uparrow \pi$ by

$$\mathfrak{Q} \uparrow \pi\,(\rho) := \mathfrak{Q}(\pi;\rho)$$

where $\pi;\rho$ denotes the concatenation of the paths π and ρ.

To show that $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} \leq c$ for any scheduler $\mathfrak{S}$ for $\mathcal{M}$ with $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond Goal) > 0$, let $\mathfrak{S}$ be such a scheduler and consider the set Π of finite $\mathfrak{S}$-paths π in $(S_0, Act^{\min})$ such that $\mathfrak{S}(\pi) \in Act(last(\pi)) \setminus Act^{\min}((last(\pi)))$. We know that for each $\pi \in \Pi$,

$$\frac{wgt(\pi) + \mathbb{PE}^{\mathfrak{S}\uparrow\pi}_{last(\pi)}}{\Pr^{\mathfrak{S}\uparrow\pi}_{last(\pi)}(\Diamond Goal)} \leq \frac{w_{last(\pi)} + \mathbb{PE}^{\mathfrak{S}\uparrow\pi}_{last(\pi)}}{\Pr^{\mathfrak{S}\uparrow\pi}_{last(\pi)}(\Diamond Goal)} \leq c.$$

We conclude that also

$$\mathbb{CE}^{\mathfrak{S}}_{s_{init}} = \frac{\sum_{\pi\in\Pi} \Pr^{\mathfrak{S}}_{s_{init}}(\pi) \cdot (wgt(\pi) + \mathbb{PE}^{\mathfrak{S}\uparrow\pi}_{last(\pi)})}{\sum_{\pi\in\Pi} \Pr^{\mathfrak{S}}_{s_{init}}(\pi) \cdot \Pr^{\mathfrak{S}\uparrow\pi}_{last(\pi)}(\Diamond Goal)} \leq c$$

as all summands in the denominator are positive. □

The proof presented here contains a construction that allows us to assume that *Goal* is reached with positive probability under all schedulers if the maximal conditional expectation is finite. More formally, we obtain the following statement:

Corollary 3.14. *Let $\mathcal{M}$ be as in Proposition 3.11. If $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$, we can construct an MDP $\mathcal{N}$ containing $\mathcal{M}$ and a new initial state t_{init} in polynomial time that satisfies $\mathbb{CE}^{\max}_{\mathcal{N},t_{init}} = \mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ and $\mathrm{Pr}^{\min}_{\mathcal{N},t_{init}}(\Diamond Goal) > 0$.*

While the finiteness of maximal partial or conditional expectations already implies that there are no positively weight-divergent end components, we can transform the MDP further to also remove 0-end components by the spider construction described in Section 3.1:

Proposition 3.15. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP. Let spider($\mathcal{M}$) be the MDP obtained from $\mathcal{M}$ by applying the modified spider construction (see Section 3.1) successively to all 0-end components. For each scheduler $\mathfrak{S}$ for $\mathcal{M}$, we can construct a scheduler $\mathfrak{T}$ for spider($\mathcal{M}$), and vice versa, such that $\mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond Goal) = \mathrm{Pr}^{\mathfrak{T}}_{spider(\mathcal{M}),s_{init}}(\Diamond Goal)$ and $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{PE}^{\mathfrak{T}}_{spider(\mathcal{M}),s_{init}}$.*

Proof. The construction of the schedulers is indicated in Section 3.1. It is clear that the property to reach *Goal* is not affected if a run enters the new trap state *fail* instead of staying in a 0-end component forever. As the probability distribution on how an end component is left, is maintained by the spider construction and by the transfer of schedulers, the probability to reach *Goal* is not affected by the construction. Similarly, the random variable $\oplus Goal$ is not affected if *Goal* is not reached. If *Goal* is reached, the spider construction ensures that the accumulated weight when exiting a 0-end component is not affected. □

If an MDP contains no positively weight-divergent end components, the spider construction can remove all 0-end components in polynomial time as we have seen in Section 3.1. So, after we checked for finiteness in polynomial time, we can also include the spider construction in a pre-processing procedure that still runs in polynomial time. All in all, we can use the results in this section to provide a polynomial-time pre-processing procedure that allows us to make simplifying assumptions in the sequel. Recall that we already assume that all states are reachable from the initial state.

Pre-processing. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP. The following steps can be executed in polynomial time:

1. Collapse all states in *Goal* to a single trap state *goal* and collapse all states that cannot reach *goal* to a trap state *fail*.

2. Check whether the maximal partial expectation is finite (see Proposition 3.11). If the value is infinite, also the maximal conditional expectation is infinite and no further analysis is necessary. Otherwise, we know that there are no positively weight-divergent end components in the MDP.

3. Remove all 0-end components using the modified spider construction. As there are no positively weight-divergent end components, the construction can be executed in polynomial time. Afterwards, all end components have negative maximal expected mean payoff.

4. If interested in the conditional expectation, check whether the maximal value is finite (see Proposition 3.13). If so apply the construction used for Corollary 3.14, to obtain an MDP with the same maximal conditional expectation in which *Goal* is reached with positive probability under all schedulers.

In the sequel, we can now always assume that this pre-processing procedure has been performed on the MDPs under consideration.

3.3.2 Inter-reducibility of threshold problems

The natural decision problem associated with the computation of maximal partial or conditional expectations is the *threshold problem*: Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ and a rational ϑ, the threshold problem asks whether $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} \bowtie \vartheta$ and whether $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} \bowtie \vartheta$, respectively, for $\bowtie \in \{<, \leq, \geq, >\}$. The threshold problem is of special interest to us as we can use this decision problem to prove lower bounds on the complexity of computing maximal partial and conditional expectations. We show that the threshold problems for partial and conditional expectations are equally hard by providing polynomial-time reductions between the problems. Note that the threshold problems concerning the minimal partial or conditional expectation, i.e., the question whether $\mathbb{PE}^{\min}_{\mathcal{M},s_{init}} \bowtie \vartheta$ and whether $\mathbb{CE}^{\min}_{\mathcal{M},s_{init}} \bowtie \vartheta$ can be addressed by multiplying all weights with -1 before considering the maximal values again.

Proposition 3.16. *The threshold problems for partial and conditional expectations are polynomial-time inter-reducible.*

Proof. First, we show that the threshold problem for conditional expectation is reducible to the threshold problem for partial expectations. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP and ϑ a rational number. W.l.o.g., we can assume that $Goal = \{goal\}$ is a singleton and that any scheduler for $\mathcal{M}$ reaches *goal* with positive probability after the pre-processing described in Section 3.3.1. We construct a new MDP $\mathcal{N}$ by adding a new state *goal*$'$ which is the new goal state in $\mathcal{N}$ and a transition with probability 1 from the old goal state *goal* to *goal*$'$ with weight $-\vartheta$. We claim that $\mathbb{CE}^{\max}_{\mathcal{M}} \bowtie \vartheta$ if and only if

$\mathbb{PE}^{\max}_{\mathcal{N}} \bowtie 0$ for $\bowtie \in \{<, \leq, \geq, >\}$. In fact, we show that the claim holds scheduler-wise. Clearly, any scheduler for $\mathcal{M}$ can be seen as a scheduler for $\mathcal{N}$ and vice versa. Let $\mathfrak{S}$ be a scheduler for both MDPs. Then, we have

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} = \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} - \vartheta \cdot \Pr^{\mathfrak{S}}_{\mathcal{M}}(\lozenge goal) \bowtie 0 \text{ iff } \frac{\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}}}{\Pr^{\mathfrak{S}}_{\mathcal{M}}(\lozenge goal)} \bowtie \vartheta \text{ iff } \mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} \bowtie \vartheta.$$

The rational weight $-\vartheta$ can be turned into an integer weight by multiplying all weights in $\mathcal{N}$ with the denominator of ϑ.

In the other direction, let again $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP and ϑ a rational number. Again, we assume that $Goal = \{goal\}$. We construct an MDP $\mathcal{N}$ by adding a new initial state t_{init} and a new goal state $goal'$. In t_{init}, one action leading to s_{init} and $goal'$ with probability 1/2 each and weight 0 is enabled. Further, the process $\mathcal{N}$ moves from $goal$ to $goal'$ with probability 1 and weight $+\vartheta$. Again, each scheduler $\mathfrak{S}$ for $\mathcal{M}$ can also be used for $\mathcal{N}$ and vice versa. For each scheduler, we have that

$$\mathbb{CE}^{\mathfrak{S}}_{\mathcal{N},t_{init}} = \frac{1/2 \cdot (\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} + \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal) \cdot \vartheta)}{1/2 \cdot (1 + \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal))} \bowtie \vartheta \quad \text{iff} \quad \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} \bowtie \vartheta.$$

From the scheduler-wise equivalence, we conclude the equivalence for the maximal partial and conditional expectation, respectively. □

The first reduction presented in the proof introduces the weight $-\vartheta$. If all weights in $\mathcal{M}$ are non-negative, only positive values for ϑ make sense. In this case, $-\vartheta$ is negative and the constructed MDP $\mathcal{N}$ does not have only non-negative weights. We will address the setting with non-negative weights in Section 3.5. There, we will provide a further reduction for the threshold problems on acyclic MDPs with non-negative weights that does not introduce a negative weight (Lemma 3.31). That reduction relies on results on the existence of optimal deterministic schedulers which we obtain in the next Section 3.4.

3.4 Existence of optimal schedulers

We are now going to prove that optimal partial and conditional expectations can be obtained by weight-based deterministic schedulers. After showing that, if finite, the maximal partial expectation $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ can be approximated by weight-based deterministic schedulers, we take an analytic approach. We define a metric on the space of weight-based deterministic schedulers. Under this metric, we obtain a compact space. Then, we prove that the function assigning the partial expectation to weight-based deterministic schedulers is upper semi-continuous. We conclude that there is a weight-based deterministic scheduler obtaining the maximum. The first goal is to prove the following proposition:

Proposition 3.17. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP and assume that $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$. Let WD be the set of weight-based deterministic schedulers for $\mathcal{M}$. We have*

$$\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} = \sup_{\mathfrak{S}\in WD} \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}.$$

Before we start proving this Proposition 3.17, let us show that it implies the analogous result for conditional expectations.

Corollary 3.18. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$. Let WD be the set of weight-based deterministic schedulers for $\mathcal{M}$. We have*

$$\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} = \sup_{\mathfrak{S}\in WD} \mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}.$$

Proof. After the pre-processing procedure, we can assume that $\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond Goal) > 0$. Note that weight-based deterministic schedulers in the pre-processed MDP correspond to weight-based deterministic schedulers in the original MDP. Let ϑ be an arbitrary rational with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$. In the proof of Proposition 3.16, we have reduced the threshold problem whether $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$ to the threshold problem by constructing an MDP $\mathcal{N}$ by adding a terminal weight of $-\vartheta$ on a transition to a new goal state to $\mathcal{M}$. In this MDP $\mathcal{N}$, now $\mathbb{PE}^{\max}_{\mathcal{N},s_{init}} > 0$ as ϑ was chosen such that we have a positive instance of the threshold problem. So, by Proposition 3.17, there is a weight-based deterministic scheduler $\mathfrak{T}$ for $\mathcal{N}$ with $\mathbb{PE}^{\mathfrak{T}}_{\mathcal{N},s_{init}} > 0$. As the equivalence of the threshold problems for conditional expectations in $\mathcal{M}$ and partial expectations in $\mathcal{N}$ in the proof of Proposition 3.16 holds scheduler-wise, we conclude that $\mathbb{CE}^{\mathfrak{T}}_{\mathcal{M},s_{init}} > \vartheta$ when we use $\mathfrak{T}$ as a scheduler for $\mathcal{M}$. As we can choose ϑ arbitrarily close to $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$, the claim follows. □

We now prove Proposition 3.17 in two steps. First, we show that for any scheduler there is an equally good weight-based scheduler (Lemma 3.19). Afterwards, we show that also the use of randomization does not increase the partial expectation that can be obtained (Lemma 3.20).

Lemma 3.19. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$. For each scheduler $\mathfrak{S}$ for $\mathcal{M}$, there is a weight-based scheduler $\mathfrak{T}$ such that*

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{PE}^{\mathfrak{T}}_{\mathcal{M},s_{init}}.$$

Proof. After pre-processing, we can assume that all end-components have negative maximal mean payoff. Let $\mathfrak{S}$ be a scheduler for $\mathcal{M}$. For each non-trap state $s \in S$ and each $w \in \mathbb{Z}$, we let $\vartheta^{\mathfrak{S}}_{s,w}$ be the expected number of times that s is reached with accumulated weight w under $\mathfrak{S}$, and we let $\vartheta^{\mathfrak{S}}_{s,w,\alpha}$ be the expected number of times that α is chosen in this situation by $\mathfrak{S}$. We have that

$$\vartheta^{\mathfrak{S}}_{s,w} = \sum_{\substack{\pi \text{ finite path,} \\ \mathit{last}(\pi)=s, \\ \mathit{wgt}(\pi)=w}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\pi) \text{ and } \vartheta^{\mathfrak{S}}_{s,w,\alpha} = \sum_{\substack{\pi \text{ finite path,} \\ \mathit{last}(\pi)=s, \\ \mathit{wgt}(\pi)=w}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\pi) \cdot \mathfrak{S}(\pi)(\alpha).$$

Note that $\vartheta^{\mathfrak{S}}_{s,w}$ is finite for all non-trap states $s \in S$ and $w \in \mathbb{Z}$ as all end components have negative maximal expected mean payoff. The idea is now to define a weight-based randomized scheduler that visits each state-weight pair with the same expected frequency as $\mathfrak{S}$. The claim then follows directly. To this end, we define a scheduler $\mathfrak{T}$ by

$$\mathfrak{T}(s,w)(\alpha) := \begin{cases} \vartheta^{\mathfrak{S}}_{s,w,\alpha} / \vartheta^{\mathfrak{S}}_{s,w} & \text{if } \vartheta^{\mathfrak{S}}_{s,w} > 0, \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

Clearly, only state-weight pairs (s,w) which are reachable under $\mathfrak{S}$ are reachable under $\mathfrak{T}$. Further, $\mathfrak{T}$ is well-defined as $\sum_{\alpha \in \mathit{Act}(s)} \mathfrak{T}(s,w)(\alpha) = 1$ for all reachable (s,w).

For each state-weight pair (s,w), let $\vartheta^{\mathfrak{T}}_{s,w}$ be the expected number of times that (s,w) is reached under $\mathfrak{T}$. Then, the collection of all $\vartheta^{\mathfrak{T}}_{s,w}$ is the unique component-wise least non-negative solution to the following set of equations: For all (s,w),

$$x_{s,w} = i_{s,w} + \sum_{t \in S, \alpha \in \mathit{Act}(t)} P(t,\alpha,s) \cdot x_{t,w-\mathit{wgt}(t,\alpha)} \cdot \mathfrak{T}(t, w - \mathit{wgt}(t,\alpha))(\alpha) \tag{1}$$

where $i_{s,w} = 1$ iff $s = s_{\mathit{init}}$ and $w = 0$, and $i_{s,w} = 0$ otherwise. To see this, consider the map $T\colon \mathbb{R}^{S\times\mathbb{Z}}_{\geq 0} \to \mathbb{R}^{S\times\mathbb{Z}}_{\geq 0}$ given by

$$(y_{s,w})_{(s,w)\in S\times\mathbb{Z}} \mapsto i_{s,w} + \sum_{t \in S, \alpha \in \mathit{Act}(t)} P(t,\alpha,s) \cdot y_{t,w-\mathit{wgt}(t,\alpha)} \cdot \mathfrak{T}(t, w - \mathit{wgt}(t,\alpha))(\alpha).$$

A solution to equation (1) is fixed point of T. Further, T is monotone in all arguments. So, the least fixed point can be found by repeatedly applying T to the 0-vector in $\mathbb{R}^{S\times\mathbb{Z}}_{\geq 0}$. The equation (1) and hence the map T are chosen such that $T^n(0)$ contains the expected number of visits to the state-weight pairs under $\mathfrak{T}$ within the first $n-1$ steps. The summand $i_{s,w}$ reflects that after 0 steps the expected number of visits to $(s_{\mathit{init}}, 0)$ is 1. We conclude that $\lim_{n\to\infty} T^n(0) = (\vartheta^{\mathfrak{T}}_{s,w})_{(s,w)\in S\times\mathbb{Z}}$.

By spelling out the last steps of the paths in the definition of $\vartheta^{\mathfrak{S}}_{s,w}$, one can see that $(\vartheta^{\mathfrak{S}}_{s,w})_{(s,w)\in S\times\mathbb{Z}}$ provides a solution to the set of equations and is hence a fixed point of T. Let $\vartheta^{\mathfrak{S},\leq n}_{s,w}$ be the expected number of visits to (s,w) under $\mathfrak{S}$ within at most n steps. On the one hand, clearly $\vartheta^{\mathfrak{S},\leq n}_{s,w} \leq \vartheta^{\mathfrak{T}}_{s,w}$ for all n. On the other hand, $\vartheta^{\mathfrak{S}}_{s,w} = \lim_{n\to\infty} \vartheta^{\mathfrak{S},\leq n}_{s,w}$. Hence, $\vartheta^{\mathfrak{S}}_{s,w} \leq \vartheta^{\mathfrak{T}}_{s,w}$ for all (s,w). We conclude that $\vartheta^{\mathfrak{S}}_{s,w} = \vartheta^{\mathfrak{T}}_{s,w}$ for all (s,w) because $(\vartheta^{\mathfrak{T}}_{s,w})_{(s,w)\in S\times\mathbb{Z}}$ was the least fixed point of T. By the definition of $\mathfrak{T}$, the expected number of times action α is chosen in (s,w) under $\mathfrak{T}$ is hence $\vartheta^{\mathfrak{S}}_{s,w,\alpha}$ as well and the claim follows because the partial expectation only depends on the expected frequencies of the state weight pairs (goal, w) for $w \in \mathbb{Z}$ under the two schedulers. □

Now, we show that randomization is not necessary to approximate the optimal partial expectation.

Lemma 3.20. *Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{\mathit{init}}} < \infty$. For each weight-based scheduler $\mathfrak{S}$ for $\mathcal{M}$, there is a weight-based deterministic scheduler $\mathfrak{T}$ with $\mathbb{PE}^{\mathfrak{T}}_{\mathcal{M},s_{\mathit{init}}} \geq \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}$.*

Proof. Assume that $\mathit{Goal} = \{\mathit{goal}\}$ and that all states that cannot reach Goal are collapsed to a trap state *fail*. Let $\mathfrak{S}$ be a weight-based randomized scheduler. So, $\mathfrak{S}$ can be seen as a function from $S \times \mathbb{Z}$ to probability distributions over actions. For this scheduler, we can write $e^{\mathfrak{S}}_{s,w}$ for the partial expectation that is obtained when starting in state s with weight w. Observe that

$$e^{\mathfrak{S}}_{s,w} = \begin{cases} w & \text{if } s = \mathit{goal}, \\ 0 & \text{if } s = \mathit{fail}, \\ \sum_{\alpha \in \mathit{Act}(s)} \mathfrak{S}(s,w)(\alpha) \sum_{t \in S} P(s,\alpha,t) e^{\mathfrak{S}}_{t,w+\mathit{wgt}(s,\alpha)} & \text{otherwise.} \end{cases}$$

The values $e^{\mathfrak{S}}_{s,w}$ depend monotonically on each other. If one of the values $e^{\mathfrak{S}}_{s,w}$ was increased, the value at no other state-action pair would decrease. We will successively determinize the choices of the scheduler while making sure that the partial expectations from any state-weight pair on do not decrease.

Let $(s_i, w_i)_{i \in \mathbb{N}}$ be an enumeration of all state-weight pairs where s is not one of the two trap states. We recursively define a sequence of scheduler $(\mathfrak{S}_i)_{i \in \mathbb{N}}$. We set $\mathfrak{S}_0 = \mathfrak{S}$. To define $\mathfrak{S}_{i+1}$ from $\mathfrak{S}_i$, we observe that there has to be an action α_i such that

$$\sum_{t \in S} P(s,\alpha_i,t) e^{\mathfrak{S}_i}_{t,w+\mathit{wgt}(s,\alpha)} \geq \sum_{\alpha \in \mathit{Act}(s)} \mathfrak{S}_i(s,w)(\alpha) \sum_{t \in S} P(s,\alpha,t) e^{\mathfrak{S}_i}_{t,w+\mathit{wgt}(s,\alpha_i)}$$

due to the convex combination on the right-hand side. We let $\mathfrak{S}_{i+1}$ choose α_i when in state s_i with weight w_i and let $\mathfrak{S}_{i+1}$ behave as $\mathfrak{S}_i$ on all other state-weight pairs. By the observation above, we conclude that

$$e^{\mathfrak{S}_{i+1}}_{s,w} \geq e^{\mathfrak{S}_i}_{s,w}$$

for all state-weight pairs (s,w). So, the sequence $e^{\mathfrak{S}_i}(s,w)$ increases monotonically for $i \to \infty$ for each (s,w). By the assumption that the maximal partial expectation is finite, the sequence is also bounded and hence converges to a value $e^{\infty}_{s,w}$. This value is obtained by the scheduler obtained in the limit $\mathfrak{S}_\infty$. This scheduler can be defined on each state-action pair (s_i, w_i) by

$$\mathfrak{S}_\infty(s_i, w_i) \overset{\text{def}}{=} \mathfrak{S}_{i+1}(s_i, w_i) = \mathfrak{S}_k(s_i, w_i) \text{ for all } k \geq i+1.$$

So, $\mathfrak{S}_\infty$ is the desired weight-based deterministic scheduler. □

The two lemmata, Lemma 3.19 and Lemma 3.20, imply Proposition 3.17. It remains to show that the maximal partial expectation is not only the supremum over all weight-based deterministic schedulers, but that the optimal value is also obtained by a weight-based deterministic scheduler. Given an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ with arbitrary integer weights, we define the following metric $d^{\mathcal{M}}$ on the set of weight-based deterministic schedulers, i.e. on the set of functions from $S \times \mathbb{Z} \to \mathit{Act}$: For two such schedulers $\mathfrak{S}$ and $\mathfrak{T}$, we let

$$d^{\mathcal{M}}(\mathfrak{S}, \mathfrak{T}) := 2^{-R}$$

where R is the greatest natural number such that

$$\mathfrak{S} \restriction S \times \{-(R-1), \ldots, R-1\} = \mathfrak{T} \ \restriction \ S \times \{-(R-1), \ldots, R-1\}$$

or 0 if there is no greatest such natural number and the schedulers are hence the same. In other words, R is the smallest absolute value of a weight w in a state weight pair (s, w) on which $\mathfrak{S}$ and $\mathfrak{T}$ disagree.

Lemma 3.21. *Let $\mathcal{M}$ and $d^{\mathcal{M}}$ be as above. The metric space $(\mathit{Act}^{S\times\mathbb{Z}}, d^{\mathcal{M}})$ is compact.*

Proof. We can identify $\mathit{Act}^{S\times\mathbb{Z}}$ with $(\mathit{Act}^{S\times\{+,-\}})^{\mathbb{N}}$. Due to the symmetric treatment of positive and negative weights in the definition, the metric $d^{\mathcal{M}}$ induces the usual tree topology on this finitely branching tree of height ω. Therefore, the space is homeomorphic to the Cantor space 2^ω and hence compact. □

Having defined this compact space of schedulers, we can rely on the analytic notion of upper semi-continuity. Recall that a function $f\colon (X, d) \to (\mathbb{R}_\infty, d^{\mathit{euclid}})$ where (X, d) is a metric space, $\mathbb{R}_\infty = \mathbb{R} \cup \{-\infty, \infty\}$, and d^{euclid} is the Euclidean metric, is called *upper semi-continuous* if for each $\epsilon > 0$ and each x_0 with $f(x_0) > -\infty$ there is a $\delta > 0$ such that $f(x) \leq f(x_0) + \epsilon$ for all $x \in X$ with $d(x, x_0) < \delta$ and if further $f(x) \to -\infty$ for $x \to x_0$ for all x_0 with $f(x_0) = -\infty$.

Lemma 3.22 (Upper Semi-Continuity of Partial Expectations)**.** *Consider an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{\mathit{init}}} < \infty$. The function*

$$\mathbb{PE}^{\bullet}_{\mathcal{M},s_{\mathit{init}}}\colon (\mathit{Act}^{S\times\mathbb{Z}}, d^{\mathcal{M}}) \to (\mathbb{R}_\infty, d^{\mathit{euclid}})$$

assigning the value $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$ *to a weight-based deterministic scheduler* $\mathfrak{S} \in Act^{S\times\mathbb{Z}}$ *is upper semi-continuous.*

Proof. We again use the notation $e^{\mathfrak{S}}_{s,w}$ to denote the partial expectation under a weight-based deterministic scheduler $\mathfrak{S}$ starting from state s with weight w as in the proof of Lemma 3.20. Let $\mathfrak{S}$ be a weight-based deterministic scheduler with $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} > -\infty$. Given $\epsilon > 0$, we will define a natural number R such that any weight-based deterministic scheduler $\mathfrak{T}$ with $\mathfrak{T} \upharpoonright S \times [-R,R] = \mathfrak{S} \upharpoonright S \times [-R,R]$ satisfies $\mathbb{PE}^{\mathfrak{T}}_{\mathcal{M},s_{init}} < \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} + 4\epsilon$.

First, we observe that for each state s there is a natural number W_s and a probability $p_s < 1$ such that $\Pr^{\max}_{\mathcal{M},s}(\Diamond wgt > W_s) \leq p_s$ because there are no positively weight-divergent end components by the assumption that $\mathbb{PE}^{\max}_{\mathcal{M},s} < \infty$ (this argument was also used in Proposition 3.11). We define $W \overset{\text{def}}{=} \max_s W_s + \max_{s,\alpha} |wgt(s,\alpha)|$ and $p \overset{\text{def}}{=} \max_s p_s$. Then, for each state s and each natural number n, we have that

$$\Pr^{\max}_{\mathcal{M},s}(\Diamond wgt > n \cdot W) \leq p^n.$$

So, we obtain the following upper bound on the expected accumulated weight from any state on:

$$B \overset{\text{def}}{=} \sum_{n=0}^{\infty}(n+1)\cdot W \cdot p^n = \frac{W}{(1-p)^2}.$$

Further, let $\Diamond^{=n}Goal$ denote the event that $Goal$ is reached with accumulated weight n and $\Diamond wgt \leq -\ell$ the event that a prefix of the run has weight $\leq -\ell$. If ℓ is large, the probability that a path reaches positive weight again after reaching a weight below $-\ell$ decreases exponentially for $\ell \to \infty$ as we have seen. On the other hand, $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} > -\infty$ and so we observe that

$$\lim_{\ell\to\infty} \sum_{n\in\mathbb{Z}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond^{=n}Goal \wedge \Diamond wgt \leq -\ell)\cdot |n| = 0.$$

In other words, this means that the contribution of paths that reach *goal* with weight below $-\ell$ to the partial expectation vanishes for $\ell \to \infty$. We define ℓ_ϵ to be the smallest natural number such that

$$\sum_{n\in\mathbb{Z}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond^{=n}Goal \wedge \Diamond wgt \leq -\ell_\epsilon)\cdot |n| < \epsilon.$$

Note that also $\sum_{n\in\mathbb{Z}} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond^{=n}Goal \wedge \Diamond wgt \leq -\ell)\cdot |n| < \epsilon$ for all $\ell > \ell_\epsilon$ due to the use of the absolute value $|n|$. Let k_ϵ be the smallest natural number such that $p^{k_\epsilon} \cdot B < \epsilon$. Let $M \overset{\text{def}}{=} \max_{s,\alpha} |wgt(s,\alpha)|$ and define $R^- \overset{\text{def}}{=} \max\{\ell_\epsilon, k_\epsilon \cdot W + M\}$. Further, define

$$H \overset{\text{def}}{=} \min\{0, e^{\mathfrak{S}}_{s,r} | s \in S, 0 \leq r \leq M\} \leq 0.$$

Finally, let m_ϵ be the least natural number such that

$$p^{m_\epsilon} \cdot (R^- + B + M - H) < \epsilon \text{ and } p^{m_\epsilon} \cdot (m_\epsilon \cdot W + M + B) < \epsilon.$$

Define $R^+ \overset{\text{def}}{=} m_\epsilon \cdot W$ and $R \overset{\text{def}}{=} \max\{R^-, R^+\}$. Note that also $p^{m_\epsilon} \cdot (R + B + M) < \epsilon$. We claim that R does the job. So let $\mathfrak{T}$ be a scheduler with $\mathfrak{T} \restriction S \times [-R, R] = \mathfrak{S} \restriction S \times [-R, R]$. Let

$$\begin{aligned}
\mathfrak{P}^+ &\overset{\text{def}}{=} \{\pi \text{ finite path} | \mathit{wgt}(\pi) > R \text{ and any proper prefix } \pi' \text{ satisfies } \mathit{wgt}(\pi') \in [-R, R]\}, \\
\mathfrak{P}^- &\overset{\text{def}}{=} \{\pi \text{ finite path} | \mathit{wgt}(\pi) < R \text{ and any proper prefix } \pi' \text{ satisfies } \mathit{wgt}(\pi') \in [-R, R]\}.
\end{aligned}$$

The schedulers $\mathfrak{S}$ and $\mathfrak{T}$ agree on all paths without a prefix in these two sets. So,

$$\mathbb{PE}^{\mathfrak{T}}_{\mathcal{M}, s_{init}} - \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}, s_{init}} = \sum_{\pi \in \mathfrak{P}^+ \cup \mathfrak{P}^-} (e^{\mathfrak{T}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)} - e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}) \cdot \Pr^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(\pi)$$

We will split up the sum to $\mathfrak{P}^+$ and $\mathfrak{P}^-$. For a path $\pi \in \mathfrak{P}^+$, we see that $e^{\mathfrak{T}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}$ is bounded by $R+M+B$ as $\mathit{wgt}(\pi) \leq R+M$ and the partial expectation obtained on top of this weight is bounded by B. Further, $e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}$ is at least H. To see this note that the partial expectation from $(\mathit{last}(\pi), \mathit{wgt}(\pi))$ on could only be negative if the weight drops below 0 again. Along each path back to negative accumulated weight, a state-weight pair (s, r) with $0 \leq r \leq M$ has to be visited. The value H is a lower bound on the partial expectation obtained from any such state-weight pair. Finally, $\Pr^{\max}_{\mathcal{M}, s_{init}}(\Diamond \mathit{wgt} \geq R) \leq p^{m_\epsilon}$ as $R \geq m_\epsilon \cdot W$. We conclude

$$\begin{aligned}
& \sum_{\pi \in \mathfrak{P}^+} (e^{\mathfrak{T}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)} - e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}) \cdot \Pr^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(\pi) \\
\leq\ & \Pr^{\mathfrak{S}}_{s_{init}}(\mathfrak{P}^+) \cdot (R + M + B - \min_{\pi \in \mathfrak{P}^+} e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}) \\
\leq\ & p^{m_\epsilon} \cdot (R + M + B - H) < 2\epsilon.
\end{aligned}$$

For the remaining sum first consider $\sum_{\pi \in \mathfrak{P}^-} \Pr^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(\pi) \cdot e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}$. The absolute value of this sum is at most $\sum_{n \in \mathbb{Z}} \Pr^{\mathfrak{S}}_{\mathcal{M}, s_{init}}(\Diamond^{=n} \mathit{Goal} \wedge \Diamond \mathit{wgt} \leq -R) \cdot |n|$ which is less than ϵ as $\ell_\epsilon \leq R$. The sum, $\sum_{\pi \in \mathfrak{P}^-} \Pr^{\mathfrak{T}}_{\mathcal{M}, s_{init}}(\pi) \cdot e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}$ on the other hand can be bounded from above: By the definition of R^-, the probability to reach positive weight starting from a weight below R^- is at most p^{k_ϵ}. From then on at most a partial expectation of B can be obtained. As $p^{k_\epsilon} \cdot B < \epsilon$, we conclude that the considered sum is less than ϵ. All in all, we conclude

$$\sum_{\pi \in \mathfrak{P}^-} (e^{\mathfrak{T}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)} - e^{\mathfrak{S}}_{\mathit{last}(\pi), \mathit{wgt}(\pi)}) \cdot \Pr^{\mathfrak{S}}_{s_{init}}(\pi)$$

$$= \sum_{\pi \in \mathfrak{P}^-} e^{\mathfrak{T}}_{last(\pi),wgt(\pi)} \cdot \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\pi) - \sum_{\pi \in \mathfrak{P}^-} e^{\mathfrak{S}}_{last(\pi),wgt(\pi)} \cdot \Pr^{\mathfrak{S}}_{s_{init}}(\pi) < 2\epsilon.$$

Put together, we obtain $\mathbb{PE}^{\mathfrak{T}}_{\mathcal{M},s_{init}} - \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} < 4\epsilon$. This finishes the case $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} > -\infty$. If $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = -\infty$, we have to show for each $b \in \mathbb{R}$ that there is an R such that all schedulers agreeing with $\mathfrak{S}$ on the weight-window $[-R, R]$ have a partial expectation below b. But as we can make

$$\sum_{\zeta \vDash \Diamond Goal \wedge \Box wgt \in [-R,R]} wgt(\zeta) \cdot \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\zeta)$$

arbitrarily small, this follows directly. □

We arrive at the main result of this section.

Theorem 3.23 (Existence of Optimal Schedulers for Partial Expectations). *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$. There is a weight-based deterministic scheduler $\mathfrak{S}$ with $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$.*

Proof. As $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$, the map $\mathbb{PE}^{\bullet}_{\mathcal{M},s_{init}} \colon (Act^{S\times\mathbb{Z}}, d^{\mathcal{M}}) \to (\mathbb{R}_\infty, d^{euclid})$ is upper semi-continuous by Lemma 3.22. So, this map has a global maximum because $(Act^{S\times\mathbb{Z}}, d^{\mathcal{M}})$ is a compact metric space by Lemma 3.21. This maximum agrees with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ by Proposition 3.17 stating that the maximal partial expectation can be expressed as the supremum over weight-based deterministic schedulers. □

The existence of optimal weight-based deterministic schedulers follows now easily.

Corollary 3.24 (Existence of Optimal Schedulers for Conditional Expectations). *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$. There is a weight-based deterministic scheduler $\mathfrak{S}$ with $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$.*

Proof. After pre-processing, we can assume that $\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond Goal) > 0$. If we define $\vartheta \overset{\text{def}}{=} \mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ and apply the reduction of Proposition 3.16 to the threshold problem $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} \geq \vartheta$, we obtain the MDP $\mathcal{N}$ by adding a new goal state *goal′* and a terminal weight $-\vartheta$ on the last step from the original goal to *goal′*. In $\mathcal{N}$, we have $\mathbb{PE}^{\max}_{\mathcal{N},s_{init}} \geq 0$. So, there is a weight-based deterministic scheduler $\mathfrak{S}$ with $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N},s_{init}} = 0$ by Theorem 3.23. This scheduler used in $\mathcal{M}$ obtains $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \vartheta$. Note that we did not need weights to be rational for the proofs of Proposition 3.16 and Theorem 3.23. So, the argument also works if ϑ is irrational. □

3.5 Non-negative weights

As already indicated in Section 3.2.3, major obstacles make the computation of optimal partial and conditional expectations in MDPs with arbitrary integer weights difficult.

While optimal schedulers can be chosen to be weight-based and deterministic, a further characterization of the structure of optimal schedulers turns out to be challenging. In fact, we will see in Chapter 5 that we cannot expect to make progress on this problem with known techniques. An algorithm to compute optimal partial and conditional expectations would solve long-standing open number-theoretic problems. Hence, most likely, such an algorithm would require the development of new tools, if it exists at all.

Restricting our attention to the important special case of MDPs with non-negative weights, the situation becomes manageable. The problem to compute optimal conditional expectations in MDPs with non-negative weights has been solved in [BKKW17]. In [CFK$^+$13a], computation algorithms of maximal partial expectations in stochastic multiplayer games with non-negative weights have been presented. MDPs are a special case of these multiplayer games. In this section, we adapt the solution from [CFK$^+$13a] to the simpler case of MDPs with non-negative weights. A key result is the existence of a *saturation point*, a bound on the accumulated weight above which optimal schedulers do not need memory. In MDPs, we can provide a smaller saturation point than the one provided in [CFK$^+$13a]. The running time of the algorithm directly depends on the size of the saturation point. While the saturation point we provide here still leads to an exponential time algorithm, like the algorithms for stochastic multiplayer games, our saturation point for MDPs is the smallest possible saturation point. It can be considerably smaller than the saturation point provided in [CFK$^+$13a] which can be a great benefit in practice.

3.5.1 Saturation points

In order to be able to compute optimal partial expectations in MDPs with non-negative weights, we will show that we can further restrict the complexity of schedulers necessary for the optimization. The key result states that optimal schedulers can be chosen to behave memoryless as soon as a certain amount of weight is accumulated along a path. Let us first provide a formal definition of such saturation points.

Definition 3.25 (Saturation point)**.** Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ be an MDP with non-negative weights and assume that $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$ (or $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$, respectively). A *saturation point* for partial (conditional) expectations in $\mathcal{M}$ is a natural number K such that there is a weight-based deterministic scheduler $\mathfrak{S}$ that satisfies

1. $\mathfrak{S}(s, w) = \mathfrak{S}(s, w')$ for all state-weight pairs (s, w) and (s, w') with $w, w' \geq K$,
2. $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ $(\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{CE}^{\max}_{\mathcal{M},s_{init}})$. ◁

The existence of a saturation point for conditional expectations computable in polynomial time has been shown in [BKKW17]. In the sequel, we adapt the saturation point result for partial expectations in stochastic multiplayer games from [CFK$^+$13a]. MDPs are a special case of these games and we can not only adapt, but also improve the result for the case of MDPs. In order to prove the existence of a saturation point for partial expectations, we first need a scheduler maximizing the partial expectation among all schedulers that reach *Goal* with the maximal possible probability. The behavior of this scheduler is precisely the optimal behavior as soon as a saturation point is reached. We first introduce some additional notation. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with non-negative weights and assume that $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$. Let us assume that $Goal = \{goal\}$ and that all states from which *Goal* is not reachable are collapsed to one trap state *fail*. After our pre-processing procedure, we can assume that there are no non-trivial end components in $\mathcal{M}$ as there cannot be any end components with negative maximal expected mean payoff. For each state $s \in S$, we denote the maximal probability to reach *Goal* by

$$p^{\max}_s \stackrel{\text{def}}{=} \Pr^{\max}_{\mathcal{M},s}(\Diamond Goal).$$

Further, we denote the maximal probability to reach *Goal* when taking action α in s by

$$p^{\max}_{s,\alpha} \stackrel{\text{def}}{=} \sum_{t\in S} P(s,\alpha,t) \cdot p^{\max}_t.$$

We then define the set of actions $Act^{\max}(s)$ that enables us to reach *Goal* with this maximal probability for each state s by

$$Act^{\max}(s) \stackrel{\text{def}}{=} \{\alpha \in Act(s) \mid p^{\max}_{s,\alpha} = p^{\max}_s\}.$$

Consider the sub-MDP $(S, Act^{\max})$ which we call $\mathcal{M}^{\max}$. Note that all schedulers $\mathfrak{S}$ for $\mathcal{M}^{\max}$ satisfy $\Pr^{\mathfrak{S}}_{\mathcal{M}^{\max},s}(\Diamond Goal) = p^{\max}_s$ for all states s because there are no non-trivial end components in $\mathcal{M}^{\max}$. This allows us to compute maximal partial expectations in $\mathcal{M}^{\max}$ via a classical stochastic shortest path problem. We define a new weight function to obtain a variant $\mathcal{N}^{\max}$ of $\mathcal{M}^{\max}$ by $wgt_{\mathcal{N}^{\max}}(s,\alpha) \stackrel{\text{def}}{=} wgt(s,\alpha) \cdot p^{\max}_s$ for all state-action pairs (s,α) in $\mathcal{M}^{\max}$. We observe that

$$\mathbb{PE}^{\max}_{\mathcal{M}^{\max},s} = \mathbb{E}^{\max}_{\mathcal{N}^{\max},s}(\oplus\{goal, fail\}).$$

This value can hence be computed in polynomial time. More details can also be found in [CFK⁺13a]. Furthermore, a memoryless deterministic scheduler $\mathfrak{Max}$ solving this classical shortest path problem in $\mathcal{N}^{\max}$ also maximizes the partial expectation in $\mathcal{M}^{\max}$. Fix one such memoryless deterministic scheduler $\mathfrak{Max}$. For the saturation point, the following value is of interest:

$$M \stackrel{\text{def}}{=} \min_{s\in S} \mathbb{PE}^{\mathfrak{Max}}_{\mathcal{M},s}.$$

As we also assume that all states are reachable from s_{init}, we can conclude that $\mathbb{PE}^{\max}_{\mathcal{M},s} < \infty$ for all states $s \in S$. So, we can define

$$B \overset{\text{def}}{=} \max_{s\in S} \mathbb{PE}^{\max}_{\mathcal{M},s}.$$

Of course, we do not yet know how to compute this number. The first existence result of a saturation point we are going to present, however, is only an auxiliary step before we provide the smallest possible saturation point later on. Nevertheless, we could replace B by an upper bound computable in polynomial time such as $\max_{s\in S} \mathbb{E}^{\max}_{\mathcal{M},s}(\oplus\{goal, fail\})$. The final ingredient to prove the existence of a saturation point as in [CFK+13a] is the following value δ quantifying how much smaller the probability to reach *Goal* under a non-optimal scheduler is compared to the maximal reachability probability:

$$\delta \overset{\text{def}}{=} \min_{s\in S, \alpha\in Act(s)\setminus Act^{\max}(s)} p^{\max}_s - p^{\max}_{s,\alpha}.$$

If the minimum should be taken over the empty set, the scheduler $\mathfrak{Max}$ is already maximizes the partial expectation as $\mathcal{M}$ and $\mathcal{M}^{\max}$ coincide in this case.

Proposition 3.26 (see also [CFK+13a]). *Let $\mathcal{M}$ and all notation be as above. The smallest natural number K with*

$$K \geq \frac{B - M}{\delta}$$

is a saturation point for partial expectations in $\mathcal{M}$.

Proof. Let $\mathfrak{S}$ be a weight-based deterministic schedulers. Let the scheduler $\mathfrak{S} \lhd_K \mathfrak{Max}$ be defined, for all state-weight pairs (s, w), by

$$\mathfrak{S} \lhd_K \mathfrak{Max}(s, w) \overset{\text{def}}{=} \begin{cases} \mathfrak{S}(s, w) & \text{if } w < K, \\ \mathfrak{Max}(s, w) & \text{otherwise.} \end{cases}$$

We claim that $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} \leq \mathbb{PE}^{\mathfrak{S}\lhd_K\mathfrak{Max}}_{\mathcal{M},s_{init}}$. Let us denote the partial expectation under $\mathfrak{S}$ from a state-weight pair (s, w) on by $e^{\mathfrak{S}}_{s,w}$, and similarly for $\mathfrak{Max}$. As the weight along a path cannot decrease and hence $\mathfrak{S} \lhd_K \mathfrak{Max}$ never switches back to the behavior of $\mathfrak{S}$ once an accumulated weight of K is reached, it is sufficient to show that $e^{\mathfrak{S}}_{s,w} \leq e^{\mathfrak{Max}}_{s,w}$ for each state-weight pair (s, w) with $w \geq K$.

If $\mathfrak{S}$ only chooses actions in $Act^{\max}(s)$ for such state-weight pairs, both schedulers reach *Goal* with probability $p^{\max}_s$. So, the values $e^{\mathfrak{S}}_{s,w}$ and $e^{\mathfrak{Max}}_{s,w}$ only differ by the weights that are accumulated from then on, while the already accumulated weight contributes $p^{\max}_s \cdot w$ to these values. Because $\mathfrak{Max}$ maximizes the partial expectation among all schedulers choosing only actions in $Act^{\max}$, we conclude that $e^{\mathfrak{S}}_{s,w} \leq e^{\mathfrak{Max}}_{s,w}$ in this case.

So, assume that $\mathfrak{S}(s,w) \notin Act^{\max}(s)$ for some pair (s,w) with $w \geq K$. We can bound the value $e^{\mathfrak{S}}_{s,w}$ from above as follows:

$$e^{\mathfrak{S}}_{s,w} \leq (p^{\max}_s - \delta) \cdot w + B.$$

On the other hand, the value $e^{\mathfrak{Max}}_{s,w}$ satisfies

$$e^{\mathfrak{Max}}_{s,w} \geq p^{\max}_s \cdot w + M.$$

So,

$$e^{\mathfrak{S}}_{s,w} - e^{\mathfrak{Max}}_{s,w} \leq B - M - \delta \cdot w \leq B - M - \delta \cdot K \leq 0.$$ □

The existence of a saturation point implies that an optimal scheduler for the partial expectation cannot only be chosen to be weight-based and deterministic, but also eventually memoryless. This in particular means that there are optimal finite-memory schedulers as an optimal scheduler only has to keep track of the accumulated weight until it exceeds the saturation point K.

We use this first saturation point result in the proof of the following theorem that provides the smallest possible saturation point. The idea is that we can compute the largest weight w such that the scheduler $\mathfrak{Max}$ can be improved by changing the action at precisely one state-action pair with weight w and nowhere else. We then see that $\mathfrak{Max}$ cannot be improved above any weight level if it cannot be improved by such a change at a single state-weight pair. We can conclude that $w+1$ is a saturation point. We denote the partial expectation under the scheduler choosing α in s before acting like $\mathfrak{Max}$ by

$$\mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} \overset{\text{def}}{=} p^{\max}_{s,\alpha} \cdot wgt(s,\alpha) + \sum_{t \in S} P(s,\alpha,t) \cdot \mathbb{PE}^{\mathfrak{Max}}_t.$$

Theorem 3.27. *Let $\mathcal{M}$ and all notation be as above. Then, the smallest natural number K with*

$$K \geq \max \left\{ \frac{\mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} - \mathbb{PE}^{\mathfrak{Max}}_s}{p^{\max}_s - p^{\max}_{s,\alpha}} \,\middle|\, s \in S, \alpha \in Act(s) \setminus Act^{\max}(s) \right\}$$

is an upper saturation point for partial expectations in $\mathcal{M}$.

Proof. Let us denote the partial expectation under a scheduler $\mathfrak{S}$ from a state-weight pair (s,w) on by $e^{\mathfrak{S}}_{s,w}$ again. Let $\mathfrak{S}$ be a scheduler maximizing $e^{\mathfrak{S}}_{s,w}$ for all state-weight pairs (s,w). We can assume that $\mathfrak{S}$ is weight-based deterministic and, by the the existence of a saturation point (Proposition 3.26), that $\mathfrak{S}$ agrees with $\mathfrak{Max}$ from some weight level on. In fact, if we let w be the largest weight level such that there is a state s such that $\mathfrak{S}$ chooses an action not in $Act^{\max}(s)$ at the pair (s,w), we can assume that $\mathfrak{S}$ agrees with $\mathfrak{Max}$ on all state-weight pairs (t,v) with $v > w$. Let us further assume that $\mathfrak{S}$ is chosen such that w is minimal among all optimal schedulers.

Suppose now that $w \geq K$. First, suppose that there is a state s such that $\mathfrak{S}(s,w) \notin Act^{\max}(s)$ and $wgt(s, \mathfrak{S}(s,w)) > 0$. In this case,

$$e^{\mathfrak{S}}_{s,w} = \mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} + p^{\max}_{s,\alpha} \cdot w$$

as $\mathfrak{S}$ behaves like $\mathfrak{Max}$ after this step. By our assumption, we know that $e^{\mathfrak{Max}}_{s,w} - e^{\mathfrak{S}}_{s,w} \leq 0$. If we would have equality, we could change the behavior of $\mathfrak{S}$ to the behavior of $\mathfrak{Max}$ at this state-weight pair. So assume that $e^{\mathfrak{Max}}_{s,w} - e^{\mathfrak{S}}_{s,w} < 0$. As $e^{\mathfrak{Max}}_{s,w} = \mathbb{PE}^{\mathfrak{Max}}_{s} + p^{\max}_{s} \cdot w$, we observe

$$0 > \mathbb{PE}^{\mathfrak{Max}}_{s} - \mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} + w(p^{\max}_{s} - p^{\max}_{s,\alpha})$$

and hence

$$(\mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} - \mathbb{PE}^{\mathfrak{Max}}_{s})/(p^{\max}_{s} - p^{\max}_{s,\alpha}) > w.$$

This contradicts the assumption that $w \geq K$ by the definition of K.

Hence, we can assume that for all states s with $\mathfrak{S}(s,w) \notin Act^{\max}(s)$, we have $wgt(s, \mathfrak{S}(s,w)) = 0$. Denote the set of these states by T. We define for each state $t \in T$,

$$D_t := e^{\mathfrak{S}}_{t,w} - e^{\mathfrak{Max}}_{t,w}.$$

If the maximal value D_t is 0, we could change the behavior of $\mathfrak{S}$ to the behavior of $\mathfrak{Max}$ at weight level w without affecting the partial expectation. This would contradict the minimality of w. So, let $D \overset{\text{def}}{=} \max_t D_t > 0$ and defined $T' \overset{\text{def}}{=} \{t \in T | D_t = D\}$. Note that *goal* and *fail* are not in T'. Let $t \in T$ and let $\alpha = \mathfrak{S}(t,w)$. We claim that all states reachable from t via α are in T'. As $wgt(s,\alpha) = 0$, we get

$$\begin{aligned} D_t &= e^{\mathfrak{S}}_{t,w} - e^{\mathfrak{Max}}_{t,w} \\ &= \sum_{s \in S} P(t,\alpha,s) \cdot e^{\mathfrak{S}}_{s,w} - e^{\mathfrak{Max}}_{t,w} \\ &= (\mathbb{PE}^{\mathfrak{Max}}_{t,\alpha} + w \cdot p^{\max}_{t,\alpha} + \sum_{s \in S} P(t,\alpha,s) D_s) - (\mathbb{PE}^{\mathfrak{Max}}_{t} + p^{\max}_{t} \cdot w) \end{aligned}$$

We conclude

$$\begin{aligned} D_t - \sum_{s \in S} P(t,\alpha,s) D_s &\leq \mathbb{PE}^{\mathfrak{Max}}_{t,\alpha} - \mathbb{PE}^{\mathfrak{Max}}_{t} + w(p^{\max}_{t,\alpha} - p^{\max}_{t}) \\ &\leq 0. \end{aligned}$$

The last inequality follows from $w \geq K$ and the definition of K. As $t \in T'$ and hence D_t is maximal, this is only possible if $D_s = D_t = D$ for all s with $P(t,\alpha,s) > 0$. So, indeed all α-successors of t are in T' again. As this holds for all states in T' with the respective action chosen by $\mathfrak{S}$ at weight level w, there must be a non-trivial end component inside

T'. This contradicts the fact that in the pre-processed MDP $\mathcal{M}$, there are no non-trivial end components. □

The saturation point provided in this proposition is computable in polynomial time. Furthermore, let us show in more detail that it is the smallest possible saturation point: For any weight $w < K$, it is suboptimal to follow the decisions of $\mathfrak{Max}$. This is implicitly contained in the definition of K in Theorem 3.27: For a weight $w < K$, the scheduler $\mathfrak{Max}$ can be improved by changing the choice of the scheduler to the action $\alpha \notin \mathit{Act}^{\max}(s)$ at the state s that obtain the maximum

$$\max \left\{ \frac{\mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} - \mathbb{PE}^{\mathfrak{Max}}_{s}}{p^{\max}_s - p^{\max}_{s,\alpha}} \,\middle|\, s \in S, \alpha \in \mathit{Act}(s) \setminus \mathit{Act}^{\max}(s) \right\}.$$

To see that, observe that for the maximizing pair (s, α), we have

$$w \cdot p^{\max}_{s,\alpha} + \mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} > w \cdot p^{\max}_{s} + \mathbb{PE}^{\mathfrak{Max}}_{s}$$

if w is less than K and hence less than $\frac{\mathbb{PE}^{\mathfrak{Max}}_{s,\alpha} - \mathbb{PE}^{\mathfrak{Max}}_{s}}{p^{\max}_s - p^{\max}_{s,\alpha}}$.

3.5.2 Computation of optimal values

The existence of a computable saturation point allows us to reduce the problem of computing the maximal partial expectation to a weighted reachability problem. Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ again be an MDP with non-negative weights and assume that $\mathbb{PE}^{\max}_{\mathcal{M},s_{\mathit{init}}} < \infty$. After our pre-processing procedure, we can assume that $\mathcal{M}$ does not contain non-trivial end components, that all states that cannot reach Goal are collapsed to a trap state fail and that $\mathit{Goal} = \{\mathit{goal}\}$ is a singleton. In order to construct the weighted reachability problem, we extend the state space by explicitly encoding the accumulated weight into the states. States in the transformed MDP are of the form (s, w) with $s \in S$ and $w \in \mathbb{N}$. Let K be the saturation point given in Theorem 3.27. The key insight is that we can easily compute the maximal partial expectation when starting in state s with weight $w \geq K$: As the scheduler $\mathfrak{Max}$ defined in the previous section is optimal at the weight levels above K, this partial expectation is $\mathbb{PE}^{\mathfrak{Max}}_{s} + w \cdot p^{\max}_s$. In our weighted reachability problem, we can hence make state-weight pairs (s, w) with $w \geq K$ terminal with this partial expectation as terminal weight. Let W be the maximal weight occurring in $\mathcal{M}$. After making the state-weight pairs at weight levels K and above terminal, state-weight pairs with weight above $K + W - 1$ are not reachable anymore. In this way, we obtain a finite-state MDP for the weighted reachability problem.

Let us define the instance of the weighted reachability problem more formally. We construct an MDP $\mathcal{N}$. The state space is $S \times \{0, \ldots, K + W - 1\}$. The actions are the same as in $\mathcal{M}$. The transition probability function $P_{\mathcal{N}}$ is given by

$$P_{\mathcal{N}}((s,w),\alpha,(t,w+wgt(s,\alpha)))=P(s,\alpha,t)$$

for all $s,t\in S$ and $w<K$. All other transition probabilities are 0. In particular all state-weight pairs (s,w) with $w\geq K$ are traps. The initial state is $(s_{init},0)$. To all trap states, we assign a terminal weight: For $w<K$, we assign terminal weight 0 to state (fail,w) and terminal weight w to state (goal,w). For $K\leq w<K+W$ and $s\in S$, we assign terminal weight $\mathbb{PE}^{\mathfrak{Max}}_{\mathcal{M},s}+w\cdot p^{\max}_{\mathcal{M},s}$ to state (s,w). Recall that $\mathbb{PE}^{\mathfrak{Max}}_{\mathcal{M},s}$ and $p^{\max}_{\mathcal{M},s}$ are computable in polynomial time.

Let *terminal* be the random variable that assigns the terminal weight of the trap state reached to runs in $\mathcal{N}$. Note that $\mathcal{N}$ does not contain non-trivial end components. So, a trap state is reached almost surely under any scheduler. Weight-based deterministic schedulers for $\mathcal{M}$ that switch to the behavior of $\mathfrak{Max}$ as soon as the accumulated weight exceeds K now correspond precisely to memoryless deterministic schedulers for $\mathcal{N}$. As we know that one of these schedulers maximizes the partial expectation in $\mathcal{M}$ and a memoryless deterministic scheduler maximizes the expected terminal weight in $\mathcal{N}$, we can restrict our attention to such schedulers. Given such a scheduler $\mathfrak{S}$ viewed as a scheduler for $\mathcal{M}$ and as a scheduler for $\mathcal{N}$, the construction makes sure that

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}=\mathbb{E}^{\mathfrak{S}}_{\mathcal{N},(s_{init},0)}(\mathit{terminal}).$$

To see that, note that for a path π in $\mathcal{M}$ that reaches *goal* or *fail* with accumulated weight less than K, the corresponding path $\hat{\pi}$ in $\mathcal{N}$ satisfies $\mathit{terminal}(\hat{\pi})=\oplus\mathit{Goal}(\pi)$. For a path π in $\mathcal{M}$ that reaches an accumulated weight of at least K in the last step, the corresponding path $\hat{\pi}$ in $\mathcal{N}$ has precisely the expected partial expectation under $\mathfrak{Max}$ of the continuations of π as its terminal weight. We conclude

$$\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}=\mathbb{E}^{\max}_{\mathcal{N},(s_{init},0)}(\mathit{terminal}).$$

Therefore, we can use the linear program computing the maximal expected terminal weight in a weighted reachability problem to compute maximal partial expectations. We give the linear program in the following proposition. The non-existence of non-trivial end components makes sure that this linear program for weighted reachability has a unique solution.

Proposition 3.28. *Let $\mathcal{M}$ be an MDP with non-negative weights as above, let K be the saturation point provided in Theorem 3.27, and let W be the maximal weight occurring in $\mathcal{M}$. The maximal partial expectation $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ is the value of the variable $x_{s_{init},0}$ in the unique solution to the following linear program with variables $(x_{s,w})_{s\in S,w\in\{0,\dots,K+W-1\}}$:*

Minimize $\sum_{s\in S, w\in\{0,\dots,K+W-1\}} x_{s,w}$ *under the following constraints:*

$$
\begin{aligned}
&x_{s,w} = p^{\max}_{\mathcal{M},s}\cdot w + \mathbb{PE}^{\mathfrak{Max}}_{\mathcal{M},s} && \textit{for } w \geq K,\\
&x_{goal,w} = w \textit{ and } x_{fail,w} = 0 && \textit{for } w < K,\\
&x_{s,w} \geq \sum_{t\in S} P(s,\alpha,t)\cdot x_{t,s+wgt(s,\alpha)}. && \textit{for } w < K, s\in S\setminus\{goal, fail\} \textit{ and } \alpha\in Act(s).
\end{aligned}
$$

From a solution x to the linear program, we can easily extract an optimal weight-based deterministic scheduler. This scheduler only needs finite memory because the accumulated weight increases monotonically along paths and as soon as the saturation point is reached $\mathfrak{Max}$ provides the optimal decisions.

Further, we can construct and solve the linear program in time exponential in the size of $\mathcal{M}$. As the saturation point K is computable in polynomial time, the numeric value of $K+W$ is at most exponential in the size of $\mathcal{M}$. Further, all values occurring in the linear program can be computed in polynomial time. So, this linear program is exponential in the size of the MDP and can hence be solved in exponential time. We state this result in the concluding theorem:

Theorem 3.29 (see also [CFK+13a])**.** *Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ with non-negative weights and with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$, the value $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ is computable in exponential time.*

The analogous result for conditional expectations is shown in [BKKW17]. The algorithm presented there iteratively fixes optimal decisions for different weight levels starting at a saturation point moving downwards. For each weight level, optimal decisions are derived from the solutions to linear programs.

3.5.3 Lower bounds

For conditional expectations, a lower bound for the threshold problem is provided in [BKKW17].

Theorem 3.30 (see [BKKW17])**.** *The threshold problem for maximal or minimal conditional expectations in acyclic MDPs with non-negative weights is PSPACE-hard.*

To obtain the analogous result for partial expectations, we provide a further reduction between from the threshold problem for conditional expectation to the threshold problem for partial expectations. As mentioned in Section 3.3.2, the first such reduction we presented introduces a negative weight when starting with an MDP with non-negative weights. For acyclic MDPs we can avoid this negative weight:

Lemma 3.31. *The threshold problem for conditional expectations in acyclic MDPs with non-negative weights is polynomial-time reducible to the threshold problem for partial expectations in such MDPs.*

Proof. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an acyclic MDP with non-negative integer weights. After pre-processing, we assume that all states are reachable from s_{init}, that $Goal = \{goal\}$ is a singleton, and that all states from which $goal$ is not reachable are collapsed to one trap state $fail$. Further, we can assume that $\Pr^{\min}_{\mathcal{M},s_{init}}(\lozenge Goal) > 0$ because $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$ as $\mathcal{M}$ is acyclic. Let ϑ be a rational. We are going to construct an acyclic MDP $\mathcal{N}$ and threshold values ϑ_1 and ϑ_2 such that $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} > \vartheta_1$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} \geq \vartheta_2$ and an analogous statement for minimal conditional/partial expectations. The structure of this MDP $\mathcal{N}$ is sketched in the following Figure 3.6:

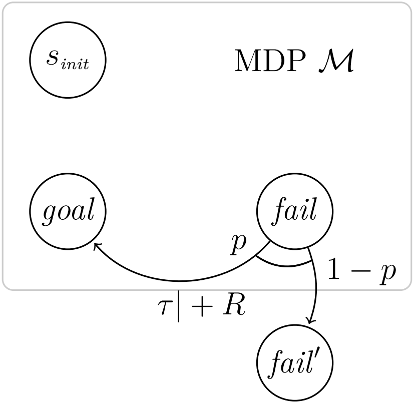


Figure 3.6: Construction of the MDP $\mathcal{N}$. The probability p and the weight R are chosen such that $pR = \vartheta$.

So, the essential task is to find appropriate values for p and R and corresponding threshold values for $\mathcal{N}$. For this we need to tackle the problem that different paths from s_{init} to $fail$ in $\mathcal{M}$ might have different accumulated weights.

For each non-trivial transition probability $P(s, \alpha, t) \in (0, 1)$ in $\mathcal{M}$, let $m_{s,\alpha,t}$ be the denominator of the probability in a co-prime representation as a fraction of non-negative integers. Let m be the product of the values $m_{s,\alpha,t}$ for all such transitions (s, α, t). The number of digits of a binary representation of m is polynomially bounded in the size of $\mathcal{M}$. The length of the denominators $m_{s,\alpha,t}$ is taken into account in the size of $\mathcal{M}$. As there are only polynomially many transitions in $\mathcal{M}$, the product of all these denominators still has a polynomially long binary representation.

As $\mathcal{M}$ is acyclic, the probability of each path π from s_{init} to $goal$ or $fail$ is a rational number of the form ℓ/m for for some natural number ℓ. The same applies to the probabilities for reaching $goal$ from s_{init} and the partial expectations under deterministic schedulers. That is, if $\mathfrak{S}$ is a deterministic scheduler then

$$\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal), \quad \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} \in \{\ell/m \,|\, \ell \in \mathbb{N}\}$$

Consider a representation of the threshold ϑ as the quotient a/b of two positive integers a, b. Let

$$\delta \stackrel{\text{def}}{=} \frac{1}{bm}$$

Then, for each deterministic scheduler $\mathfrak{S}$, the value $\vartheta \cdot \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond goal)$ and the partial expectation $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}}$ are integer-multiples of δ. This yields:

$$\text{If } y = \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond goal) \text{ and } \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} > \vartheta y \text{ then } \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \geq \vartheta y + \delta. \tag{*}$$

Let now

$$w \stackrel{\text{def}}{=} 1 + \max\Big\{ \, wgt(\pi) \, | \, \pi \text{ is a path from } s_{init} \text{ to } fail \text{ in } \mathcal{M} \, \Big\}$$

and define

$$p \stackrel{\text{def}}{=} \frac{\delta}{2w} \quad \text{and} \quad R \stackrel{\text{def}}{=} \frac{2w\vartheta}{\delta}.$$

Note that w is finite (recall that $\mathcal{M}$ is acyclic) and computable in polynomial time and that the logarithmic length of the numerator and denominator of the rational numbers p and R is polynomial in the sizes of the given MDP $\mathcal{M}$ and the threshold value ϑ. Obviously:

$$pw \; = \; \frac{\delta}{2} \quad \text{and} \quad pR = \vartheta \tag{$\dagger$}$$

We now construct a new MDP $\mathcal{N}$ that extends $\mathcal{M}$ by a fresh state $fail'$ and an action τ that is enabled in state $fail$ with weight $wgt_{\mathcal{N}}(fail, \tau_i) = R$ and the transition probabilities $P_{\mathcal{N}}(fail, \tau, goal) = p$, $P_{\mathcal{N}}(fail, \tau, fail') = 1{-}p$. For all other states, the enabled actions and their transition probabilities and weights are the same as in $\mathcal{M}$.

We claim that $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} \geq \vartheta + \delta$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} > \vartheta + \frac{\delta}{2}$. Let us first observe that $\mathcal{M}$ and $\mathcal{N}$ have the same schedulers. Moreover, if $\mathfrak{S}$ is a scheduler and $y = \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond goal)$ then:

(1) $\Pr^{\mathfrak{S}}_{\mathcal{N},s_{init}}(\Diamond goal) \; = \; y + p(1{-}y)$

(2) $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) \quad \leq \quad \mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \quad < \quad \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) + \frac{\delta}{2}$

Proof of (2): The claim is obvious if $y = 1$ because then $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} = \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} = \mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}}$.

Suppose now that $y < 1$. As the accumulated weight of all paths from s_{init} to $fail$ is at most $w{-}1$ we have:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + pR(1{-}y) \;\; \leq \;\; \mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \;\; < \;\; \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + p(R{+}w)(1{-}y)$$

The claim then follows from (†).

Suppose now that $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$. Pick a deterministic scheduler $\mathfrak{S}$ such that $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} > \vartheta$. Thus, with $y = \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal)$ we have $y > 0$ and $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} > \vartheta y$. By (*) we have

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \ \ \geq \ \ \vartheta y + \delta.$$

Using the first inequality of statement (2) we obtain:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \ \ \overset{(2)}{\geq} \ \ \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) \ \ \geq \ \ \vartheta y \ + \ \delta \ + \ \vartheta \ - \ \vartheta y \ \ = \ \ \vartheta + \delta.$$

Hence, $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} \geq \vartheta + \delta$. Suppose now that $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} > \vartheta + \frac{\delta}{2}$. Pick a deterministic scheduler $\mathfrak{S}$ such that $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} > \vartheta + \frac{\delta}{2}$. Let $y \overset{\text{def}}{=} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal)$. The assumption $\Pr^{\min}_{\mathcal{M},s_{init}}(\lozenge goal) > 0$ yields $y > 0$. Using the second inequality of statement (2) we obtain:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) + \frac{\delta}{2} \ \ \overset{(2)}{>} \ \ \mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \ \ > \ \ \vartheta + \frac{\delta}{2}.$$

This yields:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \ \ > \ \ \vartheta y$$

But then $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} > \vartheta$, and therefore $\mathbb{CE}^{\max}_{\mathcal{N},t_{init}} > \vartheta$. So, $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} \geq \vartheta + \delta$ iff $\mathbb{PE}^{\max}_{\mathcal{N},t_{init}} > \vartheta + \frac{\delta}{2}$.

With a similar argument, we will show that also $\mathbb{CE}^{\min}_{\mathcal{M},s_{init}} < \vartheta$ iff $\mathbb{PE}^{\min}_{\mathcal{N},t_{init}} \leq \vartheta - \frac{\delta}{2}$ iff $\mathbb{PE}^{\min}_{\mathcal{N},t_{init}} < \vartheta - \frac{\delta}{2}$. Instead of (*), we use here the following analogue fact:

$$\text{If } y = \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal) \text{ and } \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} < \vartheta y \text{ then } \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \leq \vartheta y - \delta. \qquad (**)$$

Suppose first that $\mathbb{CE}^{\min}_{\mathcal{M},s_{init}} < \vartheta$. Pick a deterministic scheduler $\mathfrak{S}$ such that $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} < \vartheta$ and let $y = \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal)$. Then, $y > 0$ and $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} < \vartheta y$. By (**) we get $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \ \leq \ \vartheta y{-}\delta$. We now rely on the second inequality of statement (2) and obtain:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \ \ \overset{(2)}{<} \ \ \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) + \frac{\delta}{2} \ \ \leq \ \ \vartheta y \ - \ \delta \ + \ \vartheta \ - \ \vartheta y \ + \ \frac{\delta}{2} \ \ = \ \ \vartheta - \frac{\delta}{2}$$

Hence, $\mathbb{PE}^{\min}_{\mathcal{N},t_{init}} < \vartheta - \frac{\delta}{2}$. Suppose now that $\mathbb{PE}^{\min}_{\mathcal{N},t_{init}} \leq \vartheta - \frac{\delta}{2}$. Pick a deterministic scheduler $\mathfrak{S}$ such that $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \leq \vartheta - \frac{\delta}{2}$. By assumption, $y \overset{\text{def}}{=} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\lozenge goal) > 0$. The first inequality of statement (2) yields:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} + \vartheta(1{-}y) \ \ \overset{(2)}{\leq} \ \ \mathbb{PE}^{\mathfrak{S}}_{\mathcal{N}} \ \ \leq \ \ \vartheta - \frac{\delta}{2}$$

Hence:

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} \ \ \leq \ \ \vartheta y - \frac{\delta}{2} \ \ < \ \ \vartheta y$$

But then $\mathbb{CE}^{\mathfrak{S}}_{\mathcal{M}} < \vartheta$, which implies $\mathbb{CE}^{\min}_{\mathcal{N},t_{init}} < \vartheta$. □

This reduction now allows us to conclude PSPACE-hardness also for all threshold problems for partial expectations:

Theorem 3.32. *All threshold problems for maximal or minimal partial expectations in acyclic MDPs with non-negative integer weights are PSPACE-hard.*

Proof. Immediate from Theorem 3.30 and Lemma 3.31. □

3.6 Conditional value-at-risk

We turn our attention to the conditional value-at-risk. The conditional value-at-risk is a measure focusing on the tail of the distribution of a random variable. The value-at-risk is the worst p-quantile, i.e., the value such that p of the outcomes are worse. The value-at-risk provides the best bound on the p worst outcomes, but it is not affected by the distribution of outliers. While a pure worst-case analysis is overly pessimistic in a probabilistic setting, the conditional value-at-risk is a good compromise taking outliers with their respective probabilities into account. It is defined as the expectation of X under the condition that the outcome is worse than the value-at-risk. In Figure 3.7, the value-at-risk and the conditional value-at-risk for some value p and two distributions of a random variable X are illustrated for which high outcomes are the bad cases.

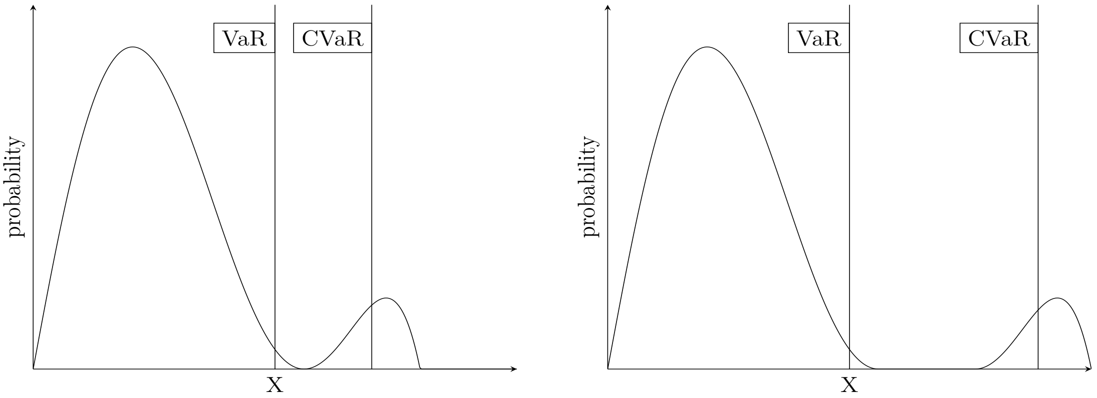


Figure 3.7: Illustration of value-at-risk (VaR) and conditional value-at-risk (CVaR).

For MDPs, the distribution of a random variable depends on the chosen scheduler. Providing guarantees on the worst- or best-case conditional value-at-risk hence turns into an optimization problem again. In the context of weighted MDPs, the conditional value-at-risk has been studied for mean-payoffs and for weighted reachability where on each run only once a terminal weight is collected when a target state is reached [KM18]. In the sequel, we consider the conditional value-at-risk for the more general accumulated weight before reaching the goal, i.e. for the classical stochastic shortest path problem. To the best of our knowledge, this problem has not been studied. For MDPs with non-negative weights, we provide a simple saturation point that allows the computation of worst- and best-case conditional values-at-risk in exponential time. We return to the problem with arbitrary weights in Chapter 5.

Formal Definition. Given an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ with a scheduler $\mathfrak{S}$, a random variable X defined on runs of the MDP with values in $\mathbb{R}$ and a value $p \in [0,1]$, we define the value-at-risk as $\mathit{VaR}^{\mathfrak{S}}_p(X) = \inf\{r \in \mathbb{R} | \Pr^{\mathfrak{S}}_{\mathcal{M}}(X \geq r) \leq p\}$. So, the value-at-risk is the point at which the cumulative distribution function of X reaches or exceeds $1-p$. The conditional value-at-risk is now the expectation of X under the condition that the outcome belongs to the p worst outcomes. Denote $\mathit{VaR}^{\mathfrak{S}}_p(X)$ by v. Following the treatment of random variables that are not continuous in general in [KM18], we define the conditional value-at-risk as follows:

$$\mathit{CVaR}^{\mathfrak{S}}_p(X) = 1/p(\Pr^{\mathfrak{S}}_{\mathcal{M}}(X > v) \cdot \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(X | X > v) + (p - \Pr^{\mathfrak{S}}_{\mathcal{M}}(X > v)) \cdot v).$$

Outcomes of X which are greater than v are treated differently to outcomes equal to v as it is possible that the outcome v has positive probability and we only want to account exactly for the p worst outcomes. Hence, we take only $p - \Pr^{\mathfrak{S}}_{\mathcal{M}}(X > v)$ of the outcomes which are exactly v into account as well. To provide worst-case guarantees or to find risk-averse policies, we are interested in the maximal and minimal conditional value-at-risk

$$\mathit{CVaR}^{\max}_p(X) = \sup_{\mathfrak{S}} \mathit{CVaR}^{\mathfrak{S}}_p(X) \text{ and } \mathit{CVaR}^{\min}_p(X) = \inf_{\mathfrak{S}} \mathit{CVaR}^{\mathfrak{S}}_p(X).$$

In our formulation here, high outcomes are considered to be bad. Completely analogously, one can define the conditional value-at-risk for the lowest p outcomes. If it is not clear from context, we will sometimes write $\mathit{CVaR}^{\mathfrak{S}}_{\uparrow,p}(X)$ to denote the conditional value-at-risk as defined here, and $\mathit{CVaR}^{\mathfrak{S}}_{\downarrow,p}(X)$ to denote the analogous value if low values are considered to be bad.

Conditional value-at-risk for the classical SSPP. The random variable for which we want to study the conditional value-at-risk in MDPs is $\oplus \mathit{Goal}$, the accumulated weight before reaching a goal state. Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ be an MDP and assume $\mathbb{E}^{\max}_{\mathcal{M},s_{\mathit{init}}}(\oplus \mathit{Goal}) < \infty$. Further, we assume that all weights are non-negative here. After the pre-processing, we can assume that there are no non-trivial end components anymore. Note that removing 0-end components does not affect the possible probability distributions over path lengths of schedulers that reach Goal with probability 1. In the following theorem we will show how to compute worst-case conditional values-at-risk. Afterwards, we sketch how to treat the case where low outcomes are considered bad.

Theorem 3.33. *Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ with non-negative weights and no non-trivial end-components as well as a rational probability value $p \in (0,1)$, the value $CVaR^{\max}_{\uparrow,p}(\oplus goal)$ is computable in exponential time.*

Proof. Let N be the number of states of $\mathcal{M}$, δ be the minimal non-zero transition probability, and W the maximal weight occurring in $\mathcal{M}$. As there are no end components, *Goal* is reached within N steps from any other state under any scheduler with probability at least δ^N. Let ℓ be such that $(1-\delta^N)^\ell \leq p$. Note that ℓ simply has to be chosen bigger than $\frac{\log(p)}{\log(1-\delta^N)}$ and hence can be computed in polynomial time. So, its numerical value is at most of exponential size. Then, the probability that a path accumulates a weight higher than $K = \ell \cdot N \cdot W$ is less than p under any scheduler. So, the value-at-risk $VaR^{\mathfrak{S}}_p(\oplus goal)$ is less than K under any scheduler $\mathfrak{S}$. So all paths that reach an accumulated weight of at least K certainly belong to the p worst paths under any scheduler. On these paths, the best thing to do in order to maximize the conditional value-at-risk is to maximize the expected accumulated weight before reaching the goal. We know that these optimal values are computable in polynomial time. Hence, we will assign weight $wgt(\pi) + \mathbb{E}^{\max}_{\mathcal{M},last(\pi)}(\oplus)$ to paths π that exceed an accumulated weight of K in their last step. We can now reduce the problem to a conditional value-at-risk problem for weighted reachability. We can achieve this by explicitly encoding the accumulated weight up to K into the state space:

We define a new MDP $\mathcal{N}$ with a set of weighted target states as follows: The state space S' is $S \times \{0, \ldots, K+W-1\}$. The initial state s'_{init} is $(s_{init}, 0)$. The set of actions stays the same. The transition probability function P' is defined by $P'((s,i),\alpha,(t,j) = P(s,\alpha,t)$ if $i + wgt(s,\alpha) = j$ and $i < K$, and $P'((s,i),\alpha,(t,j)) = 0$ otherwise. There is no weight function in $\mathcal{N}$, but instead a set of weighted target states. The target states are $(goal, i)$ with weight i for all $i < K$ and (s,j) with weight $j\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus goal)$ for all $j \geq K$ and $s \in S$.

Now, we can compute the optimal conditional value-at-risk with the probability value p for the random variable assigning the terminal weight to a path with the methods for weighted reachability presented in [KM18] in polynomial time in the size of $\mathcal{N}$ to obtain the value $CVaR^{\max}_{\uparrow,p}(\oplus goal)$ in $\mathcal{M}$. The linear program presented there requires a guess of the value-at-risk. However, the value-at-risk in our setting is a natural number between 0 and K, so there are only exponentially many candidates. This results in an exponential time algorithm for our problem. □

If we consider paths with low weight to be bad, the problem is somewhat simpler in MDPs with non-negative weights. As in the proof of the theorem, we can compute a natural number K' such that the probability that at least weight K' is accumulated is less than $1-p$ under any scheduler. The value-at-risk for the probability p is then always less than K' and paths accumulating weight at least K' are certainly not among the worst p paths. We can assign weight K' to these paths without affecting the conditional value-at-risk. Encoding the accumulated weight into the state space allows us to reduce the problem to a conditional value-at-risk problem for weighted reachability in an MDP of exponential size yielding an exponential time algorithm again. No matter whether we consider high or low weights to be bad, the minimal conditional value-at-risk can be computed completely analogously via the presented construction.

CHAPTER

FOUR

LONG-RUN SATISFACTION OF PATH PROPERTIES

If a system goes through an intialization phase before working for a long time under normal operating conditions, the behavior of the system in the long-run is of particular interest. Checking whether the whole execution satisfies a specification given in LTL or another temporal logic allows to verify basic properties of the system. It is, e.g., possible to express properties like $\Box\Diamond\varphi$ – that a formula φ is satisfied infinitely often. In a messaging system for example, this can be used to check that every message is eventually processed. This check does, however, not provide any guarantees on the time needed to process a message or the fraction of time in which the system is ready to process messages. In this chapter, we investigate the optimization problems for three quantitative measures that address the long-run behavior of a system in order to answer such verification questions.

In the non-probabilistic setting, we study the notion of *long-run frequencies* for ω-regular properties. As the name suggests, long-run frequencies measure in the long-run how frequently a property holds and are expressed as the limit of the fraction of the number of suffixes that satisfy the property over the number of all suffixes. For finite-state transition systems, we study the optimization of the long-run frequency of a given property over all paths.

When turning to the probabilistic world, we introduce the corresponding concept of *long-run probabilities*. On Markov chains, long-run probabilities are limit-average probabilities for path properties, indicating the probability for a property to hold on the suffix of a path after many steps. Among others, this notion aims to provide refined measures for the system availability, understood as the proportion of time a system is functioning under "normal" operating conditions (after the initialization phase). For finite-state MDPs, the corresponding optimization problem is to compute the optimal long-run probability of a given property when ranging over all schedulers.

To address the long-run behavior with respect to quantitative aspects of a system such as resource consumption or utility, we define *long-run expectations* in a similar fashion. Long-run expectations describe the long-run average of the weight that is expected to be accumulated before a certain target state is reached for the next time. In a sense, it expresses the average distance to the next target state. For example, we can determine how long a system needs to finish its current task before being able to process a request that is sent at some unknown time after the system has been running for a long time.

While a relation to stochastic shortest path problems is obvious for long-run expectations, we will see that also long-run probabilities share a lot of similarities with variants of stochastic shortest path problems. To some extend, the same techniques as for stochastic shortest path problems are applicable to the optimization of long-run probabilities. To convey a first idea of these long-run notions and the difficulties arising, we illustrate the notion of long-run probability in an example.

Example 4.1. Consider the MDP $\mathcal{N}$ shown in Fig. 4.1. The only non-deterministic choice is the choice between actions α and β in state a. Action α yields a uniform distribution over the three successors.

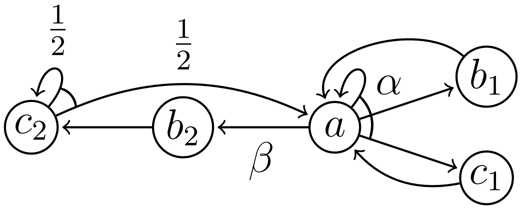


Figure 4.1: MDP with labels indicated by the state names requiring counting to maximize the long-run probability of $a \,\mathrm{U}\, b$.

We want to determine the maximal long-run probability of $a \,\mathrm{U}\, b$. Under the memoryless scheduler $\mathfrak{S}_\alpha$ that always picks action α, the probability of $a \,\mathrm{U}\, b$ in the a-state is $\frac{1}{2}$ under this scheduler. The states b_1 and c_1 appear equally often. The probability of $a \,\mathrm{U}\, b$ is 1 in state b_1 and 0 in state c_1. We thus conclude that the long-run probability under $\mathfrak{S}_\alpha$ is $\frac{1}{2}$. Similarly, the steady-state probability of the states a and b_2 under the memoryless scheduler $\mathfrak{S}_\beta$ are $\frac{1}{4}$, and the probability that $a \,\mathrm{U}\, b$ holds from there is 1. The long-run probability of $a \,\mathrm{U}\, b$ under $\mathfrak{S}_\beta$ equals $\frac{1}{2}$ as well. Interestingly, these two memoryless schedulers are not optimal. Consider the scheduler $\mathfrak{S}$ that chooses α first and, if it returns to a directly, chooses β afterwards. In the first visit to the a state, the probability for $a \,\mathrm{U}\, b$ is $\frac{2}{3}$. States b_1 and c_1 are reached with probability $1/3$ afterwards. If state a is reached again directly, the probability of $a \,\mathrm{U}\, b$ is now 1. Also state b_2 is reached with probability $1/3$ before returning to a from b_1, c_1, or c_2. Tho compute the long-run probability under this scheduler, we sum up the satisfaction probabilities for all states that can be visited before returning to a from b_1, c_1, or c_2 multiplied with the probabilities of the visits. We divide the result by the expected number of steps before returning. Note that we sum up probability $2/3 + 1/3 \cdot 1$ for the two possible visits to state a. We obtain a long-run probability of

$$\frac{2/3 + 1/3 + 1/3 + 1/3}{2/3 \cdot 2 + 1/3 \cdot 5} = \frac{5}{9}.$$

The intricate interaction of satisfaction probability and steady-state probability of each state makes the optimization of long-run probability particularly challenging. We will see that indeed counting the number of consecutive a-states and basing the decisions on the counter value is the key to achieve maximal long-run probabilities for $a \,\mathrm{U}\, b$. $\triangleleft$

Outline. We investigate long-run frequencies in Section 4.1. We show how the optimal long-run frequency of a regular co-safety property given by an NFA $\mathcal{A}$ in a transition system $\mathcal{T}$ can be computed in time polynomial in $\mathcal{T}$ and exponential in $\mathcal{A}$. An accompanying PSPACE-hardness result for the threshold problem implies that we cannot expect a considerable improvement of the upper bound. Section 4.2 addresses long-run probabilities in MDPs. After identifying easily solvable instances, we provide a construction allowing to express the optimal long-run probability of a regular co-safety property as the optimal expected mean payoff in an infinite-state MDP. This construction will also play a role in Chapter 5. For the special case of constrained reachability properties ($a \,\mathrm{U}\, b$), we prove the existence of a saturation point, a bound on the number of consecutive visits to a-states before an optimal scheduler can behave memorylessly until the set of a-states is left. This allows us to obtain a finite-state MDP from the general construction for co-safety properties and to compute the optimal long-run probability in exponential time. Again, we also provide a lower bound: the threshold problem for the long-run probability of constrained reachability properties is NP-hard. In Section 4.3, we introduce the notion of long-run expectation. Interestingly, we are able to provide a saturation point allowing the computation of optimal values in exponential time even in MDPs with arbitrary integer weights. The saturation point is not a bound on the accumulated weight, but on the number of steps without visiting a goal state here. The NP-hardness for long-run probabilities can easily be transferred to long-run expectations as well.

Related work. The notions we investigate follow the spirit of quantifying the validity of a property along a path as in frequency-LTL [BDL12, FK15, FKK15] and averaging LTL [BMM14]. In the work on frequency-LTL [BDL12], a quantitative variant $a \,\mathrm{U}_q\, b$ of the until operator relaxes the standard meaning of $a \,\mathrm{U}\, b$ by requiring that a holds at a fraction of at least q of the positions before b holds. Other variants of frequency-LTL [FK15, FKK15] allow only quantitative variants $\Box_q$ of the globally operator. The semantics here is that $\Box_q \phi$ holds on a path if the long-run average of the frequency of positions at which ϕ holds is at least q. Alternatively, averaging LTL [BMM14] rather than truth values, assigns quantities to pairs of paths and formula. It is based on a quantitative labeling function for atomic propositions and inductively defines the semantics of $\Box\varphi$ as the average of the value of φ along the path. Both the full logics have an undecidable model checking problem [BDL12, BMM14]. Decidable fragments of frequency-LTL can be obtained by restricting the nesting of temporal operators or the allowed frequency thresholds [BDL12, FK15, FKK15]. To contrast frequency-LTL with our notion of long-run probabilities note that frequency-LTL is a logic to specify quantitative measures for the satisfaction of properties along paths using the $\Box_q$-modality, while long-run probabilities are a quantitative measure across behaviors. For finite strongly connected Markov chains, the probabilities for $\Box_q$-formulas are 0 or 1, while long-run probabilities can be strictly between 0 and 1. There is still a close connection as for each finite, strongly connected Markov chain $\mathcal{M}$, $\Box_q(a \mathbin{\mathrm{U}} b)$ holds in $\mathcal{M}$ with probability 1 iff the long-run probability of $a \mathbin{\mathrm{U}} b$ is at least q. Nevertheless, the contribution for MDPs in [FK15, FKK15] are orthogonal to ours. On the one hand, they can treat much more complex properties with nested $\Box_q$-formulas. On the other hand, they cannot deal with formulas of the type $\Box_q(aUb)$ for $q < 1$. The results in [FK15] only apply to $q = 1$. The fragment in [FKK15] can deal with $\Box_q$-modalities for arbitrary q, but imposes the constraint that no until operator occurs in the scope of the $\Box_q$-modality.

Long-run probabilities can be seen as mean-payoff, where the weights are the satisfaction probabilities. A crucial difference however with mean-payoff and other long-run properties is that, for long-run probabilities, the "weights" along a path are not fixed *a priori*, but do depend on the scheduler. In this aspect, there is some conceptual relation to dynamic Markov processes [Pap85] where cost or transition probabilities depend on previously made decisions, or the stochastic variant of the Canadian traveler problem [FSBW13]. These problems, however, are concerned with finite-horizon objectives; moreover, their weights are affected by the past, whereas our "weights" (satisfaction probabilities) are induced by the future scheduler.

Many works address the long-run behavior of MDPs with respect to the accumulation of weights, e.g., mean payoffs [Kal83, CD11, BBC+14] and other cost objectives [dA98] or ratios [dA97, vEJ11]. To the best of our knowledge, long-run expectations have not been addressed in the literature.

Note on the publication of the results. This chapter is based on joint work with Christel Baier, Nathalie Bertrand, and Ocan Sankur published at LICS 2019 [BBPS19]. The construction for the long-run probabilities of regular co-safety properties in MDPs has been added (Section 4.2.2). Furthermore, we introduce the notion of long-run expectation here and the results of Section 4.3 have not been published.

4.1 Non-probabilistic long-run frequencies

Intuitively, the frequency with which a path property is satisfied on a path quantifies the fraction of the number of suffixes that satisfy the path property over the number of all suffixes. In this section, we investigate the optimization of long-run frequencies in non-probabilistic models. Transitions systems are a standard model for systems that exhibit non-deterministic, but no probabilistic behavior. As this is the only section in which we work with non-probabilistic models, we first define our notation for transitions systems. We assume familiarity with the model.

Definition 4.2. A transition system is a tuple $\mathcal{T} = (S, \Delta, \mathsf{AP}, \mathsf{L})$ where S is a finite set of states, $\Delta \subseteq S \times S$ is a transition relation, AP is a finite set of atomic propositions, and $\mathsf{L}\colon S \to 2^{\mathsf{AP}}$ is a labeling function. An infinite path in a transition system is a sequence $\pi = s_0 s_1 s_2 \ldots$ of states such that $(s_i, s_{i+1}) \in \Delta$ for all i. The trace of a path $\pi = s_0 s_1 s_2 \ldots$ is the word $L(s_0)L(s_1)L(s_2)\ldots$ from $(2^{\mathsf{AP}})^\omega$. If we talk about the properties a path satisfies, we do not distinguish between the path and its trace.

Formally, the long-run frequency is now defined as follows.

Definition 4.3. Let $\mathcal{T} = (S, \Delta, \mathsf{AP}, \mathsf{L})$ be a transition system and φ a path property. The *long-run frequency* of φ along an infinite path ζ of $\mathcal{T}$ is defined as:

$$\mathit{lrf}_\varphi(\zeta) \quad = \quad \liminf_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} \mathbb{1}_{\zeta_{[i\ldots]}\models\varphi}$$

where $\mathbb{1}_{\zeta_{[i\ldots]}\models\varphi}$ is 1 if $\zeta_{[i\ldots]} \models \varphi$ and 0 otherwise. The *maximal long-run frequency* of φ is given by

$$\mathbb{LF}^{\max}_{\mathcal{T},s}(\varphi) \;=\; \sup_{\zeta} \mathit{lrf}_\varphi(\zeta)$$

where $s \in S$ and ζ ranges over all infinite paths starting in state s. The *minimal long-run frequency* $\mathbb{LF}^{\min}_{\mathcal{T},s}(\varphi)$ is defined analogously.

The value $\mathit{lrf}_\varphi(\zeta)$ is not affected by the addition of a finite prefix to a path, and hence all states belonging to the same strongly connected component (SCC) of $\mathcal{T}$ have the same extremal values. It thus suffices to determine the optimal values for the SCCs of $\mathcal{T}$. The optimal value for a given state s of $\mathcal{T}$ is then the maximum or minimum of the optimal values of the SCCs reachable from s. In the sequel we therefore assume $\mathcal{T}$ is strongly connected, and simply write $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi)$ and $\mathbb{LF}^{\min}_{\mathcal{T}}(\varphi)$.

As a consequence of a result established later for the probabilistic setting (see Theorem 4.11), the extremal long-run frequencies for invariants, reachability, and Rabin and Streett conditions are computable in polynomial time. For transition system, these techniques essentially require to identify "good" cycles ξ where the property under consideration holds from all states along ξ.

Reasoning about long-run frequencies becomes more challenging when considering properties that are not prefix-independent and where a classification of cycles into good and bad ones is not sufficient. As a stepping stone in this direction, we consider regular co-safety properties where satisfaction is witnessed by “good” prefixes in this section. We illustrate the problem in the following example for the constrained reachability property $a\,\mathrm{U}\,b$ which is a simple co-safety property for which the problem already becomes interesting.

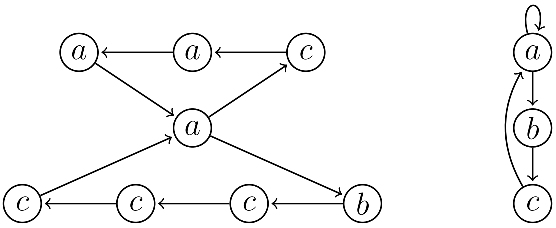


Figure 4.2: Transition systems requiring memory to maximize the long-run frequency of $a\,\mathrm{U}\,b$.

Example 4.4. Fig. 4.2 gives two examples of transition system labeled with atomic propositions a, b, and c on which one wants to evaluate the long-run frequency of an until property. The transition system on the left illustrates, that, memory is required to maximize the long-run frequency: For $a\,\mathrm{U}\,b$, the maximal long-run frequency is achieved by alternating between the two simple cycles and amounts to $\frac{4}{9}$, which is indeed more than $\frac{2}{5}$ the long-run frequency of iterating the bottom cycle only. For the transition system on the right, the long run frequency of $a\,\mathrm{U}\,b$ along *e.g.* the path $(abc)^\omega$ is $\frac{2}{3}$. The maximal long-run frequency is 1, which is achieved, *e.g.*, by the infinite path $abca^2bca^4bca^8bc\ldots$ that successively doubles the number of times the self-loop at state a is taken. However, there is no finite-memory strategy for generating an infinite path where the long-run frequency for $a\,\mathrm{U}\,b$ is 1.

Let us take a first glimpse at a construction with which these results can be obtained algorithmically. In Figure 4.3, we create two copies of each state labeled a that we label with 0 and 1, respectively. From copies labeled $(a, 0)$, we remove all outgoing edges to states labeled with b or to states labeled with $(a, 1)$. Likewise, we remove all edges from copies labeled $(a, 1)$ to states labeled c or $(a, 0)$. The label 1 is supposed to indicate that $a\,\mathrm{U}\,b$ will hold from a state on. If there are no cycles consisting only of states labeled a, the construction indeed ensures that this holds on all infinite paths in the constructed transition system. For the transition system on the left, we can hence assign weight 1 to all states labeled with $(a, 1)$ or b and weight 0 to all other states as $a\,\mathrm{U}\,b$ holds exactly on the suffixes of an infinite path that start in a state with weight 1. The long-run frequency of $a\,\mathrm{U}\,b$ is then equal to the maximal mean payoff in the constructed structure.

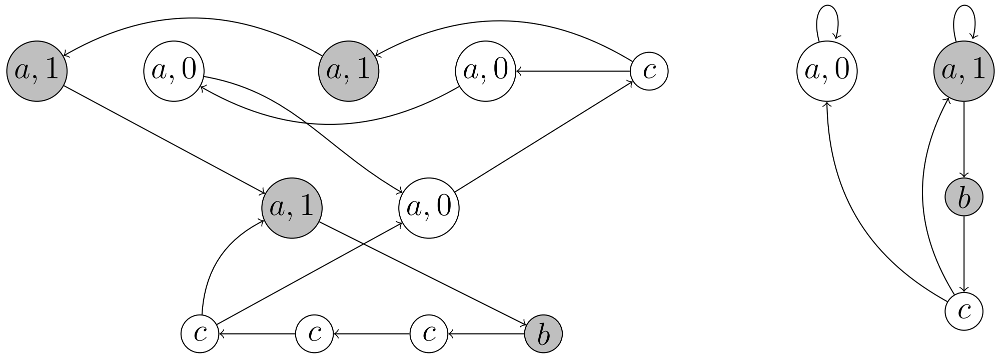


Figure 4.3: Weighted transition systems obtained for the transition system from Fig. 4.2: gray states have weight 1, while others have weight 0.

The transition system on the right contains a cycle of a-states, namely the self-loop on the upper state. Now, maximizing the mean payoff in the constructed transition system does not necessarily maximize the long-run frequency of $a \,\mathrm{U}\, b$. The mean payoff of the path looping in state $(a, 1)$ is 1 while no suffix satisfies $a \,\mathrm{U}\, b$. To overcome this problem, we mix the cycle maximizing the mean payoff with a cycle making sure that $a \,\mathrm{U}\, b$ is satisfied from states labeled with 1, in this case the cycle $(a, 1)bc$. Using infinite memory, we can make sure that the frequency of this cycle in the path is 0 although it occurs infinitely often. This is exactly what happens in the path $(a, 1)bc(a, 1)^2bc(a, 1)^4bc(a, 1)^8bc \ldots$. ◁

Regular co-safety properties. We now address extremal long-run frequencies in transition system for regular co-safety properties φ. Fix $\mathcal{T} = (S, \Delta, \mathsf{AP}, \mathsf{L})$, a strongly connected transition system, and let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ be an NFA over the alphabet $\Sigma = 2^{\mathsf{AP}}$ representing φ, *i.e.*, an infinite path of $\mathcal{T}$ satisfies φ iff it has a prefix accepted by $\mathcal{A}$. Hence, $\mathit{lrf}_{\varphi}(\zeta)$, also denoted $\mathit{lrf}_{\mathcal{A}}(\zeta)$, is the long-run average of positions in ζ where a word in $\mathcal{L}(\mathcal{A})$ starts, and we write $\mathbb{LF}^*_{\mathcal{T}}(\mathcal{A})$ rather than $\mathbb{LF}^*_{\mathcal{T}}(\varphi)$. We show that the computation of $\mathbb{LF}^{\max}_{\mathcal{T}}(\mathcal{A})$ and $\mathbb{LF}^{\min}_{\mathcal{T}}(\mathcal{A})$ reduces to determine the maximal and minimal mean-payoff in a weighted transition system $\mathcal{G}$ with a generalized Büchi side condition. The size of this transition system is exponential in the size of $\mathcal{T}$.

For simplicity, we suppose here that $Q_0 = \{q_0\}$ is a singleton and that q_0 is not accessible from any other state in $\mathcal{A}$. We also assume that $q_0 \notin F$ (otherwise $\mathcal{A}$ accepts the empty word and the long-run frequency of φ along any infinite path is 1). We fix an arbitrary state $s_0 \in S$ which we treat as a starting state of $\mathcal{T}$. (Since $\mathcal{T}$ is strongly connected, the extremal long-run frequencies in $\mathcal{T}$ do not depend on the choice of the starting state.) We define a weighted transition system $\mathcal{G}$ as follows. Let $\ell = |Q|$ denote the number of states in the NFA $\mathcal{A}$. Then, the state space $S_{\mathcal{G}}$ of $\mathcal{G}$ is equal to

$$S \times (\{\#\} \cup Q)^{\ell+1} \times 2^Q$$

while we require that at least one of the entries in $(\{\#\} \cup Q)^{\ell+1}$ is #. The idea is that in each step of a run of the original transition system $\mathcal{T}$, we non-deterministically choose whether the suffix starting with that step is accepted by $\mathcal{A}$ or not. We can view this as a promise about the future choices we will make. In the states of $\mathcal{G}$, the information on runs of the automaton $\mathcal{A}$ that still have to be accepted or rejected is stored which allows us to check that the promises on suffixes accepted or rejected by $\mathcal{A}$ were correct in each step. In the $(\{\#\} \cup Q)^{\ell+1}$-component, we store the current state of up to $\ell + 1$ runs that still have to be accepted in $\ell + 1$ tracks available in the component. The symbol # denotes tracks currently not in use. As soon as two of these runs reach the same state, we can stop keeping track of one of the runs. Hence we always have at most ℓ runs to check. As a new run that is accepting can be started at any step, it is convenient to have an additional track to start the new run. In the 2^Q-component, we store all possible states of runs that have to be rejected. Note that in order to make sure that the NFA $\mathcal{A}$ does not accept a suffix of a run in $\mathcal{T}$, we have to keep track of *all* successor states in $\mathcal{A}$ after each step while for the accepted suffixes it is sufficient to specify *one* successor state after each step. The transition relation in $\mathcal{G}$ will make sure that from no state currently in the 2^Q-component, an accepting state can be reached. Hence, we do not have to distinguish individual runs, but can simply keep track of the set of possible states. For the accepting runs on the other hand, we have to make sure that all of them are eventually accepted. Hence, we store the runs explicitly in the ℓ tracks provided in the $(\{\#\} \cup Q)^{\ell}$-component. A generalized Büchi condition will make sure that all of these runs are in fact accepting.

Let us define the transition relation $\Delta_{\mathcal{G}}$ of $\mathcal{G}$. Let us do that by describing how the possible successors of a state $g = (s, (p_0, \dots, p_\ell), A)$ with $p_0, \dots, p_\ell \in \{\#\} \cup Q$ and $A \subseteq Q$ are determined. For a state $(t, (r_0, \dots, r_\ell), B)$ to be a possible successor of g, we first require that $(s, t) \in \Delta$ is a possible transition in $\mathcal{T}$. Now there are two possible promises whether the suffix of the run of $\mathcal{T}$ starting in state t now should be accepted by $\mathcal{A}$.

If it is supposed to be accepted, a successor is constructed as follows: The set $B = \bigcup_{q \in A} \delta(q, L(s))$. The transition is only possible if B is disjoint from F. To obtain a possible tuple $(r_0, \dots, r_\ell)$, first choose $r'_i \in \delta(p_i, L(s))$ for all i with $p_i \in Q$ and let $r'_i = \#$ if $p_i = \#$. Then, do the following:

1. Let $r_i = q_0$ if $r'_i = \#$ and $r'_j \neq \#$ for all $j < i$,

2. let $r_i = \#$ if $r'_i \in F$ or if $r'_i = r'_j$ for some $j < i$,

3. for all remaining indices i, let $r_i = r'_i$.

Note that by the condition that at least one of the $\ell + 1$ tracks contains # in each state, step 1 is always possible. Furthermore, step 2 merges tracks which contain the same state and makes sure that there is at least one track containing # after the transition as there are only ℓ distinct states in Q.

If the suffix starting currently is not supposed to be accepted, let $B = \bigcup_{q \in A} \delta(q, L(t)) \cup \{q_0\}$ and require it to be disjoint from F, and skip step 1 in the construction of the possible tuples $(r_0, \dots, r_\ell)$.

Note that if in $\mathcal{A}$ one of the states in the set A has an $L(s)$-successor in F, the state g has no successors. This makes sure that the promise that a suffix will not be accepted by $\mathcal{A}$ cannot be violated on infinite paths. To make sure that also the positive promises about acceptance are met, we have to employ a generalized Büchi condition. Whenever a run stored in one of the $\ell + 1$ tracks is accepted or joined with a track of lower index, it contains # afterwards. Let $\#_i$ denote the set of states where the ith of the $\ell + 1$ tracks contains #. The generalized Büchi condition

$$\Phi = \bigwedge_{i=0}^{\ell} \Box \Diamond \#_i$$

makes sure that this happens infinitely often on each track. As we join tracks only with tracks of lower index, this ensures that all positive promises are eventually made true.

To conclude the construction of $\mathcal{G}$, we let the initial states be

$$(s_0, (q_0, \#, \dots, \#), \emptyset) \text{ and } (s_0, (\#, \#, \dots, \#), \{q_0\})$$

and assign weight +1 to all states containing q_0 in one of the tracks in the $(\{\#\} \cup Q)^{\ell+1}$-component while all other states get weight 0.

Lemma 4.5. *Let $\mathcal{T}$ and $\mathcal{A}$ be given and let $\mathcal{G}$ and Φ be constructed from $\mathcal{T}$ and $\mathcal{A}$ as above. For each infinite path ζ in $\mathcal{T}$, there is a path ξ in $\mathcal{G}$ satisfying Φ – and vice versa – such that $lrf_{\mathcal{A}}(\zeta) = MP(\xi)$ where $MP(\xi)$ is the mean payoff of ξ according to the weight function of $\mathcal{G}$.*

Proof. Consider any infinite path $\zeta = (s_k)_{k \geq 0}$ of $\mathcal{T}$ that starts at s_0. We will construct a path $\xi = (s_k, (r_0^k, \dots, r_\ell^k), A_k)_{k \geq 0}$ in $\mathcal{G}$ satisfying Φ and whose mean payoff is exactly $lrf_{\varphi}(\zeta)$. Note that the path follows ζ in its first component. In each step, we make the promise that the suffix will be accepted by $\mathcal{A}$ iff this is in fact the case for the suffix of ζ starting at that step. The only non-deterministic choices that remain to be made are the successors r_0^{k+1} if $r_0^k \in Q$. As we only promise suffixes that are accepted by $\mathcal{A}$ to be accepted, we can simply follow an accepting run of the suffix in each of these tracks $r_j^\cdot$. The switch to # when the run is accepted or merged with a run of lower index is determined by the construction of possible successors. Also the third component is uniquely determined by the construction and the fact that we add q_0 whenever the suffix of ζ starting now is not accepted by $\mathcal{A}$. We now have that $\zeta[i \dots]$ is accepted by $\mathcal{A}$ if and only if the weight of $\xi[i]$ is 1. So, we indeed have $lrf_{\mathcal{A}}(\zeta) = MP(\xi)$. Further, we only promise a run to be accepting if it is indeed eventually accepted. So, each of the tracks $r_j^\cdot$ is set back to # infinitely often and so $\xi \models \Phi$.

Conversely, consider a path $\xi = (s_k, (r_0^k, \dots, r_\ell^k), A_k)_{k \geq 0}$ of $\mathcal{G}$ satisfying Φ. Let $\zeta = (s_k)_{k\geq 0}$ be the corresponding path in $\mathcal{T}$. If the weight of $\xi[i]$ is 1, the lowest indexed track $r_j^{\cdot}$ that contained $\#$ at r_j^{i-1} is set to q_0. To obtain a run of $\mathcal{A}$ witnessing that the suffix $\zeta[i\dots]$ is accepted by $\mathcal{A}$ in this case, we can follow the run $(r_j^m)_{m\geq i}$ provided in this track until the track is set to $\#$. As $\xi \vDash \Phi$, this will happen. Then, the run either reached an accepting state or was merged with a lower indexed track. In the latter case, we follow the run of this lower indexed track from then on. After at most j-many merges to another track, an accepting state will be reached.

If the weight of $\xi[i]$ is 0, the state q_0 is added to A_i. In each step, all possible successor states are added to this set and none of these successor states can ever be accepting as this is prohibited in the transition relation of $\mathcal{G}$. So, $\zeta[i\dots]$ is not accepted by $\mathcal{A}$. Hence, we can conclude that $\mathit{lrf}_{\mathcal{A}}(\zeta) = \mathit{MP}(\xi)$ again. □

Using [BCHK11,Kar78], one can compute the extremal mean payoff values in $\mathcal{G}$ under the generalized Büchi side condition Φ. The procedure looks for reachable SCCs $\mathcal{E}$ in $\mathcal{G}$ in which Φ can be satisfied. As this only requires the SCCs to contain one state satisfying $\#_i$ for each i, this can be done in time polynomial in the size of $\mathcal{G}$. In each of these components $\mathcal{E}$, we then compute the maximal mean-payoff in that component in polynomial time. The maximum M among these values is also the maximal mean payoff $\mathit{MP}_{\mathcal{G}}^{\max}(\Phi)$ in $\mathcal{G}$ under the side condition Φ: Clearly, $\mathit{MP}_{\mathcal{G}}^{\max}(\Phi) \leq M$. On the other hand from a cycle σ with mean payoff M in one of these reachable components $\mathcal{E}$, we can construct a path satisfying Φ with the same mean payoff. We let ρ be a cycle that starts in the first state of σ and visits states satisfying $\#_i$ for each $0 \leq i \leq \ell$. The path $\sigma^1\rho\sigma^2\rho\sigma^4\rho\dots$ satisfies Φ and has mean payoff M because the frequency of ρ in this path is 0. The minimal mean payoff under a generalized Büchi side condition can be treated analogously. This allows us to conclude the following result as $\mathcal{G}$ is of size polynomial in $\mathcal{T}$ and exponential in $\mathcal{A}$.

Theorem 4.6. *$\mathbb{LF}_{\mathcal{T},s}^{\max}(\varphi)$ and $\mathbb{LF}_{\mathcal{T},s}^{\min}(\varphi)$ are computable in time exponential in the size of $\mathcal{A}$ and polynomial in the size of $\mathcal{T}$.*

For a fixed regular co-safety property, the optimal long-run frequency can hence be computed in polynomial time. In particular, this applies to constrained reachability properties:

Corollary 4.7. *The values $\mathbb{LF}_{\mathcal{T},s}^{\max}(a\,\mathrm{U}\,b)$ and $\mathbb{LF}_{\mathcal{T},s}^{\min}(a\,\mathrm{U}\,b)$ are computable in polynomial time.*

We now establish the complexity of the decision problem associated with the maximization of long-run frequency for regular co-safety properties. Formally, given a transition system $\mathcal{T}$, an NFA $\mathcal{A}$, a rational threshold ϑ, and a comparison operator $\bowtie \in \{\geq, >, <, \leq\}$, the threshold problem asks whether $\mathbb{LF}_{\mathcal{T}}^{\max}(\varphi) \bowtie \vartheta$.

Theorem 4.8. *The threshold problem for the long-run frequency of regular co-safety properties in transition system is PSPACE-complete.*

We split the proof into two parts in the following two lemmata:

Lemma 4.9. *Given a transition system $\mathcal{T}$, an NFA $\mathcal{A}$, a rational threshold ϑ, and a comparison operator $\bowtie \in \{\geq, >, <, \leq\}$, one can check in polynomial space whether $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) \bowtie \vartheta$.*

Proof. The idea is to use the weighted transition system $\mathcal{G}$, without constructing $\mathcal{G}$ explicitly. Let all notation concerning $\mathcal{G}$ be as above, in particular ℓ is the number of states of Q. As PSPACE equals NPSPACE (Savitch's theorem), it suffices to provide a non-deterministic polynomially space-bounded procedure to check whether $\mathbb{MP}^{\max}_{\mathcal{G}}(\Phi) \bowtie \vartheta$.

We first show how to solve the threshold problem for $\bowtie \in \{\geq, >\}$. In this case, we have to check the existence of *one* path in $\mathcal{G}$ satisfying the constraint. The algorithm starts by guessing a state $s_{\mathcal{G}}$ in $\mathcal{G}$ and checks whether $s_{\mathcal{G}}$ is reachable from one of the initial states. This check can be done in polynomial space because all states of $\mathcal{G}$ have a polynomially large representation and checking whether a state t is a possible successor of a state s can be done in polynomial time. Then, it checks whether for each $0 \leq i \leq \ell$, there is a state labeled $\#_i$ in the same SCC as $s_{\mathcal{G}}$. This requires again two reachability checks that can be done in polynomial space. If these conditions are met, the algorithm checks whether there is a simple cycle ξ_{MP} containing $s_{\mathcal{G}}$ with mean payoff of $\bowtie \vartheta$. As the length of a simple cycle is bounded by the number of states of $\mathcal{G}$ which is at most $N \cdot (\ell+1)^{\ell+1} \cdot 2^{\ell}$ where N is the number of states of $\mathcal{T}$, such a cycle can be guessed by counting the number of steps up to at most this bound while keeping track of the sum of the weights in the guessed cycle. These values can be stored in polynomial space as all weights are 0 or 1. Once, $s_{\mathcal{G}}$ is reached again, this non-deterministic algorithm terminates if the mean payoff of the guessed cycle satisfies the constraint that it is $\bowtie \vartheta$. If the algorithm finds such a cycle, we have seen how to obtain a path ζ in $\mathcal{G}$ and hence in $\mathcal{T}$ with $\mathit{lrf}_{\mathcal{A}}(\zeta) \bowtie \vartheta$.

As PSPACE is closed under complement, the problem for $\bowtie \in \{\leq, <\}$ can also be solved in polynomial space: $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) \leq \vartheta$ iff $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) \not> \vartheta$, and $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) < \vartheta$ iff $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) \not\geq \vartheta$. □

Lemma 4.10. *The threshold problem "given a transition system $\mathcal{T}$, an NFA $\mathcal{A}$ and a rational threshold ϑ, decide whether $\mathbb{LF}^{\max}_{\mathcal{T}}(\varphi) \geq \vartheta$" is PSPACE-hard.*

Proof. The PSPACE lower bound follows by a polynomial reduction from the intersection problem for deterministic finite automata (DFA): given k DFA $\mathcal{D}_1, \ldots, \mathcal{D}_k$ over the same alphabet Σ, is the intersection language $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ nonempty? This problem is known to be PSPACE-complete [Koz77].

To provide a polynomial reduction from the intersection problem for DFA, we suppose we are given DFA $\mathcal{D}_1, \ldots, \mathcal{D}_k$ over some alphabet Σ. W.l.o.g. we may assume that $k \geq 2$ and that the empty word is not included in any of the languages $\mathcal{L}(\mathcal{D}_i)$. Let Q_i be the state space of $\mathcal{D}_i$, $\ell_i = |Q_i|$ and $\ell = \ell_1 \cdot \ldots \ell_k$. Then, $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ is nonempty if and only if there is a word $w \in \Sigma^*$ of length at most ℓ such that $w \in \mathcal{L}(\mathcal{D}_i)$ for $i = 1, \ldots, k$.

Let $\$_1, \ldots, \$_k, \#$ be pairwise distinct fresh letters (not contained in Σ), and let $\Gamma = \Sigma \cup \{\$_1, \ldots, \$_k, \#\}$. Given a finite word $w = \sigma_1 \sigma_2 \ldots \sigma_n \in \Sigma^+$, let $\hat{w}$ denote the word over $\Sigma \cup \{\#\}$ that arises from w by inserting $(k{-}1)$-times the symbol $\#$ after each letter σ_j. That is,

$$\hat{w} \quad = \quad \sigma_1 \#^{k-1} \sigma_2 \#^{k-1} \ldots \sigma_n \#^{k-1}$$

For $i = 1, \ldots, k$, one can easily construct in time $\mathcal{O}(k^2 + k \cdot \mathit{size}(\mathcal{D}_i))$ a new DFA $\mathcal{B}_i$ over the alphabet Γ such that:

$$\mathcal{L}(\mathcal{B}_i) = \Big\{ \$_i^j \, \$_{i+1}^{k-1} \ldots \$_k^{k-1} \, \hat{w} \ : \ w \in \mathcal{L}(\mathcal{D}_i), 1 \leqslant j < k \Big\}$$

Furthermore, we can construct in time linear in the sizes of $\mathcal{B}_1, \ldots, \mathcal{B}_k$ an NFA $\mathcal{A}$ over the alphabet Γ with:

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B}_1) \cup \ldots \cup \mathcal{L}(\mathcal{B}_k) \cup \{\#^i : i \geq 1\}$$

Note that $\mathcal{A}$ does not accept the empty word and no word starting with a letter in Σ. Likewise, we can construct in time polynomial in k a strongly connected KS $\mathcal{T}$ with the following states:

- $s_{i,j}$ for $i = 1, \ldots, k$ and $j = 1, \ldots, k{-}1$,
- $t_1, \ldots, t_{k-1}$ and
- u_σ for each symbol $\sigma \in \Sigma$.

We treat the symbols in Γ as atomic propositions and identify the singletons $\{\gamma\}$ with γ, where γ ranges over all symbols of the alphabet Γ. The labeling function of $\mathcal{T}$ is then given by:

$$\mathsf{L}(s_{i,j}) = \$_i, \ \mathsf{L}(t_j) = \# \text{ and } \mathsf{L}(u_\sigma) = \sigma.$$

The transition relation of $\mathcal{T}$ is depicted in Figure 4.4. The words generated by $\mathcal{T}$ are the substrings of the infinite words $y_1 y_2 y_3 \ldots$ where each word y_i has the form

$$\$_1^{k-1} \$_2^{k-1} \ldots \$_k^{k-1} \hat{w}$$

for some $w \in \Sigma^+$. Let

$$\vartheta = \frac{k(k{-}1) + (k{-}1)\ell}{k(k{-}1) + k\ell}$$

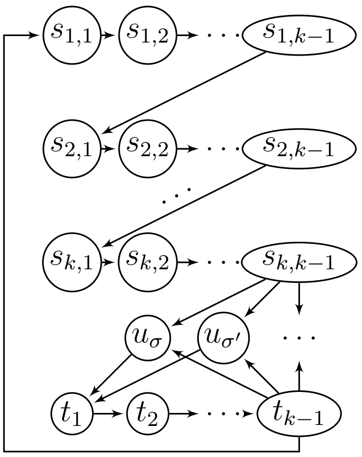


Figure 4.4: The Kripke structure $\mathcal{T}$ in the reduction.

Clearly, $\mathcal{T}, \mathcal{A}, \vartheta$ can be constructed in time polynomial in the size of the DFA $\mathcal{D}_1, \ldots, \mathcal{D}_k$. It remains to show that $\mathcal{T}$ has an infinite path ζ with long-run frequency $\mathit{lrf}_{\mathcal{A}}(\zeta)$ at least ϑ if and only if the intersection language of the $\mathcal{D}_i$'s is nonempty.

Let us recall that, formally, the relation of $\mathcal{T}$ consists of the following transitions:

$$\begin{array}{ll}
s_{i,1} \to s_{i,2} \to \ldots \to s_{i,k-1} & \text{for } i = 1, \ldots, k \\
s_{i,k-1} \to s_{i+1,1} & \text{for } i = 1, \ldots, k{-}1 \\
s_{k,k-1} \to u_\sigma \to t_1 & \text{for } \sigma \in \Sigma \\
t_1 \to t_2 \to \ldots \to t_{k-1} & \\
t_{k-1} \to s_{1,1} \text{ and } t_{k-1} \to u_\sigma & \text{for } \sigma \in \Sigma
\end{array}$$

Suppose first that there is some word $w \in \Sigma^*$ accepted by each of the DFSs $\mathcal{D}_1, \ldots, \mathcal{D}_k$. As stated before, we then can safely assume that $|w| \leqslant \ell$. $\mathcal{T}$ has a cycle ξ generating the word $v = \$_1^{k-1}\$_2^{k-1} \ldots \$_k^{k-1}\hat{w}$. We then have:

$$|v| \;=\; k(k{-}1) + k|w|$$

The word v contains exactly $k(k{-}1) + (k{-}1)|w|$ positions from which a word accepted by $\mathcal{A}$ starts. This follows from the following two observations:

- The suffixes $\$_i^j\$_{i+1}^{k-1} \ldots \$_k^{k-1}\hat{w}$ of v are accepted by $\mathcal{B}_i$, and therefore by $\mathcal{A}$ (for $j = 1, \ldots, k{-}1$).
- $\hat{w}$ contains exactly $(k{-}1) \cdot |w|$ positions from which a subword contained in $\#^+$ starts.

Thus, the long-run frequency of the infinite path ξ^ω that repeats this cycle ad infinity is:

$$\mathit{lrf}_{\mathcal{A}}(\xi^\omega) \;=\; f(|w|)$$

where $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is the following function:

$$f(x) \;=\; \frac{k(k{-}1) + (k{-}1)x}{k(k{-}1) + kx}$$

Function f is monotonically decreasing.[1] Therefore, $f(|w|) \geq f(\ell)$ As $f(\ell) = \vartheta$, we conclude that $\mathcal{T}$ has an infinite path with long-run frequency at least ϑ.

We assume now that $\mathcal{T}$ has an infinite path ζ with long-run frequency at least ϑ. We first observe that ζ must visit $s_{1,1}$ infinitely often as otherwise ζ would have an infinite suffix consisting of t- and u-states, in which case the long-run frequency would be smaller or equal than $(k{-}1)/k$, and therefore strictly smaller than ϑ. Suppose by contradiction $\mathcal{L}(\mathcal{D}_1) \cap \ldots \mathcal{L}(\mathcal{D}_k)$ is empty. Then, the average weight obtained by each cycle $s_{1,1}s_{1,2}\ldots s_{1,k-1}\ldots s_{k,1}s_{k,2}\ldots s_{k,k-1}\varrho s_{1,1}$ where ϱ consists of t- and u-states contained in ζ in $\mathcal{T}$ is less or equal

$$\frac{(k{-}1)^2 + (k{-}1)y}{k(k{-}1) + ky} \;=\; \frac{k{-}1}{k}$$

where y is the number of u-states in ϱ, in which case the number of t-states in ϱ is $(k{-}1)y$. But then again, the long-run frequency of ζ would be bounded by $(k{-}1)/k$, and therefore strictly smaller than ϑ. Contradiction. We conclude that $\mathcal{L}(\mathcal{D}_1) \cap \ldots \mathcal{L}(\mathcal{D}_k)$ must be nonempty if $\mathcal{T}$ has an infinite path with long-run frequency at least ϑ. □

This finishes the proof of Theorem 4.8. The hardness result establishes that we cannot expect an algorithm for the exact computation of maximal long-run frequencies for co-safety properties that runs in sub-exponential time.

4.2 Long-run probabilities

The aim of long-run probabilities is to provide a probabilistic notion with a similar spirit as long-run frequencies. Instead of counting the number of suffixes of a run that satisfy a path property φ, we take an average of the probability after each step that φ is satisfied on the suffix starting after that step. More formally, let $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, \mathsf{L})$ be an MDP and let φ be a path property. The *long-run probability* for φ of an infinite path ζ under a scheduler $\mathfrak{S}$ for $\mathcal{M}$ is defined as as the long-run average of the probabilities for φ in all positions of ζ with respect to the residual schedulers $\mathfrak{S}{\uparrow}\zeta[0\ldots i]$ defined by

$$\mathfrak{S}{\uparrow}\zeta[0\ldots i](\pi) = \mathfrak{S}(\zeta[0\ldots i]\cdot\pi)$$

[1] Each rational function $h(x) = (a + cx)/(b + dx)$ with $cb < ad$ is decreasing. This is a consequence of the fact the the first derivative is strictly negative. Note that $h'(x) = (cb - ad)/(b + dx)^2$, which is strictly negative if $cb < ad$.

for finite paths π starting in $\zeta[i]$:

$$lrp^{\mathfrak{S}}_{\varphi}(\zeta) \quad = \quad \liminf_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} \mathrm{Pr}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\varphi) \ .$$

The long-run probability for property φ under scheduler $\mathfrak{S}$ from state s, denoted $\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s}(\varphi)$, is defined as the expectation of the random variable $\zeta \mapsto lrp^{\mathfrak{S}}_{\varphi}(\zeta)$ under $\mathfrak{S}$ with starting state s:

$$\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s}(\varphi) = \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(lrp^{\mathfrak{S}}_{\varphi}).$$

We now address the task to compute the extremal long-run probabilities for φ:

$$\begin{aligned} \mathbb{LP}^{\max}_{\mathcal{M},s}(\varphi) &= \sup_{\mathfrak{S}} \mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s}(\varphi) \text{ and} \\ \mathbb{LP}^{\min}_{\mathcal{M},s}(\varphi) &= \inf_{\mathfrak{S}} \mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s}(\varphi) \end{aligned}$$

where $\mathfrak{S}$ ranges over all schedulers for $\mathcal{M}$. In contrast to classical optimization problems for MDPs, the random variable whose expectation we aim to optimize, namely $lrp^{\mathfrak{S}}_{\varphi}$, depends on the scheduler $\mathfrak{S}$ itself. In Example 4.1, we have already seen that this dependency of long-run probabilities on the scheduler in two different ways makes the optimization problem rather intricate.

In this section, we start with the identification of efficiently solvable instances, including the computation of long-run probabilities in Markov chains and prefix-independent properties (such as Rabin or Streett conditions) where the satisfaction only depends on the states that are visited infinitely often in MDPs. The latter can be treated by a polynomial-time analysis of end components. Afterwards, we turn our hands to regular co-safety properties. We provide a construction of an infinite-state MDP from an MDP and a regular co-safety property allowing us to express optimal long-run probabilities in terms of optimal expected mean payoffs. Due to the infinite state space, however, the construction does not lead to a procedure to compute the optimal values. Nevertheless, the construction is useful in two respects: On the one hand, it establishes a connection to non-classical stochastic shortest path problems that we exploit in Chapter 5 to prove that the threshold problem for the optimal long-run probability of regular co-safety properties is Positivity-hard and that we hence cannot expect to be able to compute the optimal values with known techniques. On the other hand, the construction turns out to allow us to compute optimal long-run probabilities for the special case of constrained reachability properties ($a \, \mathrm{U} \, b$). The key insight is the existence of a saturation point similar to the saturation points for non-classical stochastic shortest path problems with non-negative weights that allows us to obtain a finite-state MDP from the constructed infinite-state MDP. This result leads to an exponential-time algorithm for the computation of optimal long-run probabilities of constrained reachability properties. By proving that the corresponding threshold problem is NP-hard, we show that we cannot expect a polynomial computation algorithm.

It is important to emphasize that the computation of optimal long-run probabilities for constrained reachability does not easily reduce to reachability via a pre-processing of the MDP, as it typically does for most verification problems. Also, the traditional reduction to the case of a Rabin condition for the treatment of arbitrary ω-regular properties fails here. These highlight the challenge and specificity in computing long-run probabilities.

4.2.1 Efficiently solvable instances

We first investigate special cases for which one can obtain efficient algorithms to compute optimal long-run probabilities: we explain the case of Markov chains and we identify restricted classes of properties for MDPs.

Markov chains. For Markov chains, we simply write $lrp_\varphi(\zeta)$ for each infinite path ζ. If φ is an ω-regular property, for each bottom strongly connected component (BSCC) $\mathcal{B}$ of the Markov chain $\mathcal{M}$, the long-run probability for all states in $\mathcal{B}$ is the same:

$$\mathbb{LP}_{\mathcal{B}}(\varphi) \quad = \quad \sum_{t \in \mathcal{B}} \theta_t^{\mathcal{B}} \cdot \Pr_{\mathcal{M},t}(\varphi)$$

where $\theta_t^{\mathcal{B}}$ denotes the steady-state probability (defined as the long-run frequency) of state t in $\mathcal{B}$. Thus, $\mathbb{LP}_{\mathcal{B}}(\varphi)$ equals the probability for φ in $\mathcal{B}$ viewed as a Markov chain where the initial distribution is given by the long-run frequencies in $\mathcal{B}$, which again coincides with the expected mean payoff in $\mathcal{B}$ when $\Pr_{\mathcal{M},t}(\varphi)$ is viewed as weight for state t. The long-run frequencies inside the BSCC are computable in polynomial-time using a linear equation system. The values $\Pr_{\mathcal{M},t}(\varphi)$ for the states inside the BSCC are computable using standard techniques for the analysis of Markov chains against ω-regular properties (see *e.g.* [BK08]). The complexity depends on the type and representation of φ: for instance, exponential-time algorithms exist for LTL formulas [CY95]. Thus, long-run probabilities for LTL-properties in Markov chains are computable in exponential time. Moreover, $\mathbb{LP}_{\mathcal{M},s}(\varphi)$ is computable in polynomial time for those properties φ where $\Pr_{\mathcal{M},t}(\varphi)$ is computable in polynomial time, such as constrained reachability properties.

Alternatively, the computation of long-run probabilities as expected mean payoff when dealing with the weight function that assigns weight $wgt(s) = \Pr_{\mathcal{B},s}(\varphi)$ to each state s can also be written as a quotient of expectations as follows. Let s be an arbitrary state in $\mathcal{B}$, called *reference state*. Then, the long-run probability for φ in $\mathcal{B}$ equals the quotient of the expected accumulated weight along paths of length at least 1 from s until returning to s and the expected return time (i.e., expected number of steps) along such paths from

s to s. Strong connectivity ensures that both expectations are finite. That is,

$$\mathbb{LP}_{\mathcal{B}}(\varphi) \;=\; \frac{\mathbb{E}_{\mathcal{B},s}(\text{“weight until } s\text{”})}{\mathbb{E}_{\mathcal{B},s}(\text{“steps until } s\text{”})}$$

Finally, if $\mathfrak{B}$ denotes the set of all BSCCs of $\mathcal{M}$ then for each state s in $\mathcal{M}$:

$$\mathbb{LP}_{\mathcal{M},s}(\varphi) \;=\; \sum_{\mathcal{B}\in\mathfrak{B}} \Pr_{\mathcal{M},s}(\Diamond\mathcal{B}) \cdot \mathbb{LP}_{\mathcal{B}}(\varphi) \;.$$

Efficiently solvable instances on MDPs. We now identify classes of path properties for which the optimal long-run probability is computable in polynomial time. Analogously to extremal long-run frequencies in transition systems, we can assume MDPs to be strongly connected in the sequel. For general MPDs, the optimal value can be computed by first computing the optimal value in each maximal end component and afterwards solving a weighted reachability problem on the MEC-quotient. When $\mathcal{M}$ is strongly connected, the optimal long-run probabilities do not depend on the starting state and we simply write $\mathbb{LP}^{\max}_{\mathcal{M}}(\varphi)$ and $\mathbb{LP}^{\min}_{\mathcal{M}}(\varphi)$.

Theorem 4.11. *Let $\mathcal{M}$ be an MDP. The values $\mathbb{LP}^{\max}_{\mathcal{M},s}(\varphi)$ and $\mathbb{LP}^{\min}_{\mathcal{M},s}(\varphi)$ are computable in polynomial-time if φ is a condition of one the following types:*

- *reachability $\Diamond b$,*
- *invariance $\Box b$,*
- *generalized Rabin $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{\ell_i} (\Box\Diamond b_{i,j} \wedge \Diamond\Box a_{i,j})$*
- *or Streett $\bigwedge_{i=1}^{n} (\Box\Diamond a_{i,j} \to \Box\Diamond b_{i,j})$.*

In all these cases, optimal deterministic finite-memory schedulers exist. Moreover, optimal memoryless deterministic schedulers exist for reachability, invariances, Büchi and co-Büchi conditions.

Proof. We provide the argument for $\mathbb{LP}^{\max}_{\mathcal{M},s}(\varphi)$. The argument for $\mathbb{LP}^{\min}_{\mathcal{M},s}(\varphi)$ is analogous and omitted here. As stated above, we may assume that $\mathcal{M}$ is strongly connected. It is well-known [dA97, BGC09, CGK13] that for all properties listed in the theorem there is a deterministic finite-memory scheduler $\mathfrak{S}$ that maximizes the probability for φ from every visited state in the following sense:

$$\Pr^{\mathfrak{S}\uparrow\varsigma[0\ldots i]}_{\mathcal{M},\varsigma[i]}(\varphi) \;=\; \Pr^{\max}_{\mathcal{M},\varsigma[i]}(\varphi)$$

for each infinite $\mathfrak{S}$-path ς and each position $i \in \mathbb{N}$. For reachability, invariances, Büchi and co-Büchi conditions, we may even suppose that $\mathfrak{S}$ is memoryless deterministic with a single BSCC $\mathcal{B}$.

If φ is a reachability, generalized Rabin or Streett condition then $\Pr^{\max}_{\mathcal{M},s}(\varphi) = \Pr^{\max}_{\mathcal{M},t}(\varphi)$ for all states s, t in $\mathcal{M}$. Moreover, this value is either 0 or 1. But then $\mathfrak{S}$ obviously achieves the maximal long-run probability from every state.

The states in $\mathcal{M}$ can have different maximal probabilities for invariances $\varphi = \Box b$. However, for invariances we either have $\max_{s \in S} \Pr^{\max}_{\mathcal{M},s}(\varphi) = 0$, in which case $\mathbb{LP}^{\max}_{\mathcal{M},s}(\varphi) = 0$ for all states s, or the unique BSCC $\mathcal{B}$ of $\mathfrak{S}$ consists of b-states. In the latter case, $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Box b) = \Pr^{\max}_{\mathcal{M},s}(\Box b) = 1$ for all states s in $\mathcal{B}$. Let now $\mathfrak{T}$ be the following memoryless deterministic scheduler:

- From the states not in $\mathcal{E}$, $\mathfrak{T}$ mimics a memoryless deterministic scheduler maximizing the probability to reach $\mathcal{B}$ (which is 1 as $\mathcal{M}$ is strongly connected).
- For the states inside $\mathcal{B}$, $\mathfrak{T}$ behaves as $\mathfrak{S}$.

We then have $\mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s}(\Box b) = 1$ for all states s in $\mathcal{M}$, which is obviously maximal. □

An analogous result for much richer classes of properties cannot be expected given that, already in the non-probabilistic setting, infinite-memory can be necessary for until properties (see Fig. 4.2 right), and co-safety properties yield PSPACE-hardness (see Theorem 4.8).

4.2.2 Construction for general co-safety properties

In the non-probabilistic setting, we reduced the computation of maximal long-run frequencies of regular co-safety properties to the computation of maximal mean payoffs. Now, we want to transfer this idea to the computation of maximal long-run probabilities. The situation, however, becomes more involved. Non-deterministic automata are not well-suited for product constructions with MDPs. So, we will use deterministic finite-automata (DFA) to express co-safety properties. The main idea is to construct an MDP with extended state space that keeps track of the number of runs currently in each state of the automaton. We can then assign a weight to each step of the MDP depending on how many runs of the DFA enter an accepting state during that step. We prove that the optimal mean payoff in the constructed MDP coincides with the optimal long-run probability in the original MDP.

However, there is no bound on the number of runs we have to store in the state space for this construction. Therefore, the constructed MDP will have an infinite state space. In the next section, we will see that constrained reachability properties constitute a special case in which we can prove the existence of a saturation point – in this case a bound on the number of runs we have to store for the maximization of the long-run probability. This allows us to cut away states from the infinite state space to obtain a finite MDP of exponential size and hence an exponential-time algorithm for the computation of maximal long-run probabilities of constrained reachability properties.

Construction Let $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, L)$ be a strongly connected MDP and let $\mathcal{D} = (Q, 2^{\mathsf{AP}}, \delta, q_0, F)$ be a DFA over AP. As we are interested in the co-safety property given by $\mathcal{D}$, only runs of $\mathcal{D}$ up to the first accepting state are relevant. Hence, we can collapse all accepting states of $\mathcal{D}$ to one absorbing state *accept* and all states form which *accept* is not reachable to one state *reject*. Let the set of states $Q = \{q_0, q_1, \dots, q_\ell, \mathit{accept}, \mathit{reject}\}$ for some $\ell \in \mathbb{N}$.

We construct a weighted infinite-state MDP $\mathcal{M}_{\mathcal{D}} = (S', Act, P', s'_{init}, wgt)$ in the sequel. The state space is

$$S' = S \times \mathbb{N}^{\ell+1}.$$

The $\ell + 1$ natural numbers in a state store the number of runs of $\mathcal{D}$ on suffixes of the path produced by the MDP so far that are in the respective state of $\mathcal{D}$. The actions Act are the same as in $\mathcal{M}$. For the transition probability function P' we define the following: Let $s' = (s, n_0, \dots, n_\ell)$ and $t' = (t, m_0, \dots, m_\ell)$ be states such that for all i,

$$m_i = \iota_i + \sum_{j:\delta(q_j, L(t)) = q_i} n_j$$

where $\iota_i = 1$ if $i = 0$ and $\iota_i = 0$ otherwise. For such states, we set

$$P'(s', \alpha, t') = P(s, \alpha, t).$$

All other transition probabilities are 0. The weight function does not work on state-weight pairs as usual, but on single transitions in $S' \times Act \times S'$. For a transition (s', α, t') with $s' = (s, n_0, \dots, n_\ell)$ and $t' = (t, m_0, \dots, m_\ell)$, the weight is defined by

$$wgt(s', \alpha, t') = \sum_{j:\delta(q_j, L(t)) = \mathit{accept}} n_j.$$

To obtain a weight function on state-weight pairs, one could now take the weighted average over all possible transitions that can be taken via a state action-pair. As we will be interested in the mean payoff under this weight function, this change would not influence the subsequent considerations. The initial state s'_{init} is $(s_{init}, 1, 0, \dots, 0)$.

We observe that the sum of the entries in the last $\ell + 1$ components increases by at most 1 in each step. Hence, the total accumulated weight after n steps along any path is bounded by n and we can already conclude that the mean payoff in $\mathcal{M}_{\mathcal{D}}$ is bounded by 1 along each path.

A scheduler for $\mathcal{M}$ can be used as a scheduler for $\mathcal{M}_\mathcal{D}$ and vice versa as transitions in $\mathcal{M}_\mathcal{D}$ are uniquely defined by the transitions in the $\mathcal{M}$ component. If we consider a scheduler $\mathfrak{S}$ for both $\mathcal{M}$ and $\mathcal{M}_\mathcal{D}$, there is, however, one caveat: If $\mathfrak{S}$ is a finite memory-scheduler for $\mathcal{M}_\mathcal{D}$, the same scheduler is not necessarily a finite-memory scheduler for $\mathcal{M}$. So, we want to emphasize that the following lemma states that the maximal mean payoff in $\mathcal{M}_\mathcal{D}$ can be approximated by schedulers that are still finite-memory schedulers when considered as schedulers for $\mathcal{M}$.

Lemma 4.12. *Let $\mathcal{M}$ and $\mathcal{D}$ be given as above and let $\mathcal{M}_\mathcal{D}$ be the constructed MDP. For each scheduler $\mathfrak{T}$ for $\mathcal{M}_\mathcal{D}$ and each $\varepsilon > 0$, there is a finite-memory scheduler $\mathfrak{F}$ for $\mathcal{M}$ such that, if $\mathfrak{F}$ is seen as a scheduler for $\mathcal{M}_\mathcal{D}$:*

$$\mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP) - \varepsilon.$$

Proof. Let X_i be the random variable on paths of $\mathcal{M}_\mathcal{D}$ that assigns the weight $wgt(\zeta[i])$ collected in the ith step to a path ζ. By Fatou's lemma, we have:

$$\begin{aligned} \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP) &= \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}\left(\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} X_i\right) \\ &\leq \liminf_{n\to\infty}\ \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}\left(\frac{1}{n+1}\sum_{i=0}^{n} X_i\right). \end{aligned}$$

So, there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$:

$$\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i\right) \ \geq\ \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP) - \frac{\varepsilon}{2}.$$

Let $\mathfrak{Q}$ be a memoryless scheduler that minimizes the number of steps to reach a state in $\{s_{init}\} \times \mathbb{N}^{\ell+1}$. This scheduler can be obtained by interpreting a memoryless scheduler for $\mathcal{M}$ that minimizes the number of steps to reach s_{init} as a scheduler for $\mathcal{M}_\mathcal{D}$. Let $T = \max_{t\in S} \mathbb{E}^{\min}_{\mathcal{M},t}(\text{“steps until } s_{init}\text{”})$. So, also in $\mathcal{M}_\mathcal{D}$, the scheduler $\mathfrak{Q}$ requires at most T steps in expectation to reach a state in $\{s_{init}\} \times \mathbb{N}^{\ell+1}$.

We now construct a finite-memory scheduler $\mathfrak{F}$ satisfying the claim of the lemma. First, choose a natural number k with $k \geq k_0$ and $k > \frac{2T}{\varepsilon}$. The behavior of scheduler $\mathfrak{F}$ is as follows. In its first mode, it starts in s'_{init} and behaves like $\mathfrak{T}$ in the first k steps. Then, it switches to the second mode and behaves like $\mathfrak{Q}$ until it reaches a state in $\{s_{init}\} \times \mathbb{N}^{\ell}$. Afterwards, it switches back to the behavior of $\mathfrak{T}$ for k steps as if it was in state s'_{init}, and so on. The state it is really in might have entries larger than $(1, 0, \ldots, 0)$ in the $\ell + 1$ components storing numbers of runs of $\mathcal{D}$. The weights that will be accumulated before the scheduler switches to the behavior of $\mathfrak{Q}$ again, can only increase for a different vector in these components and hence we can take $\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_\mathcal{D},s'_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i\right)$ as a lower bound for the weight accumulated in each of the periods of k steps in which the scheduler acts like $\mathfrak{T}$. Furthermore, the expected number of steps which $\mathfrak{F}$ takes to follow $\mathfrak{T}$ for k steps and to return to s via $\mathfrak{Q}$ is at most $k{+}1+T$. Expressing the mean payoff of $\mathfrak{F}$ as a quotient, we obtain:

$$\begin{aligned}\mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}},s_{init}}(MP) &\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i\right)}{k+T}\\ &\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i\right)}{k(1+\varepsilon/2)}\\ &\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i\right)}{k}(1-\varepsilon/2)\\ &\geq (\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}(MP)-\varepsilon/2)\cdot(1-\varepsilon/2)\end{aligned}$$

by the choice that $k > 2T/\varepsilon$. Using the fact that $\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}(MP)$ is bounded by 1 we obtain:

$$\begin{aligned}&\mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}},s_{init}}(MP)\\ &\quad\geq (\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}(MP)-\varepsilon/2)\cdot(1-\varepsilon/2)\\ &\quad\geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}(MP)-\varepsilon.\end{aligned}$$

The constructed scheduler does not make use of the information stored in the states of $\mathcal{M}_{\mathcal{D}}$ except for the current $\mathcal{M}$-state. In the first phase, it explicitly keeps track of the full path for a finite number of steps. Afterwards, it acts as a finite memory scheduler on the $\mathcal{M}$-component. So, the scheduler is indeed a finite-memory scheduler when viewed as a scheduler for $\mathcal{M}$. □

The lemma implies that the maximal expected mean payoff can be expressed as the supremum over all finite-memory schedulers (for $\mathcal{M}$, viewed as schedulers for $\mathcal{M}_{\mathcal{D}}$). We want to carry out the analogous argument to prove that the maximal long-run probabilities can be approximated by finite-memory schedulers as well. There is one additional difficulty that we have to take care of: If we start with an arbitrary scheduler $\mathfrak{T}$ for $\mathcal{M}$ and construct a finite-memory scheduler $\mathfrak{F}$ that acts like $\mathfrak{T}$ for some number k of steps before switching to a different behavior, the probability that suffixes that have started within the first k steps but were not yet accepted by $\mathcal{D}$ will be accepted by $\mathcal{D}$ changes. For the argument to work, we have to make sure that these probabilities do not decrease. We use a result from [EKVY07] on MDPs with multiple reachability objectives, to obtain the following lemma that we will employ to achieve this.

Lemma 4.13. *Let $\mathcal{M}$ and $\mathcal{D}$ be given as above. For a state $q \in Q$ of $\mathcal{D}$, denote by $\mathcal{D}_q$ the DFA obtained from $\mathcal{D}$ by changing the starting state to q. For a scheduler $\mathfrak{S}$, write $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\mathcal{D}_q)$ to denote the probability under $\mathfrak{S}$ that a path starting in s is accepted by $\mathcal{D}_q$. Then, for each scheduler $\mathfrak{S}$, there is a finite-memory scheduler $\mathfrak{R}$ such that for each*

$q \in Q$, we have

$$\Pr^{\mathfrak{R}}_{\mathcal{M},s}(\mathcal{D}_q) \geq \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\mathcal{D}_q).$$

Proof. We construct an MDP $\mathcal{L}$ in which the events $\mathcal{D}_q$ can be expressed as reachability properties. This MDP $\mathcal{L}$ is simply the product of $\mathcal{M}$ with all of the automata $\mathcal{D}_q$:

$$\mathcal{L} = \mathcal{M} \otimes \Pi_{q \in Q} \mathcal{D}_q.$$

For each q, the q-component of $\mathcal{L}$ keeps track of the run of $\mathcal{D}_q$ on the path generated in the first component, i.e., in $\mathcal{M}$. Let accept_q denote all states in which the q-component contains the state *accept*. Now, the run of $\mathcal{M}$ in the first component is accepted by $\mathcal{D}_q$ if and only if the corresponding run in $\mathcal{L}$ satisfies $\Diamond \mathit{accept}_q$. For MDPs with multiple reachability objectives, [EKVY07, Theorem 2] applied to this MDP and these reachability objectives states that for each scheduler $\mathfrak{S}$, there is a memoryless scheduler $\mathfrak{F}$ such that

$$\Pr^{\mathfrak{F}}_{\mathcal{L}}(\Diamond \mathit{accept}_q) \geq \Pr^{\mathfrak{S}}_{\mathcal{L}}(\Diamond \mathit{accept}_q)$$

for all $q \in Q$. If we view this memoryless scheduler as a scheduler for $\mathcal{M}$, this is a finite-memory scheduler with memory modes in $Q^{|Q|}$. □

Using this result, we are now able to prove the analogue to Lemma 4.12 for long-run probabilities.

Lemma 4.14. *Let $\mathcal{M}$ and $\mathcal{D}$ be given as above. For each scheduler $\mathfrak{T}$ for $\mathcal{M}$ and each $\varepsilon > 0$, there is a finite-memory scheduler $\mathfrak{F}$ for $\mathcal{M}$ such that:*

$$\mathbb{LP}^{\mathfrak{F}}_{\mathcal{M},s_{init}}(\mathcal{D}) \geq \mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) - \varepsilon.$$

Proof. For a scheduler $\mathfrak{S}$, let $X_i^{\mathfrak{S}}$ be the random variable that assigns $\Pr^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\mathcal{D})$ to an infinite path ζ. Let $\mathfrak{T}$ be an arbitrary scheduler. By Fatou's lemma, we have:

$$\begin{aligned}\mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) &= \mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\liminf_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} X_i^{\mathfrak{T}}) \\ &\leq \liminf_{n\to\infty} \mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{n+1}\sum_{i=0}^{n} X_i^{\mathfrak{T}}\right).\end{aligned}$$

So, there exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$:

$$\mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{n+1}\sum_{i=0}^{n} X_i^{\mathfrak{T}}\right) \geq \mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) - \varepsilon/2.$$

As in the proof of Lemma 4.12, we want to follow the behavior of $\mathfrak{T}$ for some number $k \geq k_0$ of steps before returning to the initial state via a different scheduler. However, the random variables $X_i^{\cdot}$ depend on the scheduler and would be affected by this change. Hence, we first have to make sure that the value added up in these first k steps is not decreased by the change of the scheduler. For each finite $\mathfrak{T}$-path π, we define a scheduler $\mathfrak{R}_\pi$ using Lemma 4.13: The scheduler $\mathfrak{R}_\pi$ is a finite memory scheduler that satisfies:

$$\mathrm{Pr}^{\mathfrak{R}_\pi}_{\mathcal{M},last(\pi)}(\mathcal{D}_q) \geq \mathrm{Pr}^{\mathfrak{T}\uparrow\pi}_{\mathcal{M},last(\pi)}(\mathcal{D}_q) \tag{$\dagger$}$$

for all states $q \in Q$ of $\mathcal{D}$. Furthermore, let $\mathfrak{Q}$ be a memoryless scheduler that minimizes the number of steps to reach s_{init}. For $k \geq k_0$, we construct a finite-memory scheduler $\mathfrak{F}_k$ as follows: It follows $\mathfrak{T}$ for k steps and keeps track of the path π that is created. Then, it switches to the behavior of $\mathfrak{R}_\pi$ until a BSCC in the Markov chain induced by $\mathfrak{R}_\pi$ and $\mathcal{M}$ is reached and hence for all $q \in Q$, the suffix created by $\mathfrak{R}_\pi$ was accepted by $\mathcal{D}_q$ if it would be accepted eventually. Then it returns to the initial state s_{init} via $\mathfrak{Q}$ and starts all over again.

For $i \leq k$, we now have that $X_i^{\mathfrak{T}} \leq X_i^{\mathfrak{F}_k}$: Due to equation ($\dagger$), we see that no matter in which state of $\mathcal{D}$ the run on the suffix that started in the ith step of a k step long path π is, the probability that it will be accepted is not decreased by switching to the behavior of $\mathfrak{R}_\pi$ and following this scheduler until it is decided whether a path will be accepted or rejected by $\mathcal{D}_q$ for all q. This is what we make sure by following the behavior of $\mathfrak{R}_\pi$ until a BSCC in the induced Markov chain is reached.

Let $T = \max_{t\in S} \mathbb{E}^{\mathfrak{Q}}_{\mathcal{M},t}$(“steps until s_{init}”). Further, observe that all schedulers $\mathfrak{R}_\pi$ use $|Q|^{|Q|}$-memory modes that are deterministically updated and hence there are only finitely many possible schedulers. Let R be the maximal expected time until a BSCC is reached in the induced Markov chain under such a scheduler.

We can now express the expected long-run probability of $\mathfrak{F}_k$ by taking

$$\mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i^{\mathfrak{T}}\right)$$

as a lower bound on the probabilities that are added up in expectation before $\mathfrak{F}$ returned to the initial situation via $\mathfrak{Q}$ and $k + T + R$ as an upper bound on the expected time required to return. We obtain

$$\begin{aligned}
\mathbb{LP}^{\mathfrak{T}_k}_{\mathcal{M},s_{init}}(\mathcal{D}) &\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i^{\mathfrak{T}}\right)}{k+T+R} \\
&\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i^{\mathfrak{T}}\right)}{k(1+\varepsilon/2)} \\
&\geq \frac{\mathbb{E}^{\mathfrak{T}}_{\mathcal{M},s_{init}}\left(\frac{1}{k+1}\sum_{i=0}^{k} X_i^{\mathfrak{T}}\right)}{k}(1-\varepsilon/2) \\
&\geq (\mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) - \varepsilon/2)\cdot(1-\varepsilon/2)
\end{aligned}$$

by the choice that $k > 2(T+R)/\varepsilon$. Using the fact that $\mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D})$ is bounded by 1 we obtain:

$$\begin{aligned} &\mathbb{LP}^{\mathfrak{F}_k}_{\mathcal{M},s_{init}}(\mathcal{D}) \\ &\quad \geq \ (\mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) - \varepsilon/2)\cdot(1-\varepsilon/2) \\ &\quad \geq \ \mathbb{LP}^{\mathfrak{T}}_{\mathcal{M},s_{init}}(\mathcal{D}) - \varepsilon. \end{aligned}$$

This completes the proof. □

In the following lemma, we will now see that for finite-memory schedulers for $\mathcal{M}$ the long-run probability in $\mathcal{M}$ and the expected mean payoff in $\mathcal{M}_{\mathcal{D}}$ indeed agree.

Lemma 4.15. *Let $\mathcal{M}$ and $\mathcal{D}$ be given as above and let $\mathcal{M}_{\mathcal{D}}$ be the constructed MDP. Then, for each finite-memory scheduler $\mathfrak{S}$ for $\mathcal{M}$ (also viewed as a scheduler for $\mathcal{M}_{\mathcal{D}}$), we have $\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\mathcal{D}) = \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}(MP)$.*

Proof. Let $\mathcal{B}$ be the finite-state Markov chain induced by the finite-memory scheduler $\mathfrak{S}$ for $\mathcal{M}$. Let B be the set of states of $\mathcal{B}$. These states are of the form $\mathfrak{s} = (s,x)$ where $s \in S$ is a state of $\mathcal{M}$ and x a memory mode of $\mathfrak{S}$. Let $(\theta^{\mathcal{B}}_{\mathfrak{s}})_{\mathfrak{s}\in B}$ the long-run distribution in this Markov chain.

For the long-run probability under $\mathfrak{S}$ in $\mathcal{M}$, we can now rely on these steady state probabilities and obtain

$$\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\mathcal{D}) = \sum_{\mathfrak{s}\in B} \theta^{\mathcal{B}}_{\mathfrak{s}} \cdot \Pr_{\mathcal{B},\mathfrak{s}}(\mathcal{D}).$$

For the mean payoff in $\mathcal{M}_{\mathcal{D}}$, recall that after a finite path π, the value 1 is added to the q_0-component in the state $last(\pi)$. Afterwards this value 1 is moved along the transitions of $\mathcal{D}$. If it ever reaches an accepting state, it results in the reception of weight $+1$ along this path. So, the probability that this weight will eventually be received is precisely $\Pr_{\mathcal{B},(s,x)}(\mathcal{D})$ where s is the $\mathcal{M}$ component of $last(\pi)$ and x the memory mode of $\mathfrak{S}$ after π. As $\mathfrak{S}$ is a finite-memory scheduler, we furthermore now that the expected number of steps it takes between the addition of 1 to the q_0 and the reception of the weight $+1$ or the entrance to a BSCC in which the tracked run on $\mathcal{D}$ will not be accepted is finite from each pair (s,x) on. Let X_i be the random variable that assigns the weight in the ith step to a run and Y_i be the random variable that assigns the probability that the suffix after i steps will be accepted by $\mathcal{D}$ under the residual scheduler of $\mathfrak{S}$ after the finite path constructed so far. The fact that the expected time until it is decided whether the 1 added to q_0 will lead to the reception of weight 0 or $+1$, implies now that

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_{\mathcal{D}},s'_{init}}\left(\frac{1}{n+1}\sum_{i=0}^{n} X_i\right)$$

and

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_\mathcal{D},s'_{init}}\left(\frac{1}{n+1}\sum_{i=0}^{n} Y_i\right)$$

converge to the same value for $n \to \infty$ (while the second sum is at least as big as the first one for all n). So, we conclude that also

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP) = \sum_{\mathfrak{s}\in B} \theta^{\mathcal{B}}_{\mathfrak{s}} \cdot \Pr_{\mathcal{B},\mathfrak{s}}(\mathcal{D}). \qquad \square$$

The Lemmata 4.12, 4.14, and 4.15 together let us conclude the following theorem:

Theorem 4.16. *Let $\mathcal{M}$ and $\mathcal{D}$ be as above. Let $\mathcal{M}_\mathcal{D}$ be the infinite-state MDP constructed from $\mathcal{M}$ and $\mathcal{D}$ as described above. Then,*

$$\mathbb{LP}^{\max}_{\mathcal{M},s_{init}}(\mathcal{D}) = \mathbb{E}^{\max}_{\mathcal{M}_\mathcal{D},s'_{init}}(MP).$$

Of course, the construction presented here does not allow us to compute maximal long-run probabilities as the constructed MDP $\mathcal{M}_\mathcal{D}$ has an infinite state space. The possibility to express maximal long-run probabilities in terms of maximal mean payoffs, however, discloses the connections of long-run probabilities and stochastic shortest path problems. In the sequel, these connections will become apparent when we prove the existence of saturation points for a special case of long-run probabilities in the next section and when we prove Positivity-hardness of the threshold problem for long-run probabilities in Chapter 5.

4.2.3 Long-run probabilities for constrained reachability properties

Constrained reachability properties constitute a simple special case of regular co-safety properties. In this section, we show that we can adapt the key result for the solution of non-classical stochastic shortest path problems in MDPs with non-negative weights – the existence of a saturation point – to long-run probabilities of constrained reachability properties. Together with the construction for general co-safety properties we just presented, this result allows us to express optimal long-run probabilities in terms of the optimal expected mean payoff in a finite-state MDP. To illustrate the difficulties we encounter in more detail, we consider the following variant of the earlier Example 4.1.

Example 4.17. Consider the MDP $\mathcal{N}_k$ shown in Fig. 4.5. The only non-deterministic choice is the choice between actions α and β in state a. Action α yields a uniform distribution over the three successors.

In Example 4.1, we already hinted at the fact that counting the consecutive visits to a-states is the key for the optimization of the long-run probability of $a \,\mathrm{U}\, b$. Consider the

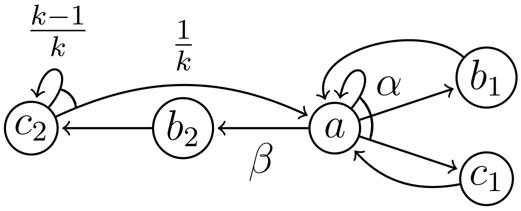


Figure 4.5: MDP $\mathcal{N}_k$ with labels indicated by the state names.

schedulers $\mathfrak{T}_n$ for $n \geq 1$, that use a counter for the number of consecutive visits to the a-state, starting with counter value 1 when entering that state via the transitions from the other states. When in the a-state, $\mathfrak{T}_n$ schedules action α if the counter value is at most n and β otherwise.

We compute $\mathbb{LP}^{\mathfrak{T}_n}_{\mathcal{N}_k}(a \,\mathrm{U}\, b)$ via the quotient representation for Markov chains shown in Section 4.2.1 using that the Markov chain $\mathcal{C}_n$ induced by $\mathfrak{T}_n$ is strongly connected and consists of the states b_1, c_1, b_2, c_2 and the states $(a,1), (a,2), \ldots, (a, n{+}1)$, where (a,i) means state a with counter value i. We pick $(a,1)$ as reference state.

Let us first compute the denominator. The expected return time from $(a,1)$ to $(a,1)$ can be written as the sum of the expected number of occurrences $\mathit{eno}(s)$ of the states s in $\mathcal{C}_n$ along the return paths from and to $(a,1)$. These values are: $\mathit{eno}(a,i) = \frac{1}{3^{i-1}}$ for $i = 1, \ldots, n{+}1$, $\mathit{eno}(b_1) = \mathit{ef}(c_1) = \left(1 - \frac{1}{3^n}\right) \cdot \frac{1}{2}$, $\mathit{eno}(b_2) = \frac{1}{3^n}$, and $\mathit{eno}(c_2) = \frac{k}{3^n}$. Note that each of the states (a,i), b_1, c_1 and b_2 occurs exactly once on each return path from $(a,1)$ to $(a,1)$. Thus, for these states s, the expected number of occurrences equals the probability of reaching s from the reference state $(a,1)$. For state c_2, we take into account that the self-loop is taken an expected $k-1$-times. We conclude:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}_n,(a,1)}(\text{“steps until } (a,1)\text{”}) \\
&= \sum_{i=1}^{n+1} \tfrac{1}{3^{i-1}} \;+\; \left(1 - \tfrac{1}{3^n}\right)\cdot\tfrac{1}{2}\cdot 2 \;+\; \tfrac{1}{3^n} \;+\; \tfrac{k}{3^n} \\
&= \tfrac{1}{4} \cdot \left(10 \;+\; (4k-2)\cdot\tfrac{1}{3^n}\right)
\end{aligned}$$

We now compute the expected accumulated ‘weight’ along the return paths from and to $(a,1)$ under scheduler $\mathfrak{T}_n$. This value can be computed as the sum of the expected number of occurrences of every state s multiplied with its probability for $a \,\mathrm{U}\, b$ in $\mathcal{C}_n$. That is:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}_n,(a,1)}(\text{“‘weight’ until } (a,1)\text{”}) \\
&= \sum_{s} \mathit{eno}(s) \cdot \Pr_{\mathcal{C}_n,s}(a \,\mathrm{U}\, b)
\end{aligned}$$

where s ranges over all states in the Markov chain $\mathcal{C}_n$ induced by $\mathfrak{T}_n$. The probability values are as follows: $\Pr_{\mathcal{C}_n,(a,i)}(a \mathbin{\mathrm{U}} b) = \frac{1}{2}\cdot\left(1 + \frac{1}{3^{n-i+1}}\right)$ for $i = 1, \ldots, n{+}1$, $\Pr_{\mathcal{C}_n,b_1}(a \mathbin{\mathrm{U}} b) = \Pr_{\mathcal{C}_n,b_2}(a \mathbin{\mathrm{U}} b) = 1$, and $\Pr_{\mathcal{C}_n,c_1}(a \mathbin{\mathrm{U}} b) = \Pr_{\mathcal{C}_n,c_2}(a \mathbin{\mathrm{U}} b) = 0$. So, we get:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{C}_n,(a,1)}(\text{“weight until } (a,1)\text{”}) \\
&= \sum_{i=1}^{n+1} \tfrac{1}{3^{i-1}}\cdot\tfrac{1}{2}\cdot\left(1 + \tfrac{1}{3^{n-i+1}}\right) \;+\; \left(1 - \tfrac{1}{3^n}\right)\cdot\tfrac{1}{2}\cdot 1 \\
&= \tfrac{1}{4}\cdot\left(5 \;+\; (2n+3)\cdot\tfrac{1}{3^n}\right), \text{ and}
\end{aligned}$$

$$\mathbb{LP}^{\mathfrak{T}_n}_{\mathcal{N}_k}(a \mathbin{\mathrm{U}} b) = \frac{5 + (2n+3)\cdot\frac{1}{3^n}}{10 + (4k-2)\cdot\frac{1}{3^n}}.$$

To determine which scheduler is optimal among the schedulers $\mathfrak{T}_n$ with $n \in \mathbb{N}$, we determine the least natural number n such that $\mathbb{LP}^{\mathfrak{T}_n}_{\mathcal{N}_k}(a \mathbin{\mathrm{U}} b) > \mathbb{LP}^{\mathfrak{T}_{n+1}}_{\mathcal{N}_k}(a \mathbin{\mathrm{U}} b)$. Treating the computed expression for $\mathbb{LP}^{\mathfrak{T}_n}_{\mathcal{N}_k}(a \mathbin{\mathrm{U}} b)$ as a real function in n, one can check that the derivative has only one root and that hence the obtained value n indeed yields the optimal scheduler among these schedulers. For $k \geq 2$, we obtain that the optimum is reached for $n = k - 1$.

We will see later that the maximal long-run probability of $\mathcal{N}_k$ is indeed achieved by $\mathfrak{T}_n$ for this n. Note that $\mathcal{N}_k$ has 5 states and its size is in $\mathcal{O}(\log k)$. So, on the one hand, the memory requirements of optimal schedulers can grow exponentially with the size of the MDP. On the other hand, the same applies to the logarithmic length of the optimal values. To see this, we observe that in

$$\mathbb{LP}^{\mathfrak{T}_{k-1}}_{\mathcal{N}_k}(a \mathbin{\mathrm{U}} b) = \frac{5 \cdot 3^{k-1} + 2k + 1}{10 \cdot 3^{k-1} + 4k - 2}$$

the greatest common divisor of enumerator and denominator is at most 4 (note that $2(5 \cdot 3^{k-1} + (2k+1)) - (10 \cdot 3^{k-1} + (4k-2)) = 4$). Therefore, the binary representation of the optimal value requires exponentially many bits in the size of $\mathcal{N}_k$. $\lhd$

There is a very simple DFA $\mathcal{D}$ that accepts words satisfying $a \mathbin{\mathrm{U}} b$: It has three states q, *accept*, and *reject*. Initially, and from q, the automaton moves to *accept* when reading a b, to *reject* when reading anything but an a or b, and it loops back to q when reading an a (and not simultaneously a b). Given an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathsf{AP}, L)$, the MDP $\mathcal{M}_\mathcal{D}$ constructed as in the previous section becomes very simple for this automaton $\mathcal{D}$: The state space of $\mathcal{M}_\mathcal{D}$ is $S \times \mathbb{N}$. Also the behavior of the counter in the second component is easily described: Whenever entering a state labeled with a (and not with b), the counter value is increased by 1. For all other states, the counter is set back to 0. When entering a state labeled with b, the current counter value plus 1 is received as weight. So, the counter simply keeps track of the number of consecutive visits to a-states.

The key result required now to be able to compute the maximal long-run probability of $a\,\mathrm{U}\,b$ in $\mathcal{M}$ is the existence of a saturation point allowing us to restrict ourselves to a finite state space by only keeping track of the counter in $\mathcal{M}_{\mathcal{D}}$ up to the saturation point. This saturation point result is similar to the saturation point result for partial expectation that we have seen in Section 3.5.

We will restrict our attention to the case that $\mathcal{M}$ is strongly connected. For general MDPs, we can maximize the long-run probability of $a\,\mathrm{U}\,b$ by maximizing the value in all maximal end components and afterwards computing the optimal value as a weighted reachability problem. In the light of Theorem 4.16, we can work with the MDP $\mathcal{M}_{\mathcal{D}}$ and show how to maximize the mean payoff in this MDP. We label states (s,n) in $\mathcal{M}_{\mathcal{D}}$ by the label of state s in $\mathcal{M}$.

Before we prove the existence of a saturation point, we fix useful notations. Given a state s of $\mathcal{M}$, let $p_s^{\max} = \mathrm{Pr}^{\max}_{\mathcal{M},s}(a\,\mathrm{U}\,b)$, and analogously $p_s^{\min} = \mathrm{Pr}^{\min}_{\mathcal{M},s}(a\,\mathrm{U}\,b)$. Further, define

$$\begin{aligned} A &= \Big\{ s \in S \mid p_s^{\max} > 0 \text{ and } p_s^{\min} < 1 \Big\}, \\ B &= \Big\{ s \in S \mid p_s^{\min} = 1 \Big\}, \quad C = S \setminus (A \cup B). \end{aligned}$$

Then, $\mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M},s}(a\,\mathrm{U}\,b) = \mathrm{Pr}^{\mathfrak{S}}_{\mathcal{M},s}(A\,\mathrm{U}\,B)$ for every s and $\mathfrak{S}$. Hence, we may safely assume that the labeling function fulfills $a \in \mathsf{L}(s)$ iff $s \in A$ and $b \in \mathsf{L}(s)$ iff $s \in B$. For $\alpha \in \mathit{Act}(s)$, let $p_{s,\alpha} = \sum_{t \in S} P(s,\alpha,t) \cdot p_t^{\max}$ and we write $\mathit{Act}^{\max}(s)$ for the set of *maximizing actions*, *i.e.* actions $\alpha \in \mathit{Act}(s)$ with $p_{s,\alpha} = p_s^{\max}$. For $T \subseteq S$, a T-EC denotes an end component consisting of T-states.

In Lemma 4.12, we have seen that the maximal mean payoff in $\mathcal{M}_{\mathcal{D}}$ can be approximated by finite-memory schedulers for $\mathcal{M}$. The existence of a saturation point will be formulated as a refinement of this statement. For a given bound K, define the class $\mathrm{FM}(K)$ of finite-memory schedulers $\mathfrak{F}$ for $\mathcal{M}_{\mathcal{D}}$ such that

1. for each finite $\mathfrak{F}$-path ϱ with $\mathit{last}(\varrho) = (s,n)$ for some $s \in S$ and $n > K$, the probability $\mathrm{Pr}^{\mathfrak{F}\uparrow\varrho}_{\mathcal{M}_{\mathcal{D}},\mathit{last}(\varrho)}(a\,\mathrm{U}\,b) = p_s^{\max}$
2. the Markov chain induced by $\mathfrak{F}$ has a single BSCC,
3. the scheduler $\mathfrak{F}$ is a finite-memory scheduler when viewed as a scheduler for $\mathcal{M}$.

By definition, any $\mathfrak{F} \in \mathrm{FM}(K)$ only schedules maximizing actions in $\mathit{Act}^{\max}(\cdot)$ for paths ending with (s,n) for some s and $n > K$ by (1); moreover, all states in $\mathcal{M}_{\mathcal{D}}$ have the same long-run probability under $\mathfrak{F}$, written $\mathbb{LP}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}}}(a\,\mathrm{U}\,b)$ by (2). We are now ready to prove the existence of a saturation point:

Lemma 4.18 (Saturation point). *There exists a natural number K computable in polynomial time that satisfies*

$$\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) = \sup_{\mathfrak{F}\in \mathrm{FM}(K)} \mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}}}(MP).$$

Proof. Recall that we relabeled states in $\mathcal{M}$ such that all states from which $a \,\mathrm{U}\, b$ holds almost surely are labeled with b and all states from which $a \,\mathrm{U}\, b$ holds with positive probability less than 1 are labeled with a. We denote by A and B the sets of states labeled with a and b, respectively, and $C = S \setminus (A \cup B)$. Let us explain how we choose and how we can compute K. Let

$$K \;=\; \max\Big\{\, |A|+1,\; \lceil (T_1+T_2)/\delta \rceil \,\Big\} \tag{*}$$

where T_1, T_2, and δ are defined in the sequel. For each $s \in S$, let

$$\delta_s \;=\; \min\Big\{\, p^{\max}_s - p_{s,\alpha} \;:\; \alpha \in Act(s) \setminus Act^{\max}(s) \,\Big\}$$

with the convention that $\min\varnothing = \infty$. The value δ is then set as $\delta = \min_{s\in A}\delta_s$ if there exists at least one A-state s with $\delta_s < \infty$ and $\delta = 1$ otherwise. Obviously, δ is computable in polynomial time in the size of $\mathcal{M}$. Intuitively, each δ_s is the minimum probability loss when the first action of an optimal scheduler for $a \,\mathrm{U}\, b$ is replaced by a non-optimal action.

Next, let $\mathfrak{Q}$ be a scheduler that maximizes the probability of $a \,\mathrm{U}\, b$ from all A-states while minimizing the expected number of steps before $B \cup C$ is reached. Such a scheduler can be found by minimizing the expected number of steps before $B \cup C$ in the MDP obtained from $\mathcal{M}$ by only allowing actions in $Act^{\max}(\cdot)$. This optimization problem is then a classical stochastic shortest path problem. The scheduler $\mathfrak{Q}$ obtaining the optimum can hence be chosen to be memoryless. We define

$$T_1 = \max_{s\in A} \mathbb{E}^{\mathfrak{Q}}_{\mathcal{M},s}(\text{“steps until } B\cup C\text{”}).$$

In order to define T_2, we consider a version of $\mathcal{M}_{\mathcal{D}}$ in which the counter only counts up to $|A|+1$. As soon as the counter value exceeds $|A|+1$, we replace the counter value by the symbol $\top$ until it is reset to 0. If the counter value is $|A|+1$ or $\top$ only actions in $Act^{\max}(\cdot)$ are enabled. Let us denote this MDP by $\mathcal{M}_{|A|+1}$. In other words, $\mathcal{M}_{|A|+1}$ is obtained from $\mathcal{M}_{\mathcal{D}}$ by collapsing all states of the form (s,k) with $k > |A|+1$ to one state $(s,\top)$ and afterwards disabling all actions not in $Act^{\max}(s)$ in the states $(s,|A|+1)$ and $(s,\top)$. (For more details, see the definition of $\mathcal{K} = \mathcal{M}_K$ after this proof.)

As a simple path from any state s to a state t in $\mathcal{M}$ contains at most $|A|$-many A-states, it is possible to reach a state of the form $(s,\cdot)$ for any s from any state in $\mathcal{M}_{|A|+1}$. For each s, let $k_s \le |A|$ be the least counter value such that (s,k_s) is reachable from any state, i.e., in particular from $B \cup C$-states. We can define the following values

$$e_s \;=\; \max_{t \in S, k \in \{0,1,\ldots,|A|,\top\}} \mathbb{E}^{\min}_{\mathcal{M}_{|A|+1},(t,k)}(\text{“steps until a state of the form } (s,k_s)\text{”}).$$

as the minimal expected number of steps from the worst possible starting state in $\mathcal{M}_{|A|+1}$ to (s, k_s). The values e_s and a corresponding MD-scheduler $\mathfrak{R}_{|A|,s}$ for $\mathcal{M}_{|A|+1}$ that minimizes the expected number of steps to a state $(s, \cdot)$ from every state are computable in polynomial-time via the classical stochastic shortest path problem. Note that the size of $\mathcal{M}_{|A|+1}$ is at most quadratic in the size of $\mathcal{M}$. We define $T_2 = \max_{s \in S} e_s$ to complete the definition of K.

By Lemma 4.12, it is now sufficient to show that for each finite-memory scheduler $\mathfrak{T}$ for $\mathcal{M}$ that is viewed as a scheduler for $\mathcal{M}_{\mathcal{D}}$ there is an FM(K)-scheduler such that

$$\mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP).$$

So, let $\mathfrak{T}$ be a finite-memory scheduler for $\mathcal{M}$ with memory modes in the finite set X. Let $\mathcal{C}^{\mathfrak{T}}$ denote the Markov chain induced by $\mathfrak{T}$. We can think of the states in $\mathcal{C}^{\mathfrak{T}}$ as pairs (s, x) consisting of a state s in $\mathcal{M}$ and a mode $x \in X$. The labels of the states of $\mathcal{C}^{\mathfrak{T}}$ depend on their first component. We may assume w.l.o.g. that $\mathcal{C}^{\mathfrak{T}}$ has a single BSCC, say $\mathcal{B}^{\mathfrak{T}}$. This yields that no matter in which state of $\mathcal{M}_{\mathcal{D}}$ and in which memory mode of $\mathfrak{T}$, we start, we obtain the same mean-payoff which we denote by $\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP)$.

If $\mathcal{B}^{\mathfrak{T}}$ consists of A-states then $\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP) = 0$ as in $\mathcal{M}_{\mathcal{D}}$ weights are only received when entering a B-state. The claim is then trivial as we can deal with any FM(K)-scheduler. If $\mathcal{B}^{\mathfrak{T}}$ consists of A- and B-states only with at least one B-state, then $\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP) = 1$ the counter in $\mathcal{M}_{\mathcal{D}}$ under $\mathfrak{T}$ will only be reset to 0 when the counter value is received as weight. In this case, any memoryless scheduler realizing this BSCC will achieve this expected mean payoff and is in FM(K) as it also maximizes the probability of $a \,\mathrm{U}\, b$ in this BSCC.

Suppose now that $\mathcal{B}^{\mathfrak{T}}$ contains at least one state in C. We now explain how to modify $\mathfrak{T}$'s decision for generating a scheduler in FM(K) with the desired property. When using $\mathfrak{T}$ as a scheduler for $\mathcal{M}_{\mathcal{D}}$, runs are equipped with the component containing the counter. So, we can extend states (s, x) of $\mathcal{B}^{\mathfrak{T}}$ by a third component to obtain (s, x, k). The third component does not influence the decision of the scheduler, but simply keeps track of consecutive visits to A-states and determines the weight that is received when entering a B-state. Our procedure works by induction on the number $\ell^{\mathfrak{T}}$ of state-action pairs with a state $\mathfrak{s} = (s, x)$ in $\mathcal{B}^{\mathfrak{T}}$ and an action $\beta \notin Act^{\max}(s)$ such that $\mathfrak{T}(s, x)(\beta) > 0$ and such that a configuration (s, x, k) in $\mathcal{M}_{\mathcal{D}}$ with $k > K$ is reachable under $\mathfrak{T}$.

If $\ell^{\mathfrak{T}} = 0$ then we can deal with $\mathfrak{S} = \mathfrak{T}$. Suppose now that $\ell^{\mathfrak{T}} \geqslant 1$. We now show how to transform $\mathfrak{T}$ into a new scheduler $\mathfrak{S}$ that is a finite-memory scheduler for $\mathcal{M}$ with a single BSCC such that $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP)$ and $\ell^{\mathfrak{S}} < \ell^{\mathfrak{T}}$.

First, fix a state (t, y) in $\mathcal{B}^{\mathfrak{T}}$ with $t \in C$. As t is in C, the counter will be reset to 0 whenever the state (t, y) is reached. So, we can express the expected mean payoff in $\mathcal{M}_{\mathcal{D}}$ under $\mathfrak{T}$ as a fraction of the expected return time from (t, y) to (t, y) and the expected weight accumulated before returning. Note that the expected weight accumulated before returning is bounded by the expected number of steps as the counter value and hence the weight that can be received increases by at most 1 per step. So, the expected accumulated weight is bounded although the weight that can be received in one step is not bounded.

The scheduler $\mathfrak{S}$ is constructed from $\mathfrak{T}$ by first adding additional finite memory always keep track of the consecutive visits to A-states up to K-many states. The scheduler $\mathfrak{S}$ will operate in three different phases. In the first phase, it operates just like $\mathfrak{T}$. If it reaches (s, x, k) for some $k > K$ – which it can detect due to the additional memory – and $\mathfrak{T}$ would choose β, it switches to the second mode instead. In the second phase, it maximizes the probability of $a \,\mathrm{U}\, b$ in a memoryless fashion by following the decisions of $\mathfrak{Q}$ until $B \cup C$ is reached. There, it switches to the third phase. It randomly chooses a target state (r, z) in $\mathcal{B}^{\mathfrak{T}}$ with probability $P((s, x), \beta, (r, z))$. This choice is implemented by a randomized update of the memory mode. So, all possible successors (r, z) that $\mathfrak{T}$ would have reached when it would have chosen β at the switch to the second phase of $\mathfrak{S}$ can randomly be chosen with precisely the probability that $\mathfrak{T}$ would have entered that successor state. In the third phase, $\mathfrak{S}$ now follows the scheduler $\mathfrak{R}_{|A|,r}$ until it reaches the chosen target state r with the least possible counter value k_r that is reachable from B- or C-states. Then, it switches back to the first phase and continues to behave like $\mathfrak{T}$ from state (r, z) on.

Let us briefly summarize the behavior of $\mathfrak{S}$: If $\mathfrak{T}$ chooses β in a state (s, x) directly after more than K consecutive A-states, $\mathfrak{S}$ instead first maximizes the probability of $a \,\mathrm{U}\, b$ before returning to a possible β successor (r, z) of (s, x) and continuing to behave like $\mathfrak{T}$.

We make a few observations: First, $\mathfrak{S}$ does not choose β in state (s, x) anymore if K or more consecutive A-states have been visited before. Further, it does not create new state-action pairs that are considered for the number $\ell^{\mathfrak{S}}$. The reason is that $\mathfrak{R}_{|A|,r}$ is constructed such that it maximizes the probability of $a \,\mathrm{U}\, b$ whenever $|A|$ or more A-states have been visited consecutively. As K is chosen to be at least $|A|$, the second and third phase of the scheduler do not induce any such state-action pairs. Further, $\mathfrak{S}$ switches back to the first phase with the least possible counter value for some state (r, z). So, also when continuing in the first phase after returning from the third phase, it does not induce any such "bad" state-action pair that was not present for $\mathfrak{T}$. So, indeed we have $\ell^{\mathfrak{S}} < \ell^{\mathfrak{T}}$.

Let us now estimate the expected mean payoff under $\mathfrak{S}$. First, let $ens^{\mathfrak{T}}$ be the expected number of steps that $\mathfrak{T}$ needs to return from (t, y) to (t, y). Let $eaw^{\mathfrak{T}}$ be the expected accumulated weight under $\mathfrak{T}$ before returning to (t, y) when starting in (t, y). So,

$$\mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP) = \frac{eaw^{\mathfrak{T}}}{ens^{\mathfrak{T}}}.$$

To compare this value to the expected mean payoff under $\mathfrak{S}$, let $eps^{\mathfrak{S}}$ be the expected number of times the scheduler $\mathfrak{S}$ switches from the first to the second phase before returning to (t, y) in the first phase when starting in (t, y). The expected return time under $\mathfrak{S}$ increases by at most $T_1 + T_2$ per switch to the second and third phase. The expected accumulated weight is affected as follows: All increases in the counter value are received as weight or reset to 0 exactly as under $\mathfrak{T}$ as long as $\mathfrak{S}$ does not switch to the second phase. If it switches to the second phase, the counter value is at least K. The probability that this counter value will be received as weight is then exactly $p^{\max}_s$ under $\mathfrak{S}$ in its second phase. Under $\mathfrak{T}$, which would choose β in (s, x) whenever $\mathfrak{S}$ switches to the second phase, this probability is $p_{s,\beta} \leq p^{\max}_s - \delta$. After $\mathfrak{S}$ switches back to the first phase in one of the β-successor states (r, z) of (s, x), all weights received due to further increases in the counter value are the same as under $\mathfrak{T}$ until the next phase switch or the return to (t, y). Therefore the expected accumulated weight from (t, y) before returning to (t, y) in the first phase under $\mathfrak{S}$ is increased by at least $K \cdot \delta \cdot eps^{\mathfrak{S}}$ compared to $\mathfrak{T}$. All in all, we obtain

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \frac{eaw^{\mathfrak{T}} + K \cdot \delta \cdot eps^{\mathfrak{S}}}{ens^{\mathfrak{T}} + (T_1 + T_2) \cdot eps^{\mathfrak{S}}}.$$

As K is chosen such that $K \cdot \delta \geq T_1 + T_2$ while $eaw^{\mathfrak{T}} / ens^{\mathfrak{T}} \leq 1$, we conclude that indeed

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{M}_{\mathcal{D}}}(MP).$$

By induction, the claim of the lemma follows. □

Using this result and Theorem 4.16, we will now be able to compute the maximal long-run frequency of constrained reachability probabilities. For this purpose, we define an MDP $\mathcal{K}$. The structure of this MDP is that of $\mathcal{M}_K$ using the notation introduced in the previous proof. Recall that $\mathcal{M}_K$ is obtained from $\mathcal{M}_{\mathcal{D}}$ by collapsing all states of the form (s, k) with $k > K$ to a state $(s, \top)$ for each s and only enabling actions in $Act^{\max}(s)$ in this state. More formally, the state space of $\mathcal{M}_K$ is $S \times \{0, \ldots, K, \top\}$. The set of actions is the same as in $\mathcal{M}$. In a state (s, k), all actions in $Act(s)$ are enabled if $k < K$. If $k = K$ or $k = \top$, then only the actions in $Act^{\max}(s)$ are enabled. The new transition probability function P' is the following for all actions α:

- For any state (s, k) of $\mathcal{M}_K$ and any state $t \in B \cup C$, we define $P'((s, k), \alpha, (t, 0)) = P(s, \alpha, t)$.

- For any state $s \in A$ and any $k \in \{1, \dots, K\}$, we define $P'((t, k-1), \alpha, (s, k)) = P(t, \alpha, s)$ for all states $t \in S$.
- For any state $s \in A$ and any $(t, k) \in S \times \{K, \top\}$, we define $P'((t, k), \alpha, (s, \top)) = P(s, \alpha, t)$ if $\alpha \in Act^{\max}(t)$.

All remaining transitions have probability 0. The initial state is $(s_{init}, 0)$ for some $s_{init} \in B \cup C$. As the reachable states are strongly connected in $\mathcal{K}$ as $\mathcal{M}$ was strongly connected, reference to the initial state is not important when we address the expected mean payoff. To complete the definition of $\mathcal{K}$, we adapt the transition-based weight function as follows:

- Transitions from states in $A \times \{k\}$ with $k \in \{1, \dots, K-1\}$ leading to a B-state have weight $k+1$.
- Transitions from a state (s, K) where $s \in A$ have weight $K \cdot p_s^{\max}$.
- Transitions from states $(s, \top)$ where $s \in A$ have weight $p_s^{\max}$.

The weight of all other transitions is 0. The MDP $\mathcal{K}$ is tailored to compute expected mean payoffs of FM(K)-schedulers for $\mathcal{M}_{\mathcal{D}}$ as shown in the following lemma.

Lemma 4.19. *With all notation as above, we have*

$$\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) = \mathbb{E}^{\max}_{\mathcal{K}}(MP).$$

Proof. The MDP $\mathcal{K}$ is constructed such that all FM(K)-schedulers can be used as schedulers for $\mathcal{K}$. As soon as K consecutive A-states have been visited, only actions in $Act^{\max}(\cdot)$ are enabled in $\mathcal{K}$. But FM(K)-schedulers also only use such actions once K consecutive A-states have been visited. Given an FM(K)-scheduler $\mathfrak{F}$ for $\mathcal{M}_{\mathcal{D}}$ that we also use as a scheduler for $\mathcal{K}$, observe that if the counter value reaches K in state s, this counter value will be received as weight with probability $p_s^{\max}$ in $\mathcal{M}_{\mathcal{D}}$. This is precisely the weight that $\mathcal{K}$ assigns to all outgoing transitions from (s, K). For each further step to an A-state t, the increased counter value will be received as weight with probability $p_t^{\max}$. Also this expected weight is accounted for in the weight function of $\mathcal{K}$ in the outgoing transitions from states $(t, \top)$. All other weights when entering a B-state with counter value below K are the same in $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{K}$. Hence,

$$\mathbb{E}^{\mathfrak{F}}_{\mathcal{M}_{\mathcal{D}}}(MP) = \mathbb{E}^{\mathfrak{F}}_{\mathcal{K}}(MP).$$

So by Lemma 4.18, we conclude that

$$\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) \leq \mathbb{E}^{\max}_{\mathcal{K}}(MP).$$

For the other direction, let $\mathfrak{T}$ be a scheduler maximizing the expected mean payoff in $\mathcal{K}$. We can assume that $\mathfrak{T}$ is memoryless and induces only one BSCC. There are two cases to consider: Either this BSCC consists only of states in $A \times \{\top\}$, or not. In the latter case, a $\mathfrak{T}$-path will leave the set of A-states infinitely often almost surely. As in $\mathcal{K}$ only actions in $Act^{\max}(\text{cot})$ are available after K or more consecutive A-states, this implies that $\mathfrak{T}$ maximizes the probability of $a \,\mathrm{U}\, b$ after K such states. Hence, $\mathfrak{T}$ can be seen as a FM(K)-scheduler for $\mathcal{M}_{\mathcal{D}}$. It follows by the considerations above that $\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{K}}(MP)$. If, however, the only BSCC of $\mathfrak{T}$ consists only of $A \times \{\top\}$-states, the scheduler does not maximize the probability for $a \,\mathrm{U}\, b$ after K consecutive A-states although it only takes actions in $Act^{\max}(\cdot)$. In fact, the probability for $a \,\mathrm{U}\, b$ is 0 in this BSCC. On the other hand, the mean payoff of $\mathfrak{T}$ in $\mathcal{K}$ is $p^{\max}_s$ for some $(s, \top)$ in the BSCC (note that all states in the BSCC have the same maximal probability for $a \,\mathrm{U}\, b$). We can construct an infinite-memory scheduler $\mathfrak{S}$ for $\mathcal{M}_{\mathcal{D}}$ that achieves this mean payoff: The scheduler acts in rounds. In the ith round, it moves to the states of the BSCC of $\mathfrak{T}$ and stays inside these A-states for 2^i-many steps. Afterwards, it maximizes the probability of $a \,\mathrm{U}\, b$ in a memoryless fashion until leaving the set of A-states. Then, it enters the next round and moves back to the BSCC of $\mathfrak{T}$. The expected time between the 2^i steps in the BSCC of $\mathfrak{T}$ and the 2^{i+1} steps in the next round is bounded. So, the frequency of steps in the BSCC of $\mathfrak{T}$ is 1 due to the increasing number of steps taken there in each round. The increases in the counter values in these steps are received as weight with probability $p^{\max}_s$ by the construction of $\mathfrak{S}$. Hence, the expected mean payoff of $\mathfrak{S}$ in $\mathcal{M}_{\mathcal{D}}$ is $p^{\max}_s$ and we conclude that $\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\mathfrak{T}}_{\mathcal{K}}(MP)$ also in this case. Therefore, we obtain

$$\mathbb{E}^{\max}_{\mathcal{M}_{\mathcal{D}}}(MP) \geq \mathbb{E}^{\max}_{\mathcal{K}}(MP). \qquad \square$$

Putting together the results shown in Theorem 4.16 and the previous Lemma 4.19, we obtain the main result of this section:

Theorem 4.20. *Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, L)$, we can compute the maximal long-run probability $\mathbb{LP}^{\max}_{\mathcal{M}}(a \,\mathrm{U}\, b)$ in time exponential in the size of $\mathcal{M}$.*

Proof. For each MEC $\mathcal{E}$ of $\mathcal{M}$, we can construct an MDP $\mathcal{K}_{\mathcal{E}}$ as described above and compute the maximal expected mean payoff in this MDP $\mathcal{K}_{\mathcal{E}}$. As the saturation point K can be computed in polynomial time, its numerical value is at most exponential in the size of $\mathcal{E}$. Hence, these maximal expected mean payoffs can be computed in exponential time. The maximal long-run frequency in $\mathcal{M}$ can then be computed by solving a weighted reachability problem in which all end components $\mathcal{E}$ are equipped with the possibility to collect the maximal expected mean payoff in $\mathcal{K}_{\mathcal{E}}$ as terminal weight. □

For the structure of optimal schedulers, we obtain the following consequence.

Corollary 4.21. *Let $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, L)$ be a strongly connected MDP without end components consisting only of a-states. Then, there is a deterministic finite-memory scheduler maximizing the long-run probability of $a \,\mathrm{U}\, b$ that uses a counter of consecutive a-states up to the saturation point K as memory.*

Proof. Let $\mathcal{K}$ be the MDP constructed from $\mathcal{M}$ as above. As there are no end components labeled with a, the memoryless deterministic scheduler $\mathfrak{S}$ that maximizes the expected mean payoff in $\mathcal{K}$ leaves the set A infinitely often almost surely. So, the expected mean payoff of this scheduler coincides with the long-run probability of $a \,\mathrm{U}\, b$. This scheduler can be seen as a deterministic finite-memory scheduler for $\mathcal{M}$ with a counter up to the saturation point K as memory. □

In fact, the statement of the corollary could be refined: If a-end components exist, infinite-memory schedulers are only necessary to maximize the long-run probability of $a \,\mathrm{U}\, b$, if all memoryless schedulers that maximize the expected mean payoff in the constructed MDP $\mathcal{K}$ induce an a-BSCC. For our purposes in the sequel, however, the non-existence of a-end components is a sufficient criterion and much easier to check.

So far, we have restricted our attention to maximal long-run probabilities. For constrained reachability properties, also minimal long-run probabilities can be treated with the same techniques as the following easy reduction shows:

Corollary 4.22. *Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, L)$, we can also compute the minimal long-run frequency $\mathbb{LP}^{\min}_{\mathcal{M}}(a \,\mathrm{U}\, b)$ in time exponential in the size of $\mathcal{M}$.*

Proof. We can again restrict ourselves to strongly connected MDPs. After defining

$$\begin{aligned} A &= \Big\{ s \in S \mid \Pr^{\max}_{\mathcal{M},s}(a \,\mathrm{U}\, b) > 0 \text{ and } \Pr^{\min}_{\mathcal{M},s}(a \,\mathrm{U}\, b) < 1 \Big\}, \\ B &= \Big\{ s \in S \mid \Pr^{\min}_{\mathcal{M},s}(a \,\mathrm{U}\, b) = 1 \Big\}, \quad C = S \setminus (A \cup B). \end{aligned}$$

as before, we observe that $\mathbb{LP}^{\min}_{\mathcal{M}}(a \,\mathrm{U}\, b) = 1 - \mathbb{LP}^{\max}_{\mathcal{M}}(A \,\mathrm{U}\, C)$ if $\mathcal{M}$ does not contain an end component of A-states because we then have $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(a \,\mathrm{U}\, b) = 1 - \Pr^{\mathfrak{S}}_{\mathcal{M},s}(A \,\mathrm{U}\, C)$ for each scheduler $\mathfrak{S}$ and each state s. If there is such an A-end component, then $\mathbb{LP}^{\min}_{\mathcal{M}}(a \,\mathrm{U}\, b) = 0$. □

In Example 4.17, we have already seen that even the binary representation of the optimal value can require exponentially many bits. Hence, this representation of the optimal value is certainly not computable in polynomial time. In the sequel, we will show that also for the threshold problem, we cannot expect a polynomial-time algorithm by establishing an NP-lower bound. In the proof, we will also encounter a problem with the length of the binary encoding of the threshold value we want to construct. By using Taylor's theorem to approximate the values of a real-valued function in the neighborhood of the intended threshold value, we will be able to choose a sufficiently good approximation of this value with a polynomial binary representation.

Theorem 4.23. *Let $\mathcal{M} = (S, Act, P, s_{init}, \mathsf{AP}, L)$ be an MDP and let $\vartheta \in \mathbb{Q}$ be given. The threshold problem "is $\mathbb{LP}^{\max}_{\mathcal{M}}(a \,\mathrm{U}\, b) \geqslant \vartheta$?" is NP-hard.*

Proof. We prove the statement by a polynomial reduction from the intersection problem for unary DFA, i.e., DFA over a one-letter alphabet. This problem is known to be NP-complete [BKM16].

So, we are given a finite number of unary DFA, say $\mathcal{D}_1, \ldots, \mathcal{D}_k$ over the alphabet $\Sigma = \{0\}$. where $\mathcal{D}_i = (Q_i, \Sigma, \delta_i, q_{0.i}, F_i)$. We simply write $\delta_i(q)$ rather than $\delta_i(q, 0)$. We may suppose the transition functions δ_i are total and that $Q_i \cap Q_j = \varnothing$ if $i \neq j$. W.l.o.g. we further assume that $|Q_i| \geq 2$ for all $i \leq k$.

We are going to construct an MDP $\mathcal{M}$ over $\mathsf{AP} = \{a, b\}$ and a rational value ϑ such that $\mathbb{LP}^{\max}_{\mathcal{M}}(a \,\mathrm{U}\, b) \geqslant \vartheta$ if and only if $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ is nonempty. The latter is equivalent to the statement that there exists some $n \in \mathbb{N}$ with $n < \ell$ and $0^n \in \mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ where $\ell = |Q_1| \cdot \ldots \cdot |Q_k|$. Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ denote the NFA resulting from the union of $\mathcal{D}_1, \ldots, \mathcal{D}_k$. That is, $Q = Q_1 \cup \ldots \cup Q_k$, $Q_0 = \{q_{0,1}, \ldots, q_{0,k}\}$, $F = F_1 \cup \ldots \cup F_k$ and $\delta(q) = \delta_i(q)$ if $q \in Q_i$. That is, besides the nondeterministic choice of the initial state, $\mathcal{A}$ behaves deterministically.

The automaton $\mathcal{A}$ will be incorporated into the MDP $\mathcal{M}$ we are going to construct now. The MDP $\mathcal{M}$ is also depicted in Figure 4.6. The state space of $\mathcal{M}$ is

$$S \;=\; Q \cup \{a, b, c, s_{init}\}$$

where the states in $Q \cup \{a\}$ are labeled by a and b is labeled by b. The action set is $Act = \{\alpha, \beta, enter, pump, \tau\}$. We define $r = \ell + 3/2$. The transition probabilities are as follows:

- In $s \in \{s_{init}, a\}$, actions *enter* and *pump* are enabled with the transition probabilities:
$$\begin{aligned} &P(s, enter, q_{0,i}) = \tfrac{r-1}{k \cdot r},\ i = 1, \ldots, k,\\ &P(s, enter, s_{init}) = \tfrac{1}{r},\\ &P(s, pump, a) = \tfrac{r-1}{r},\ P(s, pump, s_{init}) = \tfrac{1}{r}. \end{aligned}$$

- In each state $q \in Q$, action α is enabled with:
$$P(q, \alpha, \delta(q)) = \tfrac{r-1}{r},\ P(q, \alpha, s_{init}) = \tfrac{1}{r}$$

  For the final states $q \in F$, additionally action β is enabled with $P(q, \beta, b) = 1$.

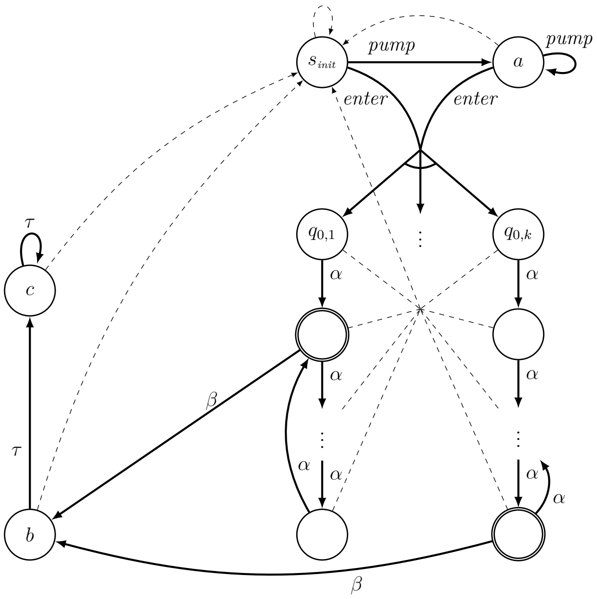


Figure 4.6: The MDP $\mathcal{M}$ for the proof of Theorem 4.23. Each state-action pair leads to the initial state s_{init} with probability $1/r$ as indicated by the dashed transitions. The remaining probability mass of $(r-1)/r$ is uniformly distributed over the remaining successors.

- In $s' \in \{b, c\}$, action τ is enabled with:

$$P(s', \tau, c) = \tfrac{r-1}{r}, \quad P(s', \tau, s_{init}) = \tfrac{1}{r}$$

The idea now is to find a threshold ϑ such that the scheduler maximizing the long-run probability can exceed the threshold if and only if the intersection $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ is non-empty. Note that the expected return time from s_{init} to s_{init} is r under any scheduler because the probability to reach s_{init} is $1/r$ in each step. By Corollary 4.21, we further know that there is a deterministic finite-memory scheduler maximizing the long-run probability of $a \,\mathrm{U}\, b$ in $\mathcal{M}$. For finite-memory schedulers, we can express the long-run probability as fraction of the expected return time and the weight accumulated according to the weight function *wgt* of an MDP $\mathcal{N}$ constructed in the previous section as shown in Lemma 4.15. Recall that for the constrained reachability property $a \,\mathrm{U}\, b$ this weight function assigns weight $k+1$ to transitions entering a state labeled b after k consecutive states labeled a and no weights to other transitions. To maximize the long-run probability hence means to maximize the expected weight before returning to s_{init}.

As we can restrict ourselves to deterministic schedulers, we can further simplify the analysis: A deterministic scheduler $\mathfrak{S}$ will choose the action *pump* for some number $n \geq 0$ of steps before choosing *enter*. After randomly entering the initial state $q_{0,i}$ among the initial states of $\mathcal{A}$, it chooses α for some number $m_i \geq 0$ of steps and afterwards β. This is possible if and only if $0^{m_i} \in \mathcal{L}(\mathcal{D}_i)$. Clearly, never choosing β is suboptimal and so we assume w.l.o.g. that $\mathfrak{S}$ chooses β at some point.

Define $\rho = \frac{r-1}{r}$. Using the expected accumulated weight before returning to s_{init} under $\mathfrak{S}$, we can express the long-run probability by

$$\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M}}(a \,\mathrm{U}\, b) = \sum_{i=0}^{k} \frac{\rho^{n+1} \cdot 1/k \cdot \rho^{m_i+1} \cdot (n + m_i + 2)}{r} = \sum_{i=0}^{k} \frac{\rho^{n+m_i+2} \cdot (n + m_i + 2)}{k \cdot r}.$$

The probability that the n times the scheduler tries to choose *pump* and the choice of *enter* do not lead back to s_{init} is ρ^{n+1}. If this does not happen, the probability to enter $q_{0,i}$ is $1/k$. Choosing m_i times α and β afterwards then leads to b with probability ρ^{m_i+1}. If b is reached, the state a has been visited n times and $m_i + 1$ states of $\mathcal{A}$ that are all labeled with a have been visited. So, the weight received in this case is $n + m_i + 2$. As discussed before, the expected return time is r.

Let us denote $n + m_i + 2$ by N_i. The long-run probability of $\mathfrak{S}$ is determined by the expressions $\rho^{N_i} \cdot N_i$. Consider the real-valued function $f(x) = \rho^x \cdot x$ for $x \geq 0$. Its derivative is

$$\frac{\partial f(x)}{\partial x} = \rho^x \cdot (x \cdot \log(\rho) + 1).$$

As $\log(\rho)$ is negative, the derivative has one zero and as the sign of the derivative switches from $+$ to $-$, the function $f(x)$ is strictly increasing until the maximum is reached and the function strictly decreases afterwards. To find the maximum among natural numbers, it is hence sufficient to consider the differences $f(N) - f(N-1)$ for natural numbers N. Using $\rho = (r-1)/r$, we have

$$f(N) - f(N-1) = \rho^{N-1} \cdot (\rho \cdot N - (N-1)) = \rho^{N-1} \cdot (1 - N/r). \qquad (*)$$

Clearly, this value is positive for $N < r$ and negative for $N > r$. As $r = \ell + 3/2$, the maximum of $f(N)$ is obtained for $N = \ell + 1$. We define

$$\mu \overset{\mathrm{def}}{=} \min_{n \in \mathbb{N}, n \neq \ell+1} f(\ell+1) - f(n) = \min\{f(\ell+1) - f(\ell), f(\ell+1) - f(\ell+2)\}.$$

It is now not hard to check that

$$\mathbb{LP}^{\mathfrak{S}}_{\mathcal{M}}(a \,\mathrm{U}\, b) = \sum_{i=0}^{k} \frac{f(N_i)}{k \cdot r} \geq \frac{f(\ell+1) - \mu/(2k)}{r}$$

if and only if $N_i = \ell + 1$ for all i: If one of the values $f(N_i) < f(\ell+1)$, it is at most $f(\ell+1) - \mu$. In this case, the sum $\sum_{i=1}^{k} f(N_i)$ is at most $k \cdot f(\ell+1) - \mu$.

If $N_i = \ell + 1$ for all i, then all m_i are equal to some $m \leq \ell - 1$ and hence $0^m \in \mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$. If there is no choice of n and m_i for all i such that $N_i = \ell + 1$ for all i, then there is no $m < \ell$ with $0^m \in \mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$. As stated above, this implies that $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ is empty. If we define

$$\vartheta' = \frac{f(\ell+1) - \mu/(2k)}{r},$$

we hence have that $\mathbb{LP}^{\max}_{\mathcal{M}}(a \,\mathrm{U}\, b) \geq \vartheta'$ if and only if $\mathcal{L}(\mathcal{D}_1) \cap \ldots \cap \mathcal{L}(\mathcal{D}_k)$ is non-empty. Unfortunately, we cannot compute the threshold ϑ' in polynomial time and its binary representation might be of exponential length. Nevertheless, any threshold ϑ with $\vartheta' \leq \vartheta \leq f(\ell+1)/r$ still works. The goal now is hence to find such a threshold ϑ that can be computed in polynomial time.

The following inequality is well-known:

$$\rho^\ell = \left(1 - \frac{1}{r}\right)^\ell \geq 1 - \frac{\ell}{r} = 1 - \frac{r - 3/2}{r} = \frac{3}{2r}.$$

Analogously, $\rho^{\ell+1} \geq \frac{1}{2r}$. To get an estimate for μ, we obtain using $(*)$ that

$$f(\ell+1) - f(\ell) = \rho^\ell(1 - \frac{r - 1/2}{r}) \geq \frac{3}{4r^2}.$$

and

$$f(\ell+2) - f(\ell+1) = \rho^{\ell+1}(1 - \frac{r + 1/2}{r}) \leq -\frac{1}{4r^2}.$$

Taking the sign into account, we obtain that $\mu \geq \frac{1}{4r^2}$. In addition, $k \leq r$ as we assume that all $\mathcal{D}_i$ have at least two states. So, it would be sufficient, if we find an approximation of $f(\ell+1)$ up to an absolute error of $\frac{1}{16r^3}$. If this approximation is η, we can choose

$$\vartheta = \frac{\eta - 1/16r^3}{r}$$

to ensure that $\vartheta' \leq \vartheta \leq f(\ell+1)/r$. As r occurs in the denominator, we can approximate $f(\ell+1)/r$ up to an absolute error of $1/16r^2$. Plugging in $\ell = r - 3/2$ and substituting $z = 1/r$, we need an approximation of

$$g(z) \overset{\text{def}}{=} z f(1/z - 1/2) = z \left(\frac{1/z - 1}{1/z}\right)^{1/z - 1/2} (1/z - 1/2)$$

up to an absolute error of $z^2/16$ at $z = 1/r$.

We will use Taylor's theorem to obtain this approximation. We consider $g(z)$ as a real-valued function for $z \in [-1/2, 1/2] \setminus \{0\}$ and define $g(0) \overset{\text{def}}{=} \frac{1}{e}$ where e is Euler's number. Using standard methods from calculus, we see that the function f is at least three times continuously differentiable on $[-1/2, 1/2]$. Calculating the derivatives, we obtain the following Taylor approximation around $z = 0$:

$$g(z) = \frac{1}{e} - \frac{z}{2e} - \frac{z^2}{12e} + \mathcal{O}(z^3)$$

for $z \to 0$. So there are reals c_0, $z_0 > 0$ such that

$$\left| g(z) - \left(\frac{1}{e} - \frac{z}{2e} - \frac{z^2}{12e} \right) \right| \leq c_0 z^3$$

for $|z| < z_0$. If $|z| < 1/32c_0$, the expression

$$\frac{1}{e} - \frac{z}{2e} - \frac{z^2}{12e} = \frac{1}{e}(1 - z/2 - z^2/12)$$

approximates $g(z)$ up to an absolute error of $z^2/32$. As $(1 - z/2 - z^2/12) < 1$, for $z > 0$, it is now sufficient to approximate $1/e$ up to an absolute error of $z^2/32$ to obtain an approximation of $g(z)$ up to an absolute error of $z^2/16$. This is doable in polynomial time.

To conclude, let $L = \max\{1/z_0, 32c_0\}$. For $r > L$, the above procedure works as $z = 1/r$ then satisfies $|z| < z_0$ and $|z| < 1/32c_0$. So, we can compute a threshold ϑ in polynomial time for such r which completes the reduction of the intersection problem for unary DFA to the threshold problem for maximal long-run probabilities in MDPs. As L is defined in terms of the function g, i.e. independent of all variables, and as there are only finitely many instances with $r = \ell + 3/2 \leq L$, this finishes the proof. □

While we have shown the general quantitative threshold problem for the long-run probability of $a \,\mathrm{U}\, b$ to be NP-hard, the qualitative variants are efficiently solvable:

Remark 4.24 (Qualitative threshold problems)**.** In a strongly connected MDP $\mathcal{M}$, deciding whether $\mathbb{LP}^{\max}_{\mathcal{M},s}(a \,\mathrm{U}\, b)$ is 1 and deciding whether it is positive are easy: The value is positive iff there is a b-state. Further, $\mathbb{LP}^{\max}_{\mathcal{M},s}(a \,\mathrm{U}\, b) = 1$ if and only if one of the following two conditions is met:

- There is an end-component consisting only of states labeled with a or b that contains at least one b state, or
- there is an a-end component such that we have $\mathrm{Pr}^{\max}_{\mathcal{M},s}(a \,\mathrm{U}\, b) = 1$ for all (any) states s of the end component.

As we have seen, infinite-memory schedulers are necessary for the maximization of the long-run probability if only the second condition is satisfied.

For the minimal value $\mathbb{LP}^{\min}_{\mathcal{M},s}(a \,\mathrm{U}\, b)$, the situation is similarly simple: The value is less than 1 iff there either is a state not labeled with a or b, or there is an end component containing no b-state. The value is 0 iff there is an end component containing no b-state. $\lhd$

4.3 Long-run expectations

In this section, we are looking at a notion of long-run expectation that is defined analogously to long-run probabilities. It captures the long-run average of the weight that is expected to be accumulated before the next visit to a goal state. The goal of this notion is the possibility to provide guarantees on the quantitative aspects of a system modeled by weights in long-run equilibrium. The following example demonstrates that the notion might be useful to provide guarantees on the system behavior that capture the intended behavior more precisely than, for example, expected return times from a goal state to a goal state.

Example 4.25. Consider the MDP $\mathcal{M}$ depicted in Figure 4.7. Suppose the quantity we are interested in is the expected time, i.e., number of steps, until reaching *goal*. So, we assign weight +1 to all state-action pairs. Suppose further that the MDP models a system component that is working on some task and sometimes receives updated information from other components. The modeled component, however, only checks whether it received updated information when it is in state *goal*. We are now interested in the expected time that passes between the update by another component at an arbitrary moment after the system has been running for a long time and the moment the component checks for the update.

There are two memoryless schedulers $\mathfrak{S}_\alpha$ and $\mathfrak{S}_\beta$ always choosing α and β, respectively. First, note that under $\mathfrak{S}_\beta$ it takes 5 steps to return from *goal* to *goal*. Under $\mathfrak{S}_\alpha$ the number of steps for this return is 3 with probability $2/3$ and it is 7 with probability $1/3$. This yields an expected return time of $13/3 < 5$.

Nevertheless, we are not interested in the time that passes between two checks for an update but the time that passes until the next check from a random moment in time. Somewhat counter-intuitively, this expected time is lower under $\mathfrak{S}_\beta$ than under $\mathfrak{S}_\alpha$: Let us first consider scheduler $\mathfrak{S}_\beta$. At a random moment of time, the process is in any of the states s, b_3, b_2, b_1, and *goal* with the same probability. The expected times until the next visit to *goal* are 4, 3, 2, 1, and 0, respectively. The average expected time is hence

$$\frac{4+3+2+1}{5} = 2.$$

We call this value the long-run expectation (of the accumulated weight before reaching a goal state). For scheduler $\mathfrak{S}_\alpha$, the situation is slightly more complicated. States s, t,

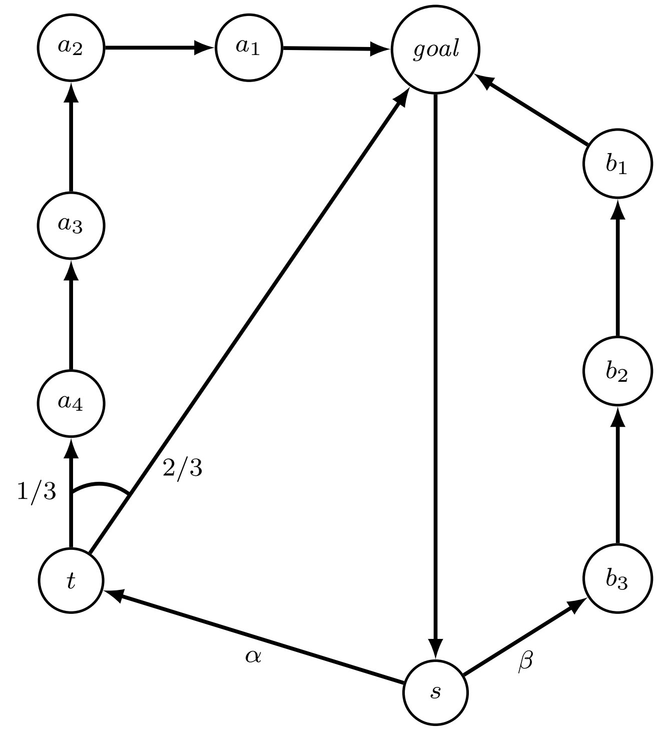


Figure 4.7: Illustration of long-run expectations.

and *goal* are visited on each path from *goal* to *goal* while $a_4, \dots, a_1$ are only visited with probability 1/3 on such a path. So, the fraction of time in which the process is in states s, t, and *goal* is 3/13 for each of these states. For states $a_4, \dots, a_1$, this fraction is only 1/13 each. The expected time until reaching *goal* on the other hand, is 10/3 from state s, 7/3 from state t, and i from state a_i for all i. For the long-run expectation, we hence obtain the value

$$\frac{3}{13} \cdot (10/3 + 7/3 + 0) + \frac{1}{13} \cdot (4 + 3 + 2 + 1) = \frac{27}{13} > 2.$$

In conclusion, that means that although the process checks for updates less frequently under $\mathfrak{S}_\beta$ (once every 5 steps) than under $\mathfrak{S}_\alpha$ (once every 13/3 steps in average), the expected time until information updated at a random moment in the long-run is recognized by the modeled component is less under $\mathfrak{S}_\beta$ (2 time steps) than under $\mathfrak{S}_\alpha$ (27/13 time steps). ◁

Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$ be a weighted MDP with integer weights and a designated set of states *Goal*. As we are interested in the long-run behavior of the system now, we do not want the states in *Goal* to be absorbing. Therefore, we slightly change the definition of the random variable $\oplus \mathit{Goal}$ expressing the accumulated weight before reaching *Goal* that we also used for stochastic shortest path problems. On infinite paths

ζ, we define $\diamondplus \mathit{Goal}(\zeta) = \mathit{wgt}(\pi)$ where π is the shortest prefix of ζ with $\mathit{last}(\pi) \in \mathit{Goal}$. If $\zeta \not\models \Diamond \mathit{Goal}$, then $\diamondplus \mathit{Goal}(\zeta)$ is undefined.

The *long-run expectation* under scheduler $\mathfrak{S}$ along ζ is now defined as

$$\mathit{lre}^{\mathfrak{S}}(\zeta) = \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus \mathit{Goal}).$$

The *long-run expectation* of a scheduler $\mathfrak{S}$ is

$$\mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} = \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\mathit{lre}^{\mathfrak{S}}).$$

Note that the long-run expectation of a scheduler $\mathfrak{S}$ is defined if the probability to reach *Goal* infinitely often $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\Box\Diamond \mathit{Goal})$ is 1. The optimal long-run expectation is

$$\mathrm{LRE}^{\max}_{\mathcal{M}} = \sup_{\mathfrak{S}} \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}}$$

where the supremum ranges over all schedulers $\mathfrak{S}$ with $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\Box\Diamond \mathit{Goal}) = 1$. The goal of this section is to compute the maximal long-run expectation.

As for expected mean payoffs and long-run probabilities, the challenging part when maximizing the long-run expectation lies in the maximization inside the MECs. If we know the optimal values in each MEC, computing the optimal value of the whole MDP can easily be done by solving weighted reachability problem in polynomial time. Hence, we consider only strongly connected MDPs in the sequel. Furthermore, we assume that the maximal expectation of $\diamondplus \mathit{Goal}$ is finite from each state. This is equivalent to the non-existence of positively weight-divergent end components without a *Goal*-state. In the presence of 0-end components and of end-components with negative maximal expected mean-payoff that do not contain a *Goal*-state, there are schedulers that do not reach *Goal* infinitely often almost surely. The treatment of MDPs with such end components will not be discussed here and is left as future work.

Here, we simplify the analysis by assuming that all end-components contain a *Goal*-state. In such an MDP $\mathcal{M}$, all schedulers $\mathfrak{S}$ satisfy $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\Box\Diamond \mathit{Goal}) = 1$. Further, there is a simple bound U computable in polynomial time such that $|\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(\diamondplus \mathit{Goal})| \leq U$ for all schedulers $\mathfrak{S}$ and all states s: If $|S|$ is the number of states and q the minimal transition probability, *Goal* is reached within $|S|$ steps with probability at least $q^{|S|}$ under each scheduler. Hence, the expected number of steps until *Goal* is at most $|S|/(q^{|S|})$. If W is the maximal absolute value of weights occurring in $\mathcal{M}$, then $U = W \cdot |S|/(q^{|S|})$ serves as such a bound.

Before we show how to maximize long-run expectations, we begin with a hardness-proof showing that we cannot expect a polynomial-time algorithm for the computation of optimal long-run expectations:

Proposition 4.26. *Given an MDP $\mathcal{M}$ and a rational ϑ, deciding whether*

$$\mathrm{LRE}^{\max}_{\mathcal{M}} \geq \vartheta$$

is NP-hard.

Proof. There is an easy reduction from the threshold problem for long-run probabilities of constrained reachability properties which we have just shown to be NP-hard. Given an MDP $\mathcal{M}$ with labels a, b, and c we construct an MDP $\mathcal{N}$ as follows. All states labeled b or c are now *Goal*-states. Furthermore, all in-going edges to b-states get weight $+1$. All other weights are 0. Now, the maximal long-run probability of $a \,\mathrm{U}\, b$ in $\mathcal{M}$ is equal to the maximal long-run expectation in $\mathcal{N}$. □

Employing Fatou's lemma as we have seen before, we can prove that the maximal long-run expectation can be approximated by finite-memory schedulers.

Lemma 4.27. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be a strongly connected MDP with a designated set of states Goal that intersects all end components. Then,*

$$\mathrm{LRE}^{\max}_{\mathcal{M}} = \sup_{\mathfrak{F}} \mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}}$$

where $\mathfrak{F}$ ranges over all finite-memory schedulers for $\mathcal{M}$.

Proof. Let $\varepsilon > 0$. By definition of $\mathrm{LRE}^{\max}_{\mathcal{M}}$, there is a scheduler $\mathfrak{S}$ with $\mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} > \mathrm{LRE}^{\max}_{\mathcal{M}} - \varepsilon/3$. By Fatou's Lemma, we get that

$$\begin{aligned}\mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} &= \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}\left(\liminf_{n\to\infty} \tfrac{1}{n+1} \textstyle\sum_{i=0}^{n} \mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus Goal)\right) \\ &\leq \liminf_{n\to\infty} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}\left(\tfrac{1}{n+1} \textstyle\sum_{i=0}^{n} \mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus Goal)\right).\end{aligned}$$

So, there is a natural number N such that, for all $n \geq N$,

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}\left(\frac{1}{n+1} \sum_{i=0}^{n} \mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus Goal)\right) \geq \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} - \varepsilon/3 > \mathrm{LRE}^{\max}_{\mathcal{M}} - 2\varepsilon/3.$$

The idea now is to obtain a finite memory scheduler $\mathfrak{F}$ form $\mathfrak{S}$ by following $\mathfrak{S}$ for a large number of steps k. In order to ensure that the accumulated expectation in the first k steps under $\mathfrak{F}$ is no less than the accumulated expectation under $\mathfrak{S}$, the scheduler $\mathfrak{F}$ switches to a new mode after k steps in which it maximizes the expectation of $\diamondplus Goal$ in a memoryless fashion. Once *Goal* is reached, $\mathfrak{F}$ returns to the initial state and starts from the beginning again.

More formally, let $\mathfrak{U}$ be a memoryless scheduler maximizing $\mathbb{E}^{\mathfrak{U}}_{\mathcal{M},s}(\diamondplus Goal)$ from all states s while minimizing the worst-case expected number of steps until *Goal*, $T_{\mathfrak{U}} = \max_t \mathbb{E}^{\mathfrak{U}}_{\mathcal{M},t}(\text{steps until } Goal)$ among all such schedulers.

Further, let $\mathfrak{R}$ be a scheduler which minimizes the expected number of steps to s_{init} from each state t while maximizing the expected value of $\diamondplus Goal$ whenever $|S|$ consecutive states not in *Goal* have been visited where S is the set of states of $\mathcal{N}$. Let $T_{\mathfrak{R}}$ be $\max_{t\in S} \mathbb{E}^{\mathfrak{R}}_{\mathcal{M},t}(\text{steps until } s_{init})$. This value and the scheduler $\mathfrak{R}$ are computable in polynomial time: We can encode the number of consecutive states not in *Goal* up to $|S|$ into the state space. If $|S|$ or more consecutive such states have been visited, only actions in $Act^{\max}$ are enabled. In the resulting MDP $\mathcal{N}$ of polynomial size, the schedulers $\mathfrak{R}$ simply minimizes the expected number of steps until a state corresponding to s_{init} is reached and hence can be computed in polynomial time. As we remove actions only after $|S|$ many steps, we do not restrict copies of which states s are reachable from any state compared to $\mathcal{M}$. In $\mathcal{N}$ the scheduler $\mathfrak{R}$ can be chosen to be memoryless. Regarded as schedulers for $\mathcal{M}$, the scheduler $\mathfrak{R}$ has $|S|$-many memory modes.

Now, let $\mathfrak{F}_k$ be the scheduler which follows $\mathfrak{S}$ for $k > N$ steps, then behaves like $\mathfrak{U}$ until *Goal* is reached and afterwards like $\mathfrak{R}$ until it starts again in the initial state. We define $T = T_{\mathfrak{U}} + T_{\mathfrak{R}}$. The expected accumulated expectation $eae^{\mathfrak{F}}$ under this scheduler until it starts again is at least

$$\begin{aligned} & \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}\left(\tfrac{1}{k}\textstyle\sum_{i=0}^{k-1}\mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus Goal)\right) - T\cdot U \\ \geq \quad & k(\mathrm{LRE}^{\max}_{\mathcal{M}} - 2\varepsilon/3) - T\cdot U \end{aligned}$$

where $U = W\cdot|S|/(q^{|S|})$ is the bound on the absolute value of the expected accumulated weight before reaching *Goal* described above with $|S|$ the number of states, q the minimal transition probability, and W the maximal absolute value of the weights occurring in $\mathcal{M}$. The expected number of steps is at most $k+T$ and at least k. We claim that for

$$k \geq 6T\cdot\frac{U}{\varepsilon},$$

$\mathrm{LRE}^{\mathfrak{F}_k}_{\mathcal{M}} \geq \mathrm{LRE}^{\max}_{\mathcal{M}} - \varepsilon$. This follows by a straight-forward computation using that $\mathrm{LRE}^{\max}_{\mathcal{M}} \leq U$ and that

$$\mathrm{LRE}^{\mathfrak{F}_k}_{\mathcal{M}} \geq \min\left\{\frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}}{-}2\varepsilon/3){-}T\cdot U}{k}, \frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}}{-}2\varepsilon/3){-}T\cdot U}{k{+}T}\right\}.$$

If the minimum is $\frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}}-2\varepsilon/3)-T\cdot U}{k}$, then

$$\mathrm{LRE}^{\mathfrak{F}_k}_{\mathcal{M}} \quad\geq\quad \frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}}{-}2\varepsilon/3){-}T\cdot U}{k} \geq \mathrm{LRE}^{\max}_{\mathcal{M}} - \varepsilon$$

follows from the fact that $k(\mathrm{LRE}^{\max}_{\mathcal{M}}{-}2\varepsilon/3){-}T\cdot U \geq k\cdot\mathrm{LRE}^{\max}_{\mathcal{M}} - \varepsilon$ because $k\cdot\varepsilon/3 \geq T\cdot U$ by the choice of k.

If the minimum is $\frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}} - 2\varepsilon/3) - T\cdot U}{k+T}$, then

$$\mathrm{LRE}^{\mathfrak{F}_k}_{\mathcal{M}} \ \geq \ \frac{k(\mathrm{LRE}^{\max}_{\mathcal{M}} - 2\varepsilon/3) - T\cdot U}{k+T} \geq \mathrm{LRE}^{\max}_{\mathcal{M}} - \varepsilon$$

follows from the fact that $k(\mathrm{LRE}^{\max}_{\mathcal{M}} - 2\varepsilon/3) - T\cdot U \ \geq \ (k+T)\cdot \mathrm{LRE}^{\max}_{\mathcal{M}} - (k+T)\cdot\varepsilon$ which holds because $k\cdot\varepsilon/3 \geq T\cdot U + T\cdot \mathrm{LRE}^{\max}_{\mathcal{M}} - T\cdot\varepsilon$ by the choice of k and the fact that $\mathrm{LRE}^{\max}_{\mathcal{M}} \leq U$. So, the scheduler $\mathfrak{F}_k$ for $k = \max\{N+1, 6T\cdot\frac{U}{\varepsilon}\}$ obtains a long-run expectation at most ε worse than the optimum. This proves that $\mathrm{LRE}^{\max}_{\mathcal{M}} = \sup_{\mathfrak{F}} \mathbb{E}^{\mathfrak{F}}_{\mathcal{M},s_{init}}(lre^{\mathfrak{F}})$. □

Note that we have made sure that the scheduler $\mathfrak{F}_k$ in the proof maximizes the expected value of $\diamondplus Goal$ whenever k consecutive states not in $Goal$ have been visited as it is in the phase where it behaves as $\mathfrak{U}$ or $\mathfrak{R}$ whenever that happens. So, we can further restrict the supremum to range over all finite-memory schedulers $\mathfrak{F}$ for which there is a natural number $N_{\mathfrak{F}}$ such that $\mathfrak{F}$ (more precisely, the appropriate residual scheduler) maximizes the expected value of $\diamondplus Goal$ whenever $N_{\mathfrak{F}}$ consecutive states not in $Goal$ have been visited. Note that the bound depends on the scheduler. In the sequel, we provide a saturation point which is such a bound that only depends on the MDP.

Saturation point. The key insight for the computation of maximal long-run expectations is now that there is again a saturation point, in this case a bound on the number of consecutive visits to states not in $Goal$ after which an optimal scheduler can behave memorylessly.

Lemma 4.28. *Let $\mathcal{M}$ be a strongly connected MDP with a designated set of states Goal that intersects all end components. There is a natural number K computable in polynomial time such that for any $\epsilon > 0$ there is a finite-memory scheduler $\mathfrak{T}$ that maximizes the expected value of $\diamondplus Goal$ whenever K consecutive steps not visiting Goal have been made such that*

$$\mathrm{LRE}^{\mathfrak{T}}_{\mathcal{M},s_{init}} \geq \mathrm{LRE}^{\max}_{\mathcal{M},s_{init}} - \epsilon.$$

Proof. We recall that $ENS^{ub} = |S|/q^{|S|}$ can be used as an upper bound for the expected number of steps until the next visit to $Goal$ from any starting state under any scheduler where S is the set of states and q the minimal transition probability. So, $E^{ub} = W\cdot ENS^{ub}$, where W is the maximal absolute value of weights occurring in $\mathcal{M}$, is an upper bound for the expected value of $\diamondplus Goal$ form any state under any scheduler. Finally, for all states s and all schedulers $\mathfrak{Q}$, the expected accumulated expectation

$$\sum_{\zeta \text{ path from } s \text{ to } Goal} \mathrm{Pr}^{\mathfrak{Q}}_{\mathcal{M},s}(\zeta) \sum_{i=0}^{length(\zeta)} \mathbb{E}^{\mathfrak{Q}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\diamondplus Goal)$$

before reaching $Goal$ is bounded by $EAE^{ub} = ENS^{ub}\cdot E^{ub}$.

We show how to compute K. First, we define $E^{\max}_s = \mathbb{E}^{\max}_{\mathcal{M},s}(\diamondplus \mathit{Goal})$ for $s \in S$ and $E^{\max}_{s,\alpha} = \mathit{wgt}(s,\alpha) + \sum_{t\in S} \Pr(s,\alpha,t)\mathbb{E}^{\max}_{\mathcal{M},t}(\diamondplus \mathit{Goal})$ for $s \in S \setminus \mathit{Goal}$ and $\alpha \in \mathit{Act}(s)$. Further, $\mathit{Act}^{\max}(s)$ denotes the set of all actions with $E^{\max}_s = E^{\max}_{s,\alpha}$. We now define

$$\delta = \min_{s\in S\setminus \mathit{Goal}, \alpha\notin \mathit{Act}^{max}(s)} E^{\max}_s - E^{\max}_{s,\alpha}.$$

If the minimum ranges over the empty set, any scheduler maximizes the expectation of $\diamondplus \mathit{Goal}$ and the claim of the theorem holds trivially.

As above, let $\mathfrak{U}$ be a memoryless scheduler maximizing $\mathbb{E}^{\mathfrak{U}}_{\mathcal{M},s}(\diamondplus \mathit{Goal})$ from all states s while minimizing the worst-case expected number of steps until Goal,

$$T_{\mathfrak{U}} = \max_t \mathbb{E}^{\mathfrak{U}}_{\mathcal{M},t}(\text{steps until } \mathit{Goal}),$$

among all such schedulers. Further, for each state s, let $\mathfrak{R}_s$ be a scheduler which minimizes the expected number of steps to s from each state t while maximizing the expected value of $\diamondplus \mathit{Goal}$ whenever $|S|$ consecutive states not in Goal have been visited. Let $T_{\mathfrak{R}}$ be $\max_{s,t\in S} \mathbb{E}^{\mathfrak{R}_s}_{\mathcal{M},t}(\text{steps until } s)$. In the previous proof, we have seen that these schedulers can be computed in polynomial time and can be chosen to be finite-memory schedulers.

Finally, let $T = T_{\mathfrak{U}} + T_{\mathfrak{R}}$. We claim that

$$K = \max\left\{ \frac{T \cdot E^{ub} + EAE^{ub} + E^{ub} \cdot ENS^{ub}}{\delta}, \frac{EAE^{ub} + 2E^{ub} \cdot T}{\delta}, |S| + 1 \right\}$$

satisfies the desired property.

To show this, let $\mathfrak{S}$ be a finite memory scheduler that is ϵ-optimal and for which there is a natural number $N_{\mathfrak{S}}$ such that $\mathfrak{S}$ maximizes the expected value of $\diamondplus \mathit{Goal}$ whenever $N_{\mathfrak{S}}$ consecutive states not in Goal are visited. Such a scheduler exists by Lemma 4.27 and the discussion after that lemma. W.l.o.g. we can assume that the Markov chain induced by $\mathfrak{S}$ has one BSCC $\mathcal{B}$. The states of $\mathcal{B}$ are pairs (s,x) of states of $\mathcal{M}$ and memory modes of $\mathfrak{S}$. We still write Goal to denote the set of states in $\mathcal{B}$ whose first component is in Goal. Let π be a path of length at least K in $\mathcal{B}$ starting in $(s_0,x_0) \in \mathit{Goal}$ and not visiting Goal again afterwards with $\mathit{last}(\pi) = (s_1,x_1)$. Assume that $\mathfrak{S}$ chooses an action $\alpha \notin \mathit{Act}^{\max}(s_1)$ in memory mode x_1 with positive probability. Let p be the probability that $\mathfrak{S}$ produces π and chooses α afterwards when starting in (s_0,x_0). Further, let $p_{(t,y)}$ be the probability that the first state in Goal that is reached under $\mathfrak{S}$ when starting in (s_1,x_1) while choosing α is (t,y).

We construct a scheduler $\mathfrak{F}$ which does not choose non-maximizing actions after π anymore: This scheduler $\mathfrak{F}$ behaves like $\mathfrak{S}$ and uses additional memory to keep track whether the run follows the path π. Whenever the path π is completed and $\mathfrak{S}$ would choose α, the scheduler $\mathfrak{S}$ switches to a new memory mode $x_{t,y}$ with probability $p_{(t,y)}$ for all (t,y) in *Goal*. In this mode it behaves like $\mathfrak{U}$ until *Goal* is reached. Then, it switches to the behavior of $\mathfrak{R}_t$ until t is reached. Afterwards it starts to behave like $\mathfrak{S}$ in memory mode y. As soon as it reaches (s_0, x_0) again it starts over again keeping track of whether π is completed. We claim that $\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M},s_0} \geq \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M},s_0}$.

We express the long-run expectation of both finite memory schedulers as the fraction of the expected accumulated value of expected accumulated weight before goal and the return time to the initial configuration in state s_0 with memory mode x_0 for $\mathfrak{S}$ and memory configuration as initially in the description above for $\mathfrak{F}$. We write $eae^{\mathfrak{S}}$ for that expected accumulated expectation which is

$$\sum_{\zeta \text{ path from } (s_0,x_0) \text{ to } (s_0,x_0)} \mathrm{Pr}^{\mathfrak{S}}_{\mathcal{B},(s_0,x_0)} \sum_{i=0}^{length(\zeta)} \mathbb{E}^{\mathfrak{S}\uparrow\zeta[0\ldots i]}_{\mathcal{M},\zeta[i]}(\oplus Goal)$$

and similarly for $\mathfrak{F}$. For the expected number of steps to return we write $ens^{\mathfrak{S}}$. So, we can write

$$\mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} = \frac{eae^{\mathfrak{S}}}{ens^{\mathfrak{S}}}.$$

The schedulers $\mathfrak{F}$ and $\mathfrak{S}$ only differ on how they reach a state (t,y) in *Goal* to continue like $\mathfrak{S}$ after having completed π. Afterwards, they behave in the same way again as they reach the states (t,y) with the same probabilities by construction of $\mathfrak{F}$. Hence, we can provide bounds on $eae^{\mathfrak{F}}$ and $ens^{\mathfrak{F}}$ by looking at the behavior after π before $\mathfrak{F}$ returns to the behavior of $\mathfrak{S}$. For the expected number of steps to return under $\mathfrak{F}$ we get $ens^{\mathfrak{S}} - p \cdot ens^{\mathfrak{S}}_{(s_1,x_1),\alpha,Goal}$ as a lower bound where $ens^{\mathfrak{S}}_{(s_1,x_1),\alpha,Goal}$ denotes the expected number of steps under $\mathfrak{S}$ from (s_1,x_1) to *Goal* when choosing α first. As an upper bound, we can simply choose $ens^{\mathfrak{S}} + p \cdot T$ because T bounds the expected number of steps $\mathfrak{F}$ needs after π before returning to the behavior of $\mathfrak{S}$ form some state (t,y) in *Goal* on.

For the expected accumulated expectation, we get

$$eae^{\mathfrak{S}} - p \cdot EAE^{ub} + p \cdot (K \cdot \delta - T \cdot E^{ub})$$

as a lower bound. The term $-p \cdot EAE^{ub}$ captures a bound on the expected accumulated expectation that is possibly lost by not following $\mathfrak{S}$ after it would choose α after π anymore. The term $p \cdot K \cdot \delta$ captures that for each $i \leq K$, we have that the expectation after $\pi[0\ldots i]$ is increased by at least $\mathrm{Pr}^{\mathfrak{S}}_{\mathcal{B},\pi[i]}(\pi[i\ldots length(\pi)]) \cdot p_\alpha \cdot \delta$ where p_α is the probability that $\mathfrak{S}$ chooses α after π. In other words, after each prefix of π the expected accumulated weight before reaching *Goal* increases by at least the probability that π is completed and α chosen afterwards times δ as the expected accumulated weight before *Goal* after π when choosing α under $\mathfrak{S}$ is at least δ lower than under $\mathfrak{F}$ as $\mathfrak{F}$ then maximizes the expected accumulated weight while $\alpha \notin Act^{\max}$. Finally, in the in expectation at most T steps $\mathfrak{F}$ needs to switch back to the behavior of $\mathfrak{S}$, it accumulates an expectation of

$-E^{ub}$ in the worst case in each step. So,

$$\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} \geq \frac{eae^{\mathfrak{S}} + p \cdot (K \cdot \delta - T \cdot E^{ub} - EAE^{ub})}{ens^{\mathfrak{S}} - p \cdot ens^{\mathfrak{S}}_{(s_1,x_1),\alpha,Goal}}$$

or

$$\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} \geq \frac{eae^{\mathfrak{S}} + p \cdot (K \cdot \delta - T \cdot E^{ub} - EAE^{ub})}{ens^{\mathfrak{S}} + p \cdot T}.$$

We show that in both cases $\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} \geq \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} = \frac{eae^{\mathfrak{S}}}{ens^{\mathfrak{S}}}$. For the first case, let $a = eae^{\mathfrak{S}}$, $b = ens^{\mathfrak{S}}$, $c = p \cdot (K \cdot \delta - T \cdot E^{ub} - EAE^{ub})$, and $d = -p \cdot ens^{\mathfrak{S}}_{(s_1,x_1),\alpha,Goal}$. So, we want to show that

$$\frac{a+c}{b+d} \geq \frac{a}{b}.$$

As $b + d$ is positive, this is equivalent to

$$a + c \geq a + \frac{ad}{b}.$$

Note that a/b is $\mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}}$ and hence its absolute value is bounded by E^{ub}. The absolute value of d is bounded by $p \cdot ENS^{ub}$. dividing by p and plugging the values back in, we obtain that the inequality we want to prove holds if

$$K \cdot \delta - T \cdot E^{ub} - EAE^{ub} \geq E^{ub} \cdot ENS^{ub}.$$

That explains the choice that $K \geq (T \cdot E^{ub} + EAE^{ub} + E^{ub} \cdot ENS^{ub})/\delta$. For the second possibility, we let d be $p \cdot T$. By the same analysis, we obtain that the inequality holds, if $K \geq (EAE^{ub} + 2E^{ub} \cdot T)/\delta$.

This finishes the proof that $\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} \geq \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}}$. Under $\mathfrak{F}$ action α is not chosen after the path π anymore, but instead actions from $Act^{\max}$ are chosen. At the same time, under $\mathfrak{F}$ there are no new such paths of length greater than K after which an action not in $Act^{\max}$ is chosen. (As $K \geq |S| + 1$ also during the execution of $\mathfrak{R}_s$ no such choices are made). As $\mathfrak{S}$ only chooses actions in $Act^{\max}$ whenever $N_{\mathfrak{S}}$-many states not in *Goal* have been visited consecutively, there are only finitely many paths of length at least K after which actions not in $Act^{\max}$ are chosen. We can repeat the construction for these paths one by one to obtain a finite-memory scheduler $\mathfrak{T}$ that maximizes the expectation of $\diamondplus Goal$ whenever K steps not visiting *Goal* have been made. This scheduler still satisfies $\mathrm{LRE}^{\mathfrak{T}}_{\mathcal{M}} \geq \mathrm{LRE}^{\mathfrak{S}}_{\mathcal{M}} \geq \mathrm{LRE}^{\max}_{\mathcal{M}} - \epsilon$. □

Computing optimal long-run expectations. In the sequel, we describe an algorithm for the computation of optimal long-run expectations exploiting the existence of a saturation point.

Given a strongly connected MDP $\mathcal{M} = (S, Act, \Pr, wgt, s_{init}, Goal)$ with a designated set of states *Goal* that intersects all end components, we can compute the saturation point K as in the previous section in polynomial time. We work with a weight function *wgt* from S to $\mathbb{Z}$. We know that it is sufficient to consider schedulers that maximize the expected weight before reaching *Goal* as soon as K consecutive states not in *Goal* have been visited. We construct an MDP $\mathcal{M}_K$ as follows:

- The state space of $\mathcal{M}_K$ is $S_K = ((S \setminus Goal) \times \{1, \ldots, K, \top\}) \cup Goal$.
- The actions of $\mathcal{M}_K$ are the actions *Act* of $\mathcal{M}$. In states of the form (s, K) or $(s, \top)$, however, only actions from $Act^{\max}(s)$ are enabled.
- The transition probability function $\Pr_K$ is defined as follows:

  For states $g, h \in Goal$, we have $\Pr_K(g, \alpha, h) = \Pr(g, \alpha, h)$.

  For a state $g \in Goal$ and a state $s \in S \setminus Goal$, we have $\Pr_K(g, \alpha, (s, 1)) = \Pr(g, \alpha, s)$ and $\Pr_K((s, i), \alpha, g) = \Pr(s, \alpha, g)$ for all $i \in \{1, \ldots, K, \top\}$.

  For states (s, i) and (t, j) in $(S \setminus Goal) \times \{1, \ldots, K, \top\}$, we have $\Pr_K((s, i), \alpha, (t, j)) = \Pr(s, \alpha, t)$ if $j = i + 1$, or if $j = \top$ and i is K or $\top$.

  All remaining transitions have probability 0.
- The weight function wgt_K is defined as follows:

  For $g \in Goal$, $wgt_K(g) = wgt(g)$.

  For states (s, i) with $i < K$, we have $wgt_K((s, i)) = i \cdot wgt(s)$.

  For states (s, K), we have $wgt_K((s, K)) = K \cdot \mathbb{E}^{\max}_{\mathcal{M},s}(\oplus Goal)$.

  For states (s, K), we have $wgt_K((s, \top)) = \mathbb{E}^{\max}_{\mathcal{M},s}(\oplus Goal)$.
- The initial state is s_{init} or $(s_{init}, 1)$.

Lemma 4.29. *Let $\mathcal{M}$ be a strongly connected MDP with a designated set of states Goal that intersects all end components. Let $\mathcal{M}_K$ be the MDP constructed above where K is a saturation point for $\mathcal{M}$. Then, the maximal long-run expectation in $\mathcal{M}$ is equal to the maximal mean-payoff in $\mathcal{M}_K$.*

Proof. There is a one-to-one correspondence between schedulers for $\mathcal{M}_K$ and schedulers $\mathfrak{S}$ for $\mathcal{M}$ that maximize $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(\oplus Goal)$ as soon as K consecutive states not in *Goal* have been visited.

We show that for each such finite-memory scheduler $\mathfrak{F}$ for $\mathcal{M}$ and the corresponding finite-memory scheduler $\mathfrak{F}'$ for $\mathcal{M}_K$, we have that $\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} = \mathbb{E}^{\mathfrak{F}'}_{\mathcal{M}_K}(\mathrm{MP})$. As we know that $\mathrm{LRE}^{\max}_{\mathcal{M}}$ is the supremum over this value for such finite-memory schedulers and $\mathbb{E}^{\max}_{\mathcal{M}_K}(\mathrm{MP})$ is the supremum over all memoryless schedulers for $\mathcal{M}_K$, which correspond to such finite-memory schedulers for $\mathcal{M}$, this is sufficient to conclude that the optimal values are equal.

So, let $\mathfrak{F}$ be a finite-memory scheduler for $\mathcal{M}$ that maximize $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(\diamondplus \mathit{Goal})$ as soon as K consecutive states not in *Goal* have been visited. Let $\mathfrak{F}'$ be the corresponding finite-memory scheduler for $\mathcal{M}_K$. The induced finite Markov chains $\mathcal{C}^{\mathfrak{F}}$ and $\mathcal{C}^{\mathfrak{F}'}$ have the same structure and we denote them by $\mathcal{C}$ form now on. In particular, if we start in a state (g, x) with $g \in \mathit{Goal}$ for some memory mode x, the distribution over path lengths until another state in *Goal* is reached and the distribution over which state in *Goal* that is are the same in both Markov chains. We want to prove that also the expected value of the accumulated sum over $\mathbb{E}^{\mathfrak{F}}_{\mathcal{M},(s,y)}(\diamondplus \mathit{Goal})$ before returning to *Goal* and the expected accumulated weight according to wgt_K before returning to *Goal* are the same.

Let $\Pi_{(g,x)}$ be the set of all finite paths in $\mathcal{C}$ ending in a state in *Goal*. The expected value of the accumulated sum over $\mathbb{E}^{\mathfrak{F}}_{\mathcal{M},(s,y)}(\diamondplus \mathit{Goal})$ before returning to *Goal* is the following value

$$\begin{aligned}
&\sum_{\pi \in \Pi_{(g,x)}} \Pr(\pi) \cdot \sum_{i=0}^{\mathit{length}(\pi)} \mathbb{E}^{\mathfrak{F}}_{\mathcal{M},\pi[i]}(\diamondplus \mathit{Goal}) \\
&= \sum_{\pi \in \Pi_{(g,x)}} \Pr(\pi) \cdot \sum_{i=1}^{\mathit{length}(\pi)} \sum_{\rho \in \Pi_{(g,x)}, \pi[0\ldots i] \in \mathit{Pref}(\rho)} \frac{\Pr(\rho)}{\Pr(\pi[0\ldots i])} \mathit{wgt}(\rho) \\
&= \sum_{\pi' \in \mathit{Pref}(\Pi_{(g,x)}), \mathit{length}(\pi') \geq 1} \Pr(\pi') \cdot \sum_{\rho \in \Pi_{(g,x)}, \pi' \in \mathit{Pref}(\rho)} \frac{\Pr(\rho)}{\Pr(\pi')} \mathit{wgt}(\rho) \\
&= \sum_{\pi' \in \mathit{Pref}(\Pi_{(g,x)}), \mathit{length}(\pi') \geq 1} \quad \sum_{\rho \in \Pi_{(g,x)}, \pi' \in \mathit{Pref}(\rho)} \Pr(\rho) \cdot \mathit{wgt}(\rho) \\
&= \sum_{\rho \in \Pi_{(g,x)}} \Pr(\rho) \cdot \sum_{i=0}^{\mathit{length}(\rho)} \mathit{wgt}(\rho[i]) \cdot \text{ (number of prefixes of } \rho[0\ldots i] \text{ of length} \geq 1) \\
&= \sum_{\rho \in \Pi_{(g,x)}} \Pr(\rho) \cdot \sum_{i=0}^{\mathit{length}(\rho)} \mathit{wgt}(\rho[i]) \cdot i = \sum_{\rho \in \Pi_{(g,x)}} \Pr(\rho) \cdot \mathit{wgt}_K(\rho)
\end{aligned}$$

So, indeed the expected value of the accumulated expected value of $\diamondplus \mathit{Goal}$ in $\mathcal{M}$ under $\mathfrak{F}$ equals the expected weight before reaching goal according to wgt_K under $\mathfrak{F}'$. As also the distribution over which states in *Goal* are reached next and how long that takes are the same, it follows that the long-run expectation $\mathrm{LRE}^{\mathfrak{F}}_{\mathcal{M}} = \mathbb{E}^{\mathfrak{F}'}_{\mathcal{M}_K}(\mathrm{MP})$. □

As a direct consequence of the lemma, we obtain the following result:

Theorem 4.30. *Let $\mathcal{M}$ be a strongly connected MDP with a designated set of states Goal that intersects all end components. The optimal long-run expectation $\mathrm{LRE}^{\max}_{\mathcal{M}}$ can be computed in exponential time.*

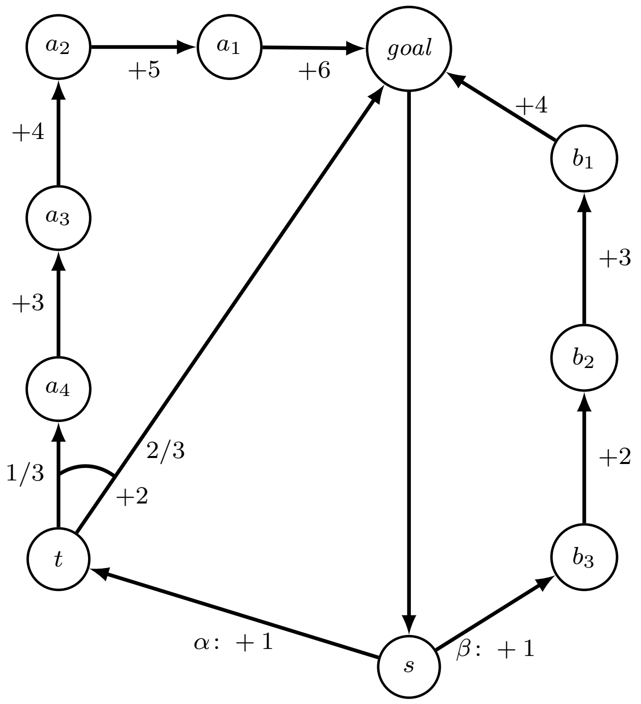


Figure 4.8: Illustration of the computation of long-run expectations via expected mean payoffs.

Example 4.31. In Example 4.25, we considered the long-run expectation in a simple MDP. Each state-action pair was assigned weight 1 in that example. Here, we depict the MDP again in Figure 4.8 with the update weights as in the construction presented above. As each state is only reachable along exactly one path from *goal* before the path returns to *goal*, the construction yields a simple result. With the new weight function, it is now not hard to compute the expected mean payoff under $\mathfrak{S}_\alpha$ always choosing α. Using the steady state probabilities of the states that are given in Example 4.25, we obtain

$$\frac{3}{13}\cdot(1+2)+\frac{1}{13}\cdot(3+4+5+6)=\frac{27}{13}.$$

For $\mathfrak{S}_\beta$, we obtain an expected mean payoff of 2 – the values agree with the long-run expectations we computed in Example 4.25.

The representation as a mean payoff sheds some light on the counter-intuitive observations in Example 4.25: Long paths even with small probability have a larger impact on the long-run expectation than on the expected return time because they contain many states from which the expected time until the next goal state is high. For the expected return time only the number of states but not the expected time from these states on plays a role. ◁

CHAPTER

FIVE

POSITIVITY-HARDNESS

The goal of this chapter is to show that a series of optimization problems on MDPs – including non-classical stochastic shortest path problems, the conditional value-at-risk for accumulated weights, the long-run probability of regular co-safety properties, and several related problems – possesses an inherent mathematical difficulty that makes a solution with known techniques unlikely. We obtain these results by reductions from the *Positivity problem* to the respective decision versions of the problems:

Definition 5.1 (Positivity problem)**.** The Positivity problem for linear recurrence sequences asks whether such a sequence stays non-negative. More formally, given a natural number $k \geq 2$, and rationals α_i and β_j with $1 \leq i \leq k$ and $0 \leq j \leq k-1$, let $(u_n)_{n\geq 0}$ be defined by the initial values $u_0 = \beta_0$, ..., $u_{k-1} = \beta_{k-1}$ and the linear recurrence relation

$$u_{n+k} = \alpha_1 u_{n+k-1} + \cdots + \alpha_k u_n$$

for all $n \geq 0$. The Positivity problem asks to decide whether $u_n \geq 0$ for all n.[1] The number k is called the order of the linear recurrence sequence.

The Positivity problem is closely related to the famous *Skolem problem.* The Skolem problem asks whether there is an n such that $u_n = 0$ for a given linear recurrence sequence $(u_n)_{n\geq 0}$. It is well-known that the Skolem problem is polynomial-time reducible to the Positivity problem (see, e.g., [EvdPSW03]). The Positivity problem and the Skolem problem are outstanding problems in the fields of number theory and theoretical computer science (see, e.g., [HHHK05, OW12, OW15]). Their decidability has been open for many decades. Deep results establish decidability for both problems for linear recurrence sequences of low order or for restricted classes of sequences

[1] We do not distinguish between the Positivity problem and its complement in the sequel. So, we also refer to the problem whether there is an n such that $u_n < 0$ as the Positivity problem.

[STM84, Ver85, OW14a, OW14b, OW14c]. A proof of decidability or undecidability of the Positivity problem for arbitrary sequences, however, withstands all known number-theoretic techniques. In [OW14b], it is shown that decidability of the Positivity problem (already for linear recurrence sequences of order 6) would entail a major breakthrough in the field of Diophantine approximation of transcendental numbers, an area of analytic number theory.

We call a problem to which the Positivity problem is reducible *Positivity-hard.* From a complexity theoretic point of view, the Positivity problem is known to be at least as hard as the decision problem for the universal theory of the reals [OW14c], a problem known to be coNP-hard and to lie in PSPACE [Can88]. As most of the problems we address are PSPACE-hard, the reductions in this chapter do not provide new lower bounds on the computational complexity. The hardness results in this section hence refer to the far-reaching consequences on major open problems that a decidability result would imply. Furthermore, of course the undecidability of the Positivity problem would entail the undecidability of any Positivity-hard problem.

The proof idea we develop in this chapter turns out to be applicable to a variety of problems:

Main result. The Positivity problem is polynomial-time reducible to the threshold problems for the optimal values of the following quantities:

- partial and conditional expectations,
- a two-sided version of partial expectations in MDPs with non-negative weights,
- long-run probabilities of regular co-safety properties,
- termination probabilities and termination times of one-counter MDPs,
- the satisfaction probabilities of energy objectives,
- the probability to satisfy an inequality on the incurred costs (cost problems), and
- conditional values-at-risk for accumulated weights before reaching a goal.

Furthermore, an algorithm for

- the model-checking problem of frequency-LTL (as defined in [FK15, FKK15]), or
- the computation of quantiles for accumulated weights (before reaching a goal)

would imply the decidability of the Positivity-problem.

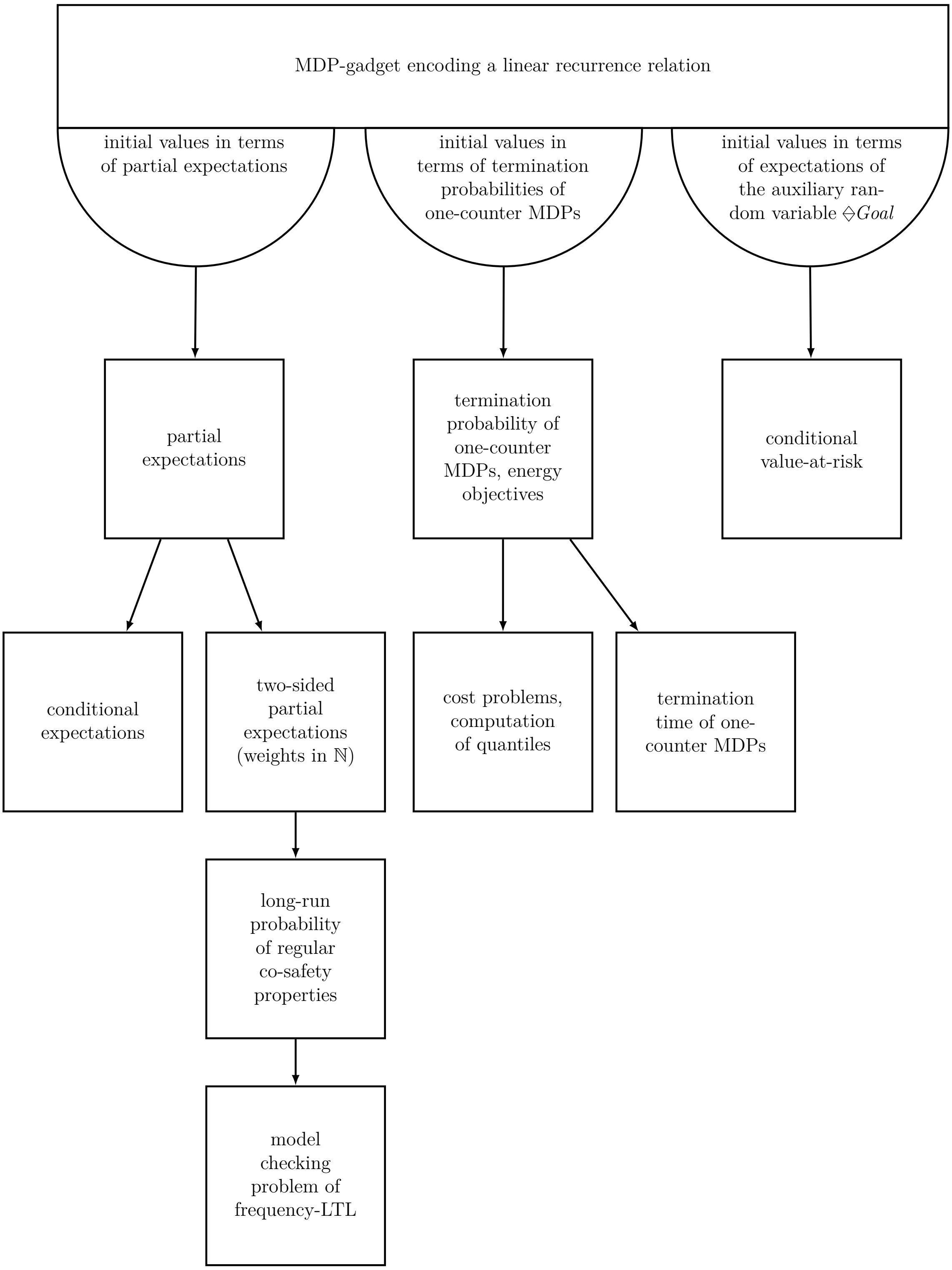

MDP-gadget encoding a linear recurrence relation
initial values in terms of partial expectations
initial values in terms of termination probabilities of one-counter MDPs
initial values in terms of expectations of the auxiliary random variable $\diamondsuit Goal$
partial expectations
termination probability of one-counter MDPs, energy objectives
conditional value-at-risk
conditional expectations
two-sided partial expectations (weights in $\mathbb{N}$)
cost problems, computation of quantiles
termination time of one-counter MDPs
long-run probability of regular co-safety properties
model checking problem of frequency-LTL

Figure 5.1: Overview of the dependencies between the Positivity-hardness results. If not stated otherwise, the squares refer to the threshold problems for the respective quantities.

Outline. To obtain these results, we construct an MDP-gadget in which a linear recurrence relation can be encoded in terms of the optimal values for a variety of optimization problems (Section 5.1). To be able to conduct a reduction from the Positivity problem, we afterwards construct gadgets encoding also the initial values of a linear recurrence sequence – first, in terms of optimal partial expectations (Section 5.2.1). By the inter-reducibility of the threshold problems, the Positivity-hardness for conditional expectations follows directly from the Positivity-hardness for partial expectations. Furthermore, we can adjust the construction slightly to obtain a Positivity-hardness result for a two-sided version of partial expectations using two weight functions with non-negative weights. Relying on the possibility to express long-run probabilities in terms of expected mean payoffs established in Chapter 4, the hardness for two-sided partial expectations allows us to prove the Positivity-hardness of the threshold problem of optimal long-run probabilities of regular co-safety properties (Section 5.2.2). This result in turn implies that the model-checking problem for a simple fixed frequency-LTL formula is Positivity-hard as well. The optimal termination probabilities of one-counter MDPs behave similarly to optimal partial expectations in terms of the dependency of the optimal values on the optimal values at successor states. We can hence reuse the MDP-gadget for linear recurrence relations to show Positivity-hardness of the threshold problem for the termination probability of one-counter MDPs after constructing a further gadget encoding initial values of a linear recurrence sequence (Section 5.2.3). Afterwards, we show how to adapt the construction to obtain the same result for the termination time of almost surely terminating one-counter MDPs. The Positivity-hardness of the termination problem of one-counter MDPs has immediate consequences for cost problems, the computation of quantiles, and the satisfaction probability of energy-objectives. Finally, we prove Positivity-hardness of the threshold problem for the optimal conditional-value-at-risk for accumulated weights. We obtain the result by proving the optimization of the expectation of an auxiliary random variable $\Diamond\!\!\!\!\!-\,\mathit{Goal}$ to be Positivity-hard. This random variable assigns the accumulated weight to a path reaching *Goal* if this weight is negative and 0 if this weight is non-negative (Section 5.2.4). An overview of the dependencies between the Positivity-hardness results is depicted in Figure 5.1.

Related work. In [AAOW15], the Skolem-hardness for decision problems for Markov chains has been established. The problems are to decide whether for given states s, t and rational number p, there is a positive integer n such that the probability to reach t from s in n steps equals p and the model checking problem for a probabilistic variant of monadic logic and a variant of LTL that treats Markov chains as linear transformers of probability distributions (a connection between similar problems and the Skolem problem has also been conjectured in [BRS06, AAGT15]). These decision problems are of quite different nature than the problems studied here, and so are the reductions from the Skolem problem. In this context also the results of [COW16] and [MSS20] are remarkable as they show the decidability (subject to Schanuel's conjecture) of reachability problems in continuous linear dynamical systems and continuous-time MDPs, respectively, as instances of the continuous Skolem problem. In other areas of formal verification, the Skolem problem and the Positivity problem play an important role in the context of the termination of linear programs [BAGM12, Tiw04, Bra06, OW15].

The Positivity-hardness results leave the possibility open that the problems under consideration are undecidable. Known undecidability results on MDPs typically require switching to more expressive models. This applies, e.g., to recursive MDPs [EY05], MDPs with two or more weight functions [BKKW14, RRS17] or partially observable MDPs [MHC99, BGB12]. We are, however, not aware of natural decision problems for standard (finite-state) MDPs with a single weight function and single objective that are known to be undecidable. Remarkable undecidability results in this context are also presented in [KK15]: The hardness of deciding almost sure termination and almost sure termination with finite expected termination time for purely probabilistic programs formulated in the probabilistic fragment of probabilistic guarded command language (pGCL) [MMM05] is pinpointed to levels of the arithmetical hierarchy (for details on the arithmetical hierarchy, see, e.g., [Odi92]). The results reach up to Π_3^0-completeness for deciding universal almost sure termination with finite expected termination time (Π_1^0-complete problems are already undecidable while still co-recursively enumerable). Undecidability is not surprising as the programs subsume ordinary programs. But the universal halting problem for ordinary programs is only Π_2^0-complete showing that deciding universal termination with finite expected termination time of probabilistic programs is strictly harder. Similarly deciding termination from a given initial configuration is Σ_1^0-complete for ordinary programs (halting problem) while deciding almost sure termination with finite expected termination time for probabilistic programs from a given initial configuration is Σ_2^0-complete. Operational semantics of pGCL-programs can be given as infinite-state MDPs [GKM14]. Applied to the purely probabilistic fragment, this leads to infinite-state Markov chains.

Note on the publication of the results. The proof technique presented in this chapter and several of the Positivity-hardness results have been published in joint work with Christel Baier at ICALP 2020 [PB20]. In this chapter, we extend these results by additionally providing the Positivity-hardness proofs for the problems addressing one-counter MDPs, cost constraints, energy objectives, and the computation of quantiles (Section 5.2.3).

5.1 MDP-gadget for linear recurrence relations

The MDP-gadget we construct in this section will form the basis to all Positivity-hardness proofs in this chapter. Let us start by the following observations on the relation between the optimal values at different states for different starting weights in stochastic shortest path and related problems. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP. The solution to the classical stochastic shortest path problem satisfies the so called *Bellman equation*. If $V(s)$ denotes the value when starting in state s, i.e., the maximal expected accumulated weight before reaching *Goal* from state s, then

$$V(s) = \max_{\alpha \in Act(s)} wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot V(t)$$

for $s \notin Goal$ and $V(s) = 0$ for $s \in Goal$. For the partial stochastic shortest path problem, we have to include the weight accumulated so far into the equation as we have seen in Chapter 3. So, let $V(s, w)$ denote the maximal partial expectation when starting in state s with weight w. Letting $V(s, w) = w$ if $s \in Goal$ and $V(s, w) = 0$ if *Goal* is not reachable from s, we obtain the following equation for all remaining states s:

$$V(s, w) = \max_{\alpha \in Act(s)} \sum_{t \in S} P(s, \alpha, t) \cdot V(t, w + wgt(s, \alpha)). \qquad (*)$$

Already in this equation, the value $V(s, w)$ hence possibly depends on the values $V(s, w-1)$, …, $V(s, w-k)$ for some k. We want to exploit this interrelation to encode linear recurrence relations

$$u_{n+k} = \alpha_1 u_{n+k-1} + \cdots + \alpha_k u_n$$

into the optimal values $V(s, w)$. Of course, the values $P(s, \alpha, t)$ are all non-negative. So, we cannot directly encode a linear recurrence into the optimal values for different weight levels at one state as the coefficients can be negative. To overcome this problem, we instead consider the difference $V(s, w) - V(t, w)$ for two different states s and t.

Given the coefficients $\alpha_1, \ldots, \alpha_k$ of a linear recurrence relation, we assume that the coefficients are all sufficiently small for the following construction – which is justified by the argument provided at the end of this section. We construct the MDP-gadget depicted in Figure 5.2. The gadget contains states *goal*, s, and t, as well as $s_1, \ldots, s_k$ and $t_1, \ldots, t_k$. In state t, an action γ is enabled which has weight 0 and leads to state t_i with probability α_i if $\alpha_i > 0$ and to state s_i with probability $|\alpha_i|$ if $\alpha_i < 0$ for all i. The remaining probability leads to *goal*. From each state t_i, there is an action leading to t with weight $-i$. The action δ enabled in s as well as the actions leading from states s_i to s are constructed in the analogously. If α_i is negative, action δ reaches state t_i with probability $|\alpha_i|$. Otherwise it reaches s_i with probability α_i. As the gadget depends on the inputs $\alpha_1, \ldots, \alpha_k$, we call it $\mathcal{G}_{\bar{\alpha}}$.

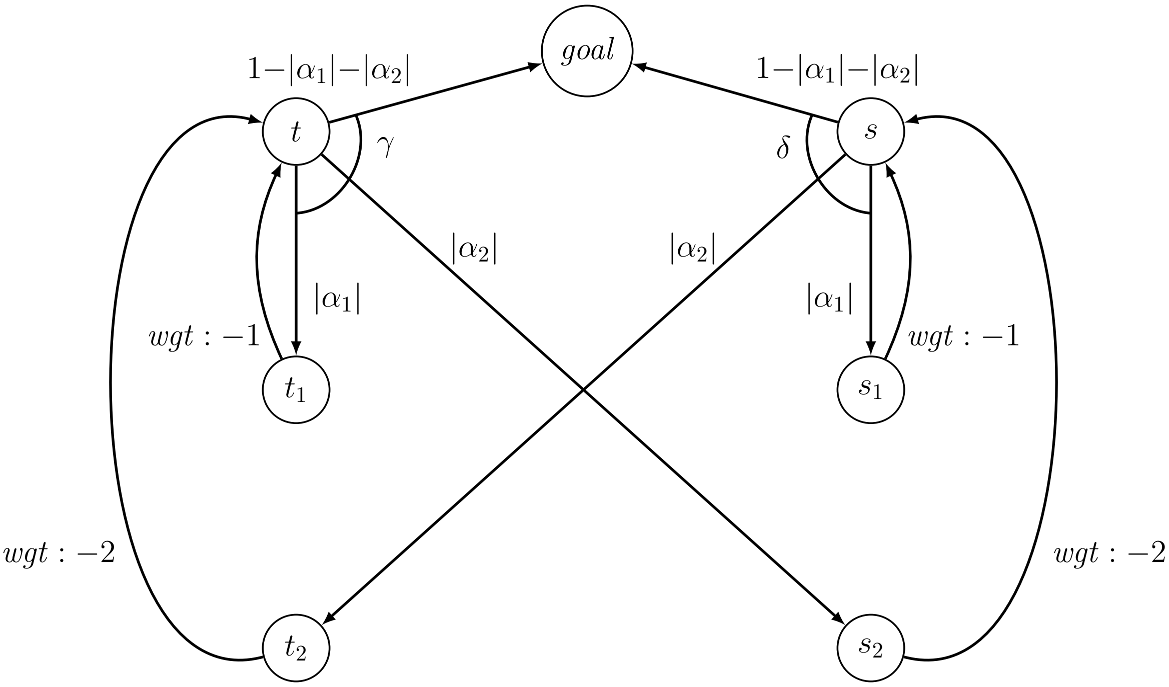


Figure 5.2: The gadget $\mathcal{G}_{\bar{\alpha}}$ to encode linear recurrence relations. The example here is depicted for a linear recurrence of depth 2 with $\alpha_1 \geq 0$ and $\alpha_2 < 0$.

This gadget $\mathcal{G}_{\bar{\alpha}}$ will be integrated into MDPs without further outgoing edges from states $s_1, \ldots, s_k, t_1, \ldots, t_k$. For any optimization problem for which the optimal values V depend on the state and the weight accumulated so far and satisfy Equation $(*)$, we can encode a linear recurrence in an MDP containing this gadget (and possibly further actions for state t and s): If we know that an optimal scheduler chooses action γ in state t and action δ in state s if the accumulated weight is w, then

$$\begin{aligned}
&V(t,w) - V(s,w) \\
&= \left(1 - \sum_{i=1}^{k} |\alpha_i|\right) (V(goal,w) - V(goal,w)) + \\
&\quad \sum_{1\leq i\leq k,\, \alpha_i \geq 0} \alpha_i V(t, w{-}i) - \alpha_i V(s, w{-}i) + \sum_{1\leq i\leq k,\, \alpha_i<0} (-\alpha_i) V(s, w{-}i) + (-\alpha_i) V(t, w{-}i) \\
&= \sum_{i=1}^{k} \alpha_i (V(t, w{-}i) - V(s, w{-}i)).
\end{aligned}$$

Note that this linear recurrence relation also holds for the optimal values in the classical stochastic shortest path problem for example. So, the gadget alone is not yet enough for a hardness proof. The missing ingredient is the encoding of the initial values of a linear recurrence sequence. In order to include the encoding of the initial values in our approach, it is necessary that optimal schedulers cannot be chosen to be memoryless. If the optimal scheduler can be chosen weight-based but not memoryless in general, we aim to encode the initial values by adding further outgoing actions to the states t and s. By fine-tuning the weights and probabilities of these actions, we can achieve that for small weights w some of the new actions are optimal while for large weights the actions γ and δ of the gadget are optimal. If we manage to design the other actions such that the differences $V(t,w+i)-V(s,w+i)$ are equal to given starting values β_i for a sequence of weights $w,w+1,\dots,w+k$ while actions γ and δ are optimal for weights larger than $w+k$, we can encode arbitrary linear recurrence sequences. This is the goal of the subsequent sections.

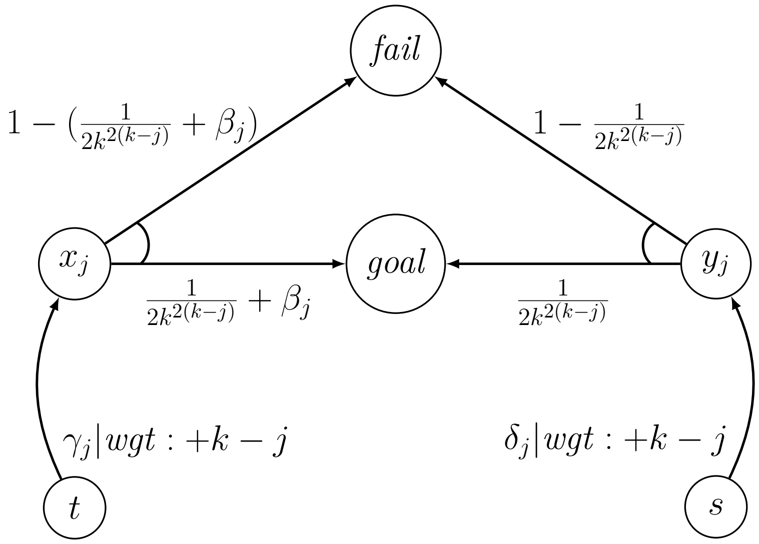
To conclude the section, we provide the missing argument that we can indeed assume that the initial values of a given linear recurrence sequence and the coefficients of the linear recurrence relation are small without loss of generality: Let $(u_n)_{n\geq 0}$ be a linear recurrence sequence specified by the initial values $u_0=\beta_0$, $\dots$, $u_{k-1}=\beta_{k-1}$ and the linear recurrence relation $u_{n+k}=\alpha_1 u_{n+k-1}+\dots+\alpha_k u_n$ for all $n\geq 0$. For any $\mu>0$ and $\lambda>0$, the sequence $(v_n)_{n\geq 0}$ defined by $v_n=\mu\cdot\lambda^n\cdot u_n$ for all n is non-negative if and only if $(u_n)_{n\geq 0}$ is non-negative. Furthermore, it satisfies $v_i=\mu\cdot\lambda^i\cdot\beta_i$ for $i<k$ and

$$v_{n+k}=\lambda\cdot\alpha_1\cdot v_{n+k-1}+\lambda^2\cdot\alpha_2\cdot v_{n+k-1}+\dots+\lambda^k\cdot\alpha_k\cdot v_n.$$

By choosing λ and μ appropriately, we can scale down the initial values and coefficients of the recurrence relation for any given input. We will use this argument throughout the chapter.

5.2 Reductions from the Positivity problem

To encode initial values of a linear recurrence sequence, we construct further MDP gadgets. For partial expectations and the termination probability and time of one-counter MDPs, we can construct these gadgets directly. Putting together these gadgets with the gadget $\mathcal{G}_{\bar{\alpha}}$ from the previous section, we obtain the basis for the Positivity-hardness results of the respective threshold problems. The Positivity-hardness of the remaining problems is obtained as a consequence of these results, possibly via some auxiliary steps. In any case, the gadget $\mathcal{G}_{\bar{\alpha}}$ constructed in the previous section lies at the heart of the reductions.



fail
$1 - (\frac{1}{2k^{2(k-j)}} + \beta_j)$
$1 - \frac{1}{2k^{2(k-j)}}$
x_j
goal
y_j
$\frac{1}{2k^{2(k-j)}} + \beta_j$
$\frac{1}{2k^{2(k-j)}}$
$\gamma_j | wgt: +k-j$
$\delta_j | wgt: +k-j$
t
s

Figure 5.3: The gadget $\mathcal{P}_{\bar{\beta}}$ encoding the initial values in the reduction to the threshold problem for partial expectations.

5.2.1 Partial and conditional stochastic shortest path problems

In this section, we show how the initial values of a linear recurrence sequence can be encoded in terms of optimal partial expectations in a way consistent with the encoding of a linear recurrence relation in $\mathcal{G}_{\bar{\alpha}}$. Afterwards, we show how to exploit this encoding to provide a reduction from the Positivity problem to the threshold problem for partial expectations.

Let k be a natural number and let $(u_n)_{n\geq 0}$ be the linear recurrence sequence given by rationals α_i for $1 \leq i \leq k$ and β_j for $0 \leq j \leq k{-}1$ via $u_0 = \beta_0, \ldots, u_{k-1} = \beta_{k-1}$ and $u_{n+k} = \alpha_1 u_{n+k-1} + \cdots + \alpha_k u_n$ for all $n \geq 0$. W.l.o.g., we can assume that $\sum_i |\alpha_i| < \frac{1}{4}$ and that $0 \leq \beta_j < \frac{1}{4k^{2k+2}}$ for all j as we have argued above.

Now we construct a gadget $\mathcal{P}_{\bar{\beta}}$ that encodes the initial values $\beta_0, \ldots, \beta_{k-1}$. The gadget is depicted in Figure 5.3 and contains states t, s, *goal*, and *fail*. For each $0 \leq j \leq k-1$, it additionally contains states x_j and y_j. In state x_j, there is one action enabled that leads to *goal* with probability $\frac{1}{2k^{2(k-j)}} + \beta_j$ and to *fail* otherwise. From state y_j, *goal* is reached with probability $\frac{1}{2k^{2(k-j)}}$ and *fail* otherwise. In state t, there is an action γ_j leading to x_j with weight $k-j$ for each $0 \leq j \leq k-1$. Likewise, in state s there is an action δ_j leading to y_j with weight $k{-}j$ for each $0 \leq j \leq k-1$.

We now glue together the two gadgets $\mathcal{G}_{\bar{\alpha}}$ and $\mathcal{P}_{\bar{\beta}}$ at states s, t, and *goal*. Finally, we equip the MDP with a simple initial gadget (see Figure 5.4): From the initial state s_{init}, one action with weight +1 is enabled. This action leads to a state c with probability $\frac{1}{2}$ and loops back to s_{init} with probability $\frac{1}{2}$. In c, the decision between action τ leading to state t and action σ leading to state s has to be made. Let us call the full MDP that we obtain in this way $\mathcal{M}$.

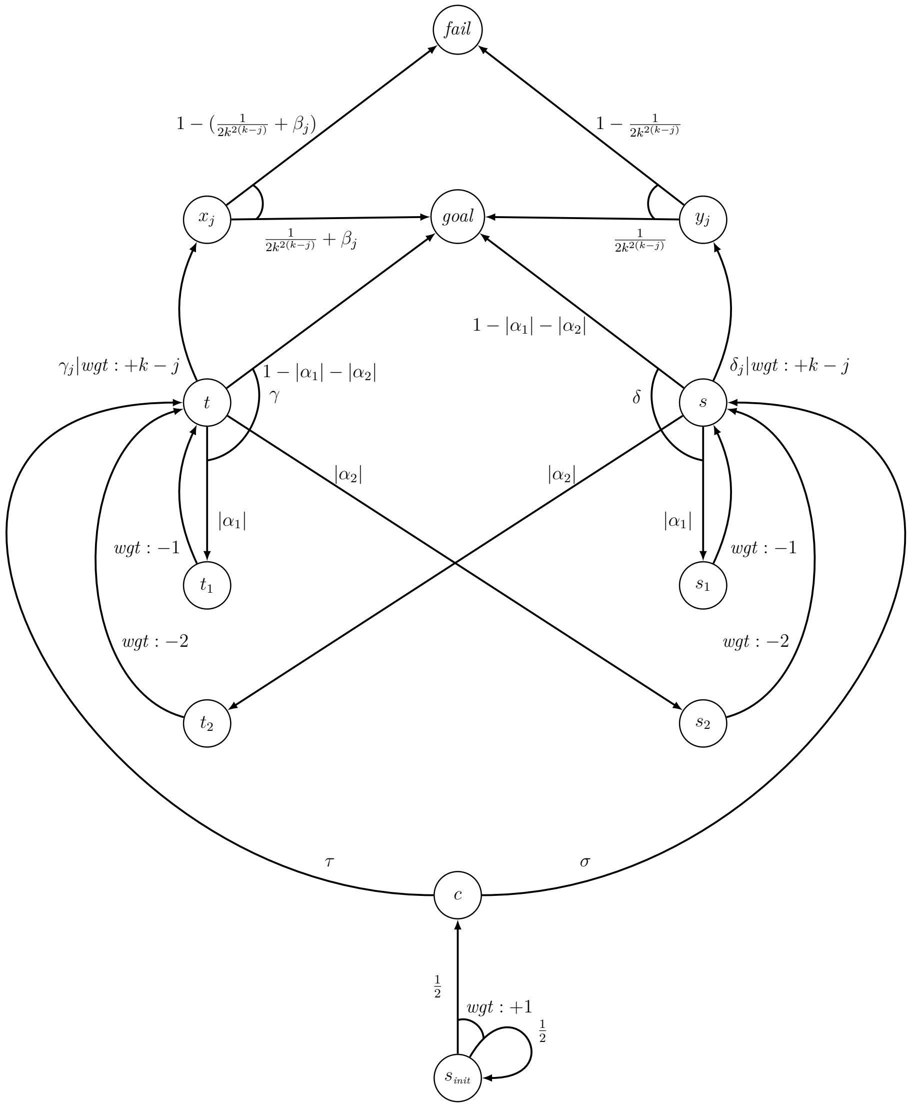


Figure 5.4: The full MDP for the Positivity-hardness proof for partial expectations. The MDP contains the upper part for all $0 \leq j \leq k-1$. The middle part is depicted for $k=2$, $\alpha_1 \geq 0$, and $\alpha_2 < 0$.

The cumbersome choices of probability values lead to the following lemma showing the correct interplay between the gadgets constructed via straight-forward computations.

Lemma 5.2. *Consider the full MDP $\mathcal{M}$. Let $0 \leq j \leq k-1$. Starting with weight $-(k-1)+j$ in state t or s, action γ_j and δ_j maximize the partial expectation. For positive starting weight, γ and δ are optimal.*

Proof. Suppose action γ_i is chosen in state t when starting with weight $-(k-1)+j$. So, state x_i is reached with weight $-(k-1)+j+(k-i)=1+j-i$. Then the partial expectation achieved from this situation is

$$(1+j-i)(\frac{1}{2k^{2(k-i)}}+\beta_i).$$

For $i > j$ this value is ≤ 0 and hence γ_i is certainly not optimal. For $i=j$, we obtain a partial expectation of

$$\frac{1}{2k^{2(k-j)}}+\beta_j.$$

For $i < j$, state x_i is reached with weight $1+j-i \leq k$. Further, $\beta_i \leq \frac{1}{4k^{2k+2}}$ and $\frac{1}{2k^{2(k-i)}} \leq \frac{1}{2k^{2(k-j)} \cdot k^2}$. So, the partial expectation obtained via γ_i is at most

$$\frac{k}{2k^{2(k-j)}\cdot k^2}+\frac{k}{4k^{2k+2}}<\frac{1}{2k^{2(k-j)}}.$$

So, indeed action γ_j maximizes the partial expectation among the actions γ_i with $0 \leq i \leq k-1$ when the accumulated weight in state t is $-(k-1)+j$. The argument for state s is the same with $\beta_i = 0$ for all i. It is easy to see that for accumulated weight $-(k-1)+j$ with $0 \leq j \leq k-1$ actions γ or δ are not optimal in state t or s: If *goal* is reached immediately, the weight is not positive and otherwise states t or s are reached with lower accumulated weight again. The values β_j are chosen small enough such that also a switch from state t to s while accumulating negative weight does not lead to a higher partial expectation.

For positive accumulated weight w, the optimal partial expectation when choosing γ first is at least $\frac{3}{4}w$ by construction and the fact that a positive value can be achieved from any possible successor state via one of the actions γ_j and δ_j with $0 \leq j \leq k-1$. Choosing γ_j on the other hands results in a partial expectation of at most $(k+w)\cdot(\frac{1}{4k^{2k+2}}+\frac{1}{2k^2})$ which is easily seen to be less as $k \geq 2$. □

For each weight w, denote by $e(t,w)$ and $e(s,w)$ the optimal partial expectation when starting in state t or s with accumulated weight w in $\mathcal{M}$ as if the respective state was reached from the initial state with weight w and probability 1. For each weight $w \geq -k+1$, denote by $d(w)$ the difference $e(t,w)-e(s,w)$ between these optimal partial expectation when starting in state t and s with weight w. Comparing action γ_j and δ_j for starting weight $-(k-1)+j$, we conclude that the difference between optimal values $d(-(k-1)+j)$ is equal to β_j, for $0 \leq j \leq k-1$. By the fact that $\mathcal{G}_{\bar{\alpha}}$ encodes the given linear recurrence relation as soon as γ and δ are the optimal actions as shown in Section 5.1, we conclude the following lemma:

Lemma 5.3. *Consider the linear recurrence sequence $(u_n)_{n\geq 0}$ given above by $\alpha_1, \ldots, \alpha_k$ and $\beta_0, \ldots, \beta_{k-1}$ and the MDP $\mathcal{M}$ constructed from this sequence. We have*

$$d(-(k-1)+n) = u_n$$

for all n with the values $d(w)$ just defined.

Let us now consider a run of the MDP $\mathcal{M}$. For any $w > 0$, state c is reached with accumulated weight w with positive probability. An optimal scheduler now has to decide whether the partial expectation when starting with weight w is better in state s or t: Action τ is optimal in c for accumulated weight w if and only if $d(w) \geq 0$. Once t or s is reached, the optimal actions are given by Lemma 5.2. Let $\mathfrak{S}$ be the scheduler that always chooses τ in c and actions $\gamma, \gamma_0, \ldots, \gamma_{k-1}, \delta, \ldots$ as described in Lemma 5.2. We conclude that $\mathfrak{S}$ is optimal if and only if the given linear recurrence sequence is non-negative. We can compute the partial expectation of scheduler $\mathfrak{S}$ in the constructed MDP. The partial expectation turns out to be a rational. Hence, using this partial expectation as the threshold ϑ, we obtain the Positivity-hardness of the threshold problem for partial expectations.

Theorem 5.4. *The Positivity problem is polynomial-time reducible to the following problem: Given an MDP $\mathcal{M}$ and a rational ϑ, decide whether $\mathbb{PE}^{\max}_{\mathcal{M}} > \vartheta$.*

Proof. We will compute the partial expectation of scheduler $\mathfrak{S}$ always choosing τ in c and actions $\gamma, \gamma_0, \ldots, \gamma_{k-1}, \delta, \ldots$ as described in Lemma 5.2 in the constructed MDP $\mathcal{M}$ depicted in Figure 5.4: Recall that the scheduler $\mathfrak{S}$ chooses γ and δ, respectively, as long as the accumulated weight is positive. For an accumulated weight of $-(k-1)+j$ for $0 \leq j \leq k-1$, it chooses actions γ_j and δ_j, respectively.

We want to recursively express the partial expectations under $\mathfrak{S}$ starting from t or s with some positive accumulated weight $n \in \mathbb{N}$ which we again denote by $e(t,n)$ and $e(s,n)$, respectively. In order to do so, we consider the following Markov chain $\mathcal{C}$ for $n \in \mathbb{N}$ that is also depicted in Figure 5.5 for the case $k = 2$: The Markov chain $\mathcal{C}$ has $5k$ states named t_{-k+1}, ..., t_{+k}, s_{-k+1}, ..., s_{+k}, and $goal_{+1}$, ..., $goal_{+k}$. States t_{-k+1}, ..., t_0, s_{-k+1}, ..., s_0, and $goal_{+1}$, ..., $goal_{+k}$ are absorbing. For $0 < i, j \leq k$, there are transitions from t_{+i} to t_{+i-j} with probability α_j if $\alpha_j > 0$, to s_{+i-j} with probability $|\alpha_j|$ if $\alpha_j < 0$, and to $goal_{+i}$ with probability $1 - |\alpha_1| - \ldots - |\alpha_k|$. Transitions from s_{+i} are defined analogously.

The idea behind this Markov chain is that the reachability probabilities describe how, for arbitrary $n \in \mathbb{N}$ and $1 \leq i \leq k$, the values $e(t, nk+i)$ and $e(s, nk+i)$ depend on n and the values $e(t, (n-1)k+j)$ and $e(s, (n-1)k+j)$ for $1 \leq j \leq k$. The transitions in $\mathcal{C}$ behave as γ and δ in $\mathcal{M}$, but the decrease in the accumulated weight is explicitly

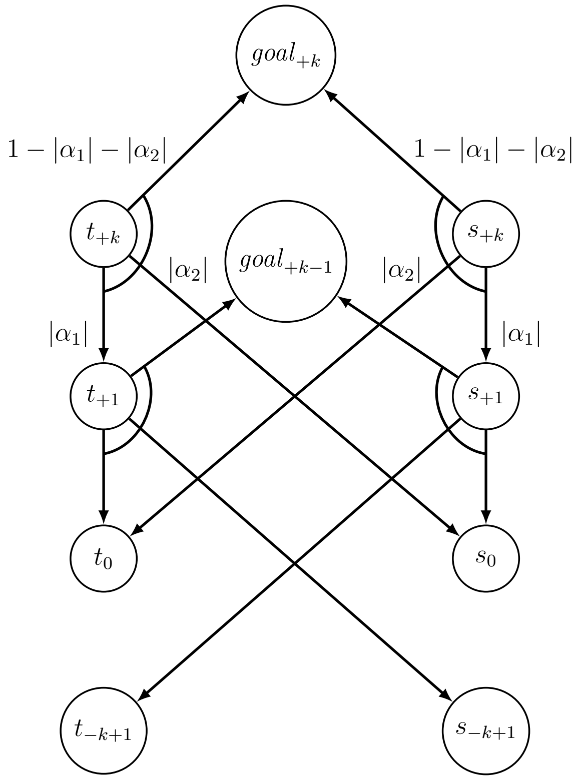


Figure 5.5: The Markov chain $\mathcal{C}$ depicted for $k = 2$ with $\alpha_1 \geq 0$ and $\alpha_2 < 0$.

encoded into the state space. Namely, for $n \in \mathbb{N}$ and $0 < i \leq k$, we have

$$\begin{aligned} e(t, nk+i) = \sum_{j=1}^{k} \Big(& \mathrm{Pr}_{\mathcal{C},t_{+i}}(\Diamond t_{-k+j}) \cdot e(t,(n{-}1)k+j) \\ & + \mathrm{Pr}_{\mathcal{C},t_{+i}}(\Diamond s_{-k+j}) \cdot e(s,(n{-}1)k+j) \Big) \\ + \sum_{j=1}^{k} & \mathrm{Pr}_{\mathcal{C},t_{+i}}(\Diamond goal_{+j}) \cdot (nk+j) \end{aligned} \tag{5.1}$$

and analogously for $e(s, nk + i)$. We now group the optimal values together in the following vectors

$$v_n = (e(t, nk+k), e(t, nk+k-1), \ldots, e(t, nk+1), e(s, nk+k), \ldots, e(s, nk+1))^t$$

for $n \in \mathbb{N}$. In other words, this vector contains the optimal values for the partial expectation when starting in t or s with an accumulated weight from $\{nk+1, \ldots, nk+k\}$. Further, we define the vector containing the optimal values for weights in $\{-k+1, \ldots, 0\}$

which are the least values of accumulated weight reachable under scheduler $\mathfrak{S}$.

$$v_{-1} = (e(t,0), e(t,-1), \ldots, e(t,-k+1), e(s,0), e(s,-1), \ldots, e(s,-k+1))^t.$$

As we have seen, these values are given as follows:

$$e(t,-k+1+j) = \frac{1}{2k^{2(k-j)}} + \beta_j \text{ and } e(s,-k+1+j) = \frac{1}{2k^{2(k-j)}}$$

for $0 \leq j \leq k-1$.

As the reachability probabilities in $\mathcal{C}$ are rational and computable in polynomial time, we conclude from (5.1) that there are a matrix $A \in \mathbb{Q}^{2k \times 2k}$, and vectors a and b in $\mathbb{Q}^{2k}$ computable in polynomial time such that

$$v_n = Av_{n-1} + na + b,$$

for all $n \in \mathbb{N}$. We claim that the following explicit representation for $n \geq -1$ satisfies this recursion:

$$v_n = A^{n+1}v_{-1} + \sum_{i=0}^{n}(n-i)A^i a + \sum_{i=0}^{n} A^i b.$$

We show this by induction: Clearly, this representation yields the correct value for v_{-1}. So, assume $v_n = A^{n+1}v_{-1} + \sum_{i=0}^{n}(n-i)A^i a + \sum_{i=0}^{n} A^i b$. Then,

$$\begin{aligned} v_{n+1} &= A(A^{n+1}v_{-1} + \sum_{i=0}^{n}(n-i)A^i a + \sum_{i=0}^{n} A^i b) + (n+1)a + b \\ &= A^{n+2}v_{-1} + \left(\sum_{i=0}^{n}(n-i)A^{i+1}a\right) + (n+1)A^0 a + \left(\sum_{i=1}^{n+1} A^i b\right) + A^0 b \\ &= A^{n+2}v_{-1} + \sum_{i=0}^{n+1}(n+1-i)A^i a + \sum_{i=0}^{n+1} A^i b. \end{aligned}$$

So, we have an explicit representation for v_n. The value we are interested in is

$$\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} = \sum_{\ell=1}^{\infty}(1/2)^{\ell} e(t,\ell).$$

Let $c = (\frac{1}{2^k}, \frac{1}{2^{k-1}}, \ldots, \frac{1}{2^1}, 0, \ldots, 0)$. Then,

$$(\frac{1}{2^k})^n c \cdot v_n = \sum_{i=1}^{k} \frac{1}{2^{nk+i}} e(t, nk+i).$$

Hence, we can write

$$\begin{aligned}\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}} &= \sum_{n=0}^{\infty}(\frac{1}{2^k})^n c\cdot v_n = c\cdot \sum_{n=0}^{\infty}(\frac{1}{2^k})^n v_n \\ &= c\cdot \sum_{n=0}^{\infty}(\frac{1}{2^k})^n (A^{n+1}v_{-1} + \sum_{i=0}^{n}(n-i)A^i a + \sum_{i=0}^{n} A^i b) \\ &= c\cdot \Big((\sum_{n=0}^{\infty}(\frac{1}{2^k})^n A^{n+1})v_{-1} + (\sum_{n=0}^{\infty}(\frac{1}{2^k})^n \sum_{i=0}^{n}(n-i)A^i)a + (\sum_{n=0}^{\infty}(\frac{1}{2^k})^n \sum_{i=0}^{n} A^i)b\Big).\end{aligned}$$

We claim that all of the matrix series involved converge to rational matrices. A key observation is that the maximal row sum in A is at most $|\alpha_1|+\ldots+|\alpha_k| < 1$ because the rows of the matrix contain exactly the probabilities to reach $t_0, \ldots t_{-k+1}, s_0, \ldots,$ and s_{-k+1} from a state t_{+i} or s_{+i} in $\mathcal{C}$ for $1\leq i\leq k$. But the probability to reach $goal_{+i}$ from these states is already $1-|\alpha_1|-\ldots-|\alpha_k|$. Hence, $\|A\|_\infty$, the operator norm induced by the maximum norm $\|\cdot\|_\infty$, which equals $\max_i \sum_{j=1}^{2k}|A_{ij}|$, is less than 1.

So, of course also $\|\frac{1}{2^k}A\|_\infty < 1$ and hence the Neumann series $\sum_{n=0}^{\infty}(\frac{1}{2^k}A)^n$ converges to $(I_{2k}-\frac{1}{2^k}A)^{-1}$ where I_{2k} is the identity matrix of size $2k\times 2k$. So,

$$\sum_{n=0}^{\infty}(\frac{1}{2^k})^n A^{n+1} = A\sum_{n=0}^{\infty}(\frac{1}{2^k}A)^n = A(I_{2k}-\frac{1}{2^k}A)^{-1}.$$

Note that $\|A\|_\infty < 1$ also implies that $I_{2k}-A$ is invertible. We observe that for all n,

$$\sum_{i=0}^{n} A^i = (I_{2k}-A)^{-1}(I_{2k}-A^{n+1})$$

which is shown by straight-forward induction. Therefore,

$$\begin{aligned}\sum_{n=0}^{\infty}(\frac{1}{2^k})^n \sum_{i=0}^{n} A^i &= (I_{2k}-A)^{-1}\left(\sum_{n=0}^{\infty}(\frac{1}{2^k})^n I_{2k} - A\sum_{n=0}^{\infty}(\frac{1}{2^k}A)^n\right) \\ &= (I_{2k}-A)^{-1}\left(\frac{2^k}{2^k-1}I_{2k} - A(I_{2k}-\frac{1}{2^k}A)^{-1}\right).\end{aligned}$$

Finally, we show by induction that

$$\sum_{i=0}^{n}(n-i)A^i = (I_{2k}-A)^{-2}(A^{n+1}-A+n(I_{2k}-A)).$$

This is equivalent to

$$(I_{2k}-A)^2\sum_{i=0}^{n}(n-i)A^i = A^{n+1}-A+n(I_{2k}-A).$$

For $n = 0$, both sides evaluate to 0. So, we assume the claim holds for n.

$$\begin{aligned}(I_{2k}-A)^2\sum_{i=0}^{n+1}(n+1-i)A^i &= (I_{2k}-A)^2\sum_{i=0}^{n}(n-i)A^i + (I_{2k}-A)^2\sum_{i=0}^{n}A^i \\ &\overset{\text{IH}}{=} A^{n+1}-A+n(I_{2k}-A)+(I_{2k}-A)^2\sum_{i=0}^{n}A^i \\ &= A-A^{n+1}+n(I_{2k}-A)+(I_{2k}-A)^2(I_{2k}-A)^{-1}(I_{2k}-A^{n+1}) \\ &= A-A^{n+1}+n(I_{2k}-A)+I_{2k}-A-A^{n+1}+A^{n+2} \\ &= A^{n+2}-A+(n+1)(I_{2k}-A).\end{aligned}$$

The remaining series is the following:

$$\begin{aligned}&\sum_{n=0}^{\infty}(\frac{1}{2^k})^n\sum_{i=0}^{n}(n-i)A^i \\ =\ &\sum_{n=0}^{\infty}(\frac{1}{2^k})^n(I_{2k}-A)^{-2}(A^{n+1}-A+n(I_{2k}-A)) \\ =\ &(I_{2k}-A)^{-2}\left(\sum_{n=0}^{\infty}(\frac{1}{2^k})^nA^{n+1}-\sum_{n=0}^{\infty}(\frac{1}{2^k})^nA+\sum_{n=0}^{\infty}(\frac{1}{2^k})^nn(I_{2k}-A)\right) \\ =\ &(I_{2k}-A)^{-2}\left(A(I_{2k}-\frac{1}{2^k}A)^{-1}-\frac{2^k}{2^k-1}A+\frac{2^k}{(2^k-1)^2}(I_{2k}-A)\right).\end{aligned}$$

We conclude that all expressions in the representation of $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}}$ above are rational and computable in polynomial time. As we have seen, the originally given linear recurrence sequence is non-negative if and only if $\mathbb{PE}^{\max}_{\mathcal{M}} \leq \mathbb{PE}^{\mathfrak{S}}_{\mathcal{M}}$ for the MDP $\mathcal{M}$ constructed from the linear recurrence sequence in polynomial time in the previous sections. □

We have obtained the first Positivity-hardness result of this chapter. The proof contains all ingredients that we need for the subsequent proofs. In particular, the computation of the value of a scheduler that is optimal iff the given linear recurrence sequence is non-negative will be very similar in later proofs.

Remark 5.5. There is no obvious way to adjust the construction such that the Positivity-hardness of the question whether $\mathbb{PE}^{\max}_{\mathcal{M}} \geq \vartheta$ would follow. One attempt would be to provide an ε such that $\mathbb{PE}^{\max}_{\mathcal{M}} > \vartheta$ iff $\mathbb{PE}^{\max}_{\mathcal{M}} \geq \vartheta + \varepsilon$. This, however, probably requires a bound on the position at which the given linear recurrence sequence first becomes negative. But this question lies at the core of the Positivity problem. The analogous observation applies to all Positivity-hardness results in the sequel.

The Positivity-hardness of the threshold problem for conditional expectations now follows directly by the inter-reducibility of the threshold problems for partial and conditional expectations (see Proposition 3.16):

Theorem 5.6. *The Positivity problem is reducible in polynomial time to the following problem: Given an MDP $\mathcal{M}$ and a rational ϑ, decide whether $\mathbb{CE}^{\max}_{\mathcal{M}} > \vartheta$.*

Two-sided partial expectation. As we have already seen in Chapter 3, the non-monotonic behavior of weights along a path prohibits the switch to a memoryless behavior of an optimal scheduler for large weights in the partial and conditional stochastic shortest path problem (see Section 3.2.3). Instead of using arbitrary integer weights, we can simulate such non-monotonic behavior with two non-negative weight functions: In the definition of the random variable $\oplus \mathit{Goal}$, we can replace the choice that paths not reaching *Goal* are assigned weight 0 by a second weight function. Let $\mathcal{M} = (S, \mathit{Act}, \Pr, s_{\mathit{init}}, \mathit{wgt}_{\mathit{goal}}, \mathit{wgt}_{\mathit{fail}}, \mathit{goal}, \mathit{fail})$ be an MDP with two designated absorbing states *goal* and *fail* and two non-negative weight functions $\mathit{wgt}_{\mathit{goal}} \colon S \times \mathit{Act} \to \mathbb{N}$ and $\mathit{wgt}_{\mathit{fail}} \colon S \times \mathit{Act} \to \mathbb{N}$. Assume that the probability $\Pr^{\min}_{\mathcal{M},s_{\mathit{init}}}(\Diamond\{\mathit{goal}, \mathit{fail}\}) = 1$. Define the following random variable X on maximal paths ζ:

$$X(\zeta) = \begin{cases} \mathit{wgt}_{\mathit{goal}}(\zeta) & \text{if } \zeta \vDash \Diamond \mathit{goal}, \\ \mathit{wgt}_{\mathit{fail}}(\zeta) & \text{if } \zeta \vDash \Diamond \mathit{fail}. \end{cases}$$

Due to the assumption that *goal* or *fail* is reached almost surely under any scheduler, the expected value $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(X)$ is well-defined for all schedulers $\mathfrak{S}$ for $\mathcal{M}$. We call the value $\mathbb{E}^{\max}_{\mathcal{M},s_{\mathit{init}}}(X) = \sup_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(X)$ the optimal *two-sided partial expectation.* We can show that the threshold problem for the two-sided partial expectation is Positivity-hard as well by a small adjustment of the construction above.

Theorem 5.7. *The Positivity problem is polynomial-time reducible to the following problem: Given an MDP $\mathcal{M} = (S, \mathit{Act}, \Pr, s_{\mathit{init}}, \mathit{wgt}_{\mathit{goal}}, \mathit{wgt}_{\mathit{fail}}, \mathit{goal}, \mathit{fail})$ as above and a rational ϑ, decide whether $\mathbb{E}^{\max}_{\mathcal{M},s_{\mathit{init}}}(X) > \vartheta$.*

Proof. Given the parameters $\alpha_1, \dots, \alpha_k$ and $\beta_0, \dots, \beta_{k-1}$ of a rational linear recurrence sequence, we can construct an MDP $\mathcal{M}' = (S, \mathit{Act}, \Pr, s_{\mathit{init}}, \mathit{wgt}, \mathit{goal}, \mathit{fail})$ with one weight function $\mathit{wgt} \colon S \times \mathit{Act} \to \mathbb{Z}$ similar to the MDP $\mathcal{M}$ depicted in Figure 5.4. W.l.o.g., we again assume that $\sum_i |\alpha_i| < \frac{1}{4}$ and that $0 \leq \beta_j < \frac{1}{4k^{2k+2}}$ for all j. The initial gadget and the gadget $\mathcal{G}_{\bar{\alpha}}$ are as before. The gadget $\mathcal{P}_{\bar{\beta}}$, however, is slightly modified and replaced by the gadget $\mathcal{T}_{\bar{\beta}}$ depicted in Figure 5.6. For this gadget, we define $\alpha = \sum_{i=1}^{k} |\alpha_i|$, $p_1 = (1-\alpha)(\frac{1}{2k^{2(k-j)}} + \beta_j)$, $p_2 = (1-\alpha)(1 - (\frac{1}{2k^{2(k-j)}} + \beta_j))$, $q_1 = (1-\alpha)\frac{1}{2k^{2(k-j)}}$, and $q_2 = (1-\alpha)(1 - \frac{1}{2k^{2(k-j)}})$. With the transitions as in the figure, the probability to reach *goal* or *fail* and the weight accumulated does not change when choosing action γ_j or δ_j compared to the gadget $\mathcal{P}_{\bar{\beta}}$. The only difference is that the expected time to reach *goal* or *fail* changes. The steps alternate between probability α and probability 0 to reach *goal* or *fail* – just as in the gadget $\mathcal{G}_{\bar{\alpha}}$. In this way, it makes no difference for the expected

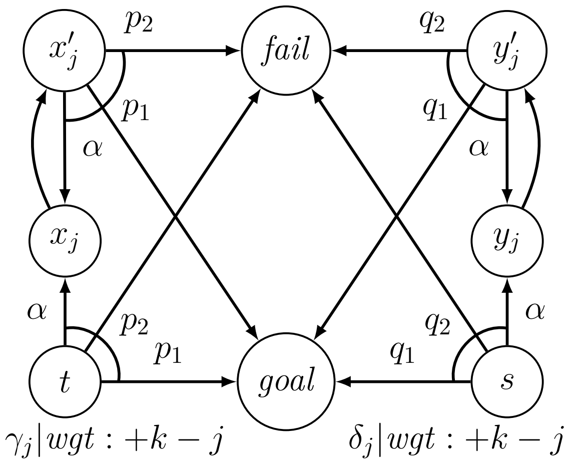


Figure 5.6: The gadget $\mathcal{T}_{\overline{\beta}}$ encoding initial values in terms of two-sided partial expectations.

time before reaching *goal* or *fail* when a scheduler stops choosing γ and δ. We can, in fact, compute the expected time T to reach *goal* or *fail* from s_{init} under any scheduler quite easily: Reaching t or s takes 3 steps in expectation. Afterwards, two further steps are taken $1/\alpha$-many times in expectation. So,

$$T = 3 + \frac{2}{\alpha}.$$

The optimal scheduler $\mathfrak{S}$ for the partial expectation in $\mathcal{M}'$ is the same as in the MDP $\mathcal{M}$ above. Also the value ϑ of this scheduler can be computed as in Theorem 5.4. So, $\mathbb{PE}^{\max}_{\mathcal{M}',s_{\mathit{init}}} > \vartheta$ if and only if the given linear recurrence sequence is eventually negative.

Note that all weights in $\mathcal{M}'$ are $\geq -k$. We define two new weight functions to obtain an MDP $\mathcal{N}$ from $\mathcal{M}'$: We let $\mathit{wgt}_{\mathit{goal}}(s,\alpha) = \mathit{wgt}(s,\alpha) + k$ and $\mathit{wgt}_{\mathit{fail}}(s,\alpha) = +k$ for all $(s,\alpha) \in S \times \mathit{Act}$. Both weight functions take only non-negative integer values.

Any scheduler $\mathfrak{S}$ for $\mathcal{M}'$ can be viewed as a scheduler for $\mathcal{N}$, and vice versa, as the two MDPs only differ in the weight functions. Further, we observe that for each maximal path ζ ending in *goal* or *fail* in $\mathcal{M}'$ and at the same time in $\mathcal{N}$, we have $X(\zeta) = \oplus \mathit{goal}(\zeta) + k \cdot \mathit{length}(\zeta)$. (Recall that $\oplus \mathit{goal}(\zeta)$ equals $\mathit{wgt}(\zeta)$ if ζ reaches *goal* and 0 if ζ reaches *fail*.) As the expected time before *goal* or *fail* is reached is constant, namely T under any scheduler, it follows that for all schedulers $\mathfrak{T}$ we have

$$\mathbb{E}^{\mathfrak{T}}_{\mathcal{N},s_{\mathit{init}}}(X) = \mathbb{PE}^{\mathfrak{T}}_{\mathcal{M}',s_{\mathit{init}}} + k \cdot T.$$

Therefore, $\mathbb{E}^{\max}_{\mathcal{N},s_{\mathit{init}}}(X) > \vartheta + k \cdot T$ if and only if the given linear recurrence sequence eventually becomes negative. □

While the two-sided partial expectation is certainly interesting in its own right, it will also play an important role in the proof of the Positivity-hardness of the threshold problem for the optimal long-run probability of a regular co-safety property in the next section.

5.2.2 Long-run probabilities and frequency-LTL

In Chapter 4, we have seen that the optimal long-run probability of a regular co-safety can be expressed in terms of an optimal expected mean-payoff. This insight allows us to draw a connection between long-run probabilities and two sided-partial expectations that we just discussed. Recall that for an MDP $\mathcal{M} = (S, \mathit{Act}, \Pr, s_{\mathit{init}}, \mathit{wgt}_{\mathit{goal}}, \mathit{wgt}_{\mathit{fail}}, \mathit{goal}, \mathit{fail})$ with two designated absorbing states *goal* and *fail* and two non-negative weight functions $\mathit{wgt}_{\mathit{goal}} \colon S \times \mathit{Act} \to \mathbb{N}$ and $\mathit{wgt}_{\mathit{fail}} \colon S \times \mathit{Act} \to \mathbb{N}$, the two-sided partial expectation was defined as the expectation of the following random variable X on maximal paths ζ:

$$X(\zeta) = \begin{cases} \mathit{wgt}_{\mathit{goal}}(\zeta) & \text{if } \zeta \vDash \Diamond \mathit{goal}, \\ \mathit{wgt}_{\mathit{fail}}(\zeta) & \text{if } \zeta \vDash \Diamond \mathit{fail}. \end{cases}$$

In Section 4.2.2, we provided a construction given an MDP $\mathcal{M}$ and a DFA $\mathcal{D}$ that encoded at each point in time j of a run ρ of $\mathcal{M}$ how many runs of $\mathcal{D}$ on the suffixes $\rho[i\ldots]$ with $i \leq j$ are currently in which of the states of $\mathcal{D}$. Whenever a label read in $\mathcal{M}$ leads to a transition from a state q in $\mathcal{D}$ to an accepting state, the number of runs in that state q is received as weight in the constructed MDP that we will call $\mathcal{M}_{\mathcal{D}}$. We want to exploit this construction to mimic a behavior similar to the payoff according to the random variable X. Consider the DFA $\mathcal{D}$ depicted in Figure 5.7. The state space is $Q = \{q_{\mathit{init}}, q_1, q_2, \mathit{accept}, \mathit{reject}\}$. The alphabet is $2^{\{a,b,c,\mathit{goal},\mathit{fail}\}}$. From the initial state letters satisfying $a \wedge b \wedge \neg c$ lead to q_1, letters satisfying $a \wedge c \wedge \neg b$ to q_2 and all remaining letters to *reject*. From q_1, letters satisfying $a \wedge \neg \mathit{goal}$ lead back to q_1, letters with $\mathit{goal} \wedge \neg a$ to *accept*, and all remaining letters lead to *reject*. Transitions from q_2 are defined analogously with *goal* replaced by *fail*.

Consider a run ρ of an MDP $\mathcal{M}$ labeled with $\{a, b, c, \mathit{goal}, \mathit{fail}\}$ for which we keep counters of the number of runs on suffixes of ρ in each of the states of $\mathcal{D}$: We only need counters c_1 and c_2 for states q_1 and q_2 as these are the only states multiple runs can be in before being accepted or rejected. The update of the counters as in Section 4.2.2 can directly be determined from the DFA $\mathcal{D}$: E.g., if $\{a, b\}$ is read, counter c_1 is increased; if $\{a, c\}$ is read, counter c_2 is increased. On $\{a\}$, both counters stay the same. If no a is read, the counters are reset to 0. If at the same time *goal* is read, the value of c_1 is received as weight. If *fail* is read, the value of c_2 is received as weight. So, the behavior of the counters is very similar to the accumulation of two non-negative weight functions. Which of the two weight functions or the two counters is used to determine the payoff depends

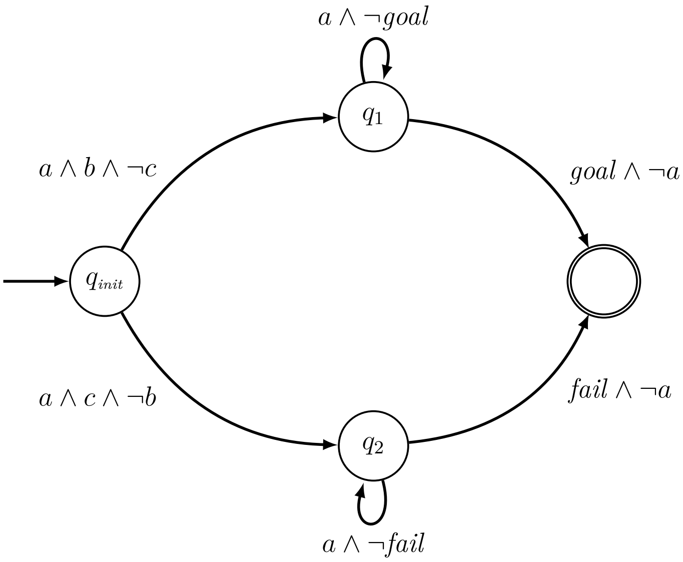


Figure 5.7: The DFA $\mathcal{D}$ mimicking two-sided partial expectations. All transitions on letters not satisfying one of the labels lead to a rejecting sink that is not depicted.

on whether *goal* or *fail* is reached next. In the sequel, we will show that indeed already the fixed co-safety property of this simple DFA $\mathcal{D}$ suffices to prove Positivity-hardness of the threshold problem for long-run probabilities.

The proof of the Positivity-hardness of the threshold problem for the two-sided partial expectation with non-negative weights contains most of the necessary ingredients we need: Let $(u_n)_{n\geq 0}$ be a rational linear recurrence sequence given by initial values $\beta_0, \dots, \beta_{k-1}$ and the coefficients $\alpha_1, \dots, \alpha_k$ of the recurrence. In the proof of Theorem 5.7, we showed that we can construct an MDP $\mathcal{M} = (S, Act, \Pr, s_{init}, wgt_{goal}, wgt_{fail}, goal, fail)$ and rationals ϑ, T with the following properties from the given parameters:

- For the two designated states *goal* and *fail*, we have $\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond\{goal, fail\}) = 1$.
- The expected number of steps until *goal* or *fail* is reached is T under any scheduler.
- The weight functions wgt_{goal} and wgt_{fail} assign a weight between 0 and $2k$ to each state-action pair.
- $\mathbb{E}^{\max}_{\mathcal{M},s_{init}}(X) > \vartheta$ if and only if there is an n with $u_n < 0$.

From this MDP $\mathcal{M}$, we now construct a labeled MDP $\mathcal{K}$. For each state-action pair (s, α) of $\mathcal{M}$ with $s \notin \{goal, fail\}$, we add a chain $r^1_{s,\alpha}, \dots, r^{4k}_{s,\alpha}$ of new states as depicted in Figure 5.8: We redirect the transition from s when choosing α to this chain by setting $P_{\mathcal{K}}(s, \alpha, r^1_{s,\alpha})$. In the states of the chain only one action τ is enabled. The process moves through the chain with probability 1 via this action, i.e., $P_{\mathcal{K}}(r^i_{s,\alpha}, \tau, r^{i+1}_{s,\alpha}) = 1$

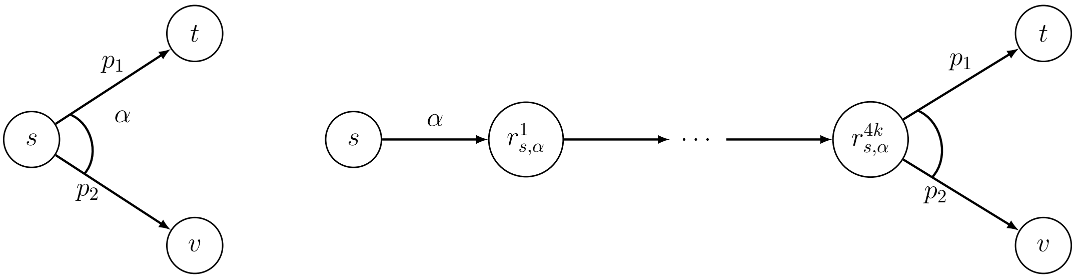


Figure 5.8: Construction of $\mathcal{K}$.

for all $i < 4k$. Then, the original transition is taken from the state $r^{4k}_{s,\alpha}$ by setting $P_{\mathcal{K}}(r^{4k}_{s,\alpha}, \tau, t) = P(s, \alpha, t)$ for all states t of $\mathcal{M}$. Instead of making *goal* and *fail* absorbing, we furthermore add transitions back to the initial state s_{init} from *goal* and *fail* with probability 1. Note that the expected time from s_{init} until s_{init} is reached again from *goal* or *fail* is now $T' = T(4k + 1) + 1$ in $\mathcal{K}$ under any scheduler.

The labeling is now defined as follows: All states except for *goal* and *fail* are labeled with a. States *goal* and *fail* are labeled with their names. Furthermore, in each of the chains $r^1_{s,\alpha}, \dots, r^{4k}_{s,\alpha}$, the first $\mathit{wgt}_{\mathit{goal}}(s, \alpha)$ of the states are labeled with b in addition to the label a. The next $\mathit{wgt}_{\mathit{fail}}(s, \alpha)$ states are labeled with c in addition to the a. As $\mathit{wgt}_{\mathit{goal}}(s, \alpha) + \mathit{wgt}_{\mathit{fail}}(s, \alpha) \leq 4k$, this is always possible.

Consider now a path π of $\mathcal{M}$ from s_{init} to *goal* or *fail*. There is a unique corresponding path $\hat{\pi}$ in $\mathcal{K}$. The counters induced by the DFA $\mathcal{D}$ as described above now behave exactly like the accumulation weight functions $\mathit{wgt}_{\mathit{goal}}$ and $\mathit{wgt}_{\mathit{fail}}$. The counter value c_1 counting the number of runs in state q_1 of $\mathcal{D}$ is precisely $\mathit{wgt}_{\mathit{goal}}(\pi)$ when entering *goal* or *fail* as in each chain of states $r^1_{s,\alpha}, \dots, r^{4k}_{s,\alpha}$ exactly $\mathit{wgt}_{\mathit{goal}}(s, \alpha)$-many runs of $\mathcal{D}$ enter state q_1. The counter c_2 behaves analogously in terms of the weight function $\mathit{wgt}_{\mathit{fail}}$. As all states not in $\{\mathit{goal}, \mathit{fail}\}$ are labeled with a, the counters are also not reset. When entering *goal*, the random variable X assigns weight $\mathit{wgt}_{\mathit{goal}}(\pi)$ to the path π. The same weight is received from the counter c_1 in this case. When entering *fail*, weight $\mathit{wgt}_{\mathit{fail}}(\pi)$ is assigned by X and received from the counters.

As the time required to reach s_{init} again from *goal* or *fail* in $\mathcal{K}$ in expectation is T' under any scheduler, a scheduler maximizing the expected mean payoff in $\mathcal{K}_{\mathcal{D}}$, i.e., according to the weight function induced by the counter for $\mathcal{D}$, hence has to maximize the expected value of X when considered as a scheduler for $\mathcal{M}$. By Theorem 4.16, the maximal mean payoff in $\mathcal{K}_{\mathcal{D}}$ equals the maximal long-run probability $\mathbb{LP}^{\max}_{\mathcal{K},s_{\mathit{init}}}(\mathcal{D})$. Putting these results together, we obtain that

$$\mathbb{LP}^{\max}_{\mathcal{K},s_{\mathit{init}}}(\mathcal{D}) > \frac{\vartheta}{T'}$$

if and only if the given linear recurrence sequence is eventually negative. We conclude the Positivity-hardness result for long-run probabilities:

Theorem 5.8. *There is a fixed DFA $\mathcal{D}$ such that the Positivity problem is polynomial-time reducible to the following problem: Given an MDP $\mathcal{M}$ and a rational χ, decide whether $\mathbb{LP}^{\max}_{\mathcal{M}}(\mathcal{D}) > \chi$.*

Note that the Positivity-hardness holds for the fixed simple DFA $\mathcal{D}$. This DFA contains two states for which we have to keep a counter. In contrast, a DFA for $a \,\mathrm{U}\, b$, contains only one such state and allowed us to prove the existence of a saturation point. This reflects exactly that partial expectations with non-negative weights can be computed via the existence of a saturation point while the threshold problem for two-sided partial expectations with non-negative weights is Positivity-hard.

A consequence of this result is that model checking of frequency-LTL in MDPs is at least as hard as the Positivity problem. The decidability of the model-checking problem for the full logic frequency-LTL has been left open in [FK15, FKK15]. Proving decidability of the model-checking problem hence would settle the decidability of the Positivity problem. The frequency-globally modality $G^{>\vartheta}_{\inf}(\varphi)$ of frequency-LTL is defined to hold on a path π iff

$$\liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} \mathbb{1}_{\pi[i\ldots]\models\varphi} > \vartheta,$$

i.e. iff the long-run frequency of φ exceeds ϑ.

Theorem 5.9. *There is a polynomial-time reduction from the Positivity problem to the following qualitative model checking problem for frequency-LTL for a fixed LTL-formula φ: Given an MDP $\mathcal{M}$ and a rational ϑ, is $\mathrm{Pr}^{\max}_{\mathcal{M}}(G^{>\vartheta}_{\inf}(\varphi)) = 1$?*

Proof. Consider the MDP $\mathcal{K}$, the DFA $\mathcal{D}$, and the threshold $\vartheta' = \vartheta/T'$ constructed above. As the sets of states labeled with b and with c are disjoint and included in the set of states labeled with a, and likewise the sets of states labeled with a, *goal*, and *fail* are pairwise disjoint in $\mathcal{K}$, a path of $\mathcal{K}$ has a prefix accepted by $\mathcal{D}$ if and only if the path satisfies

$$\varphi = (b \wedge (a \,\mathrm{U}\, \mathit{goal})) \vee (c \wedge (a \,\mathrm{U}\, \mathit{fail})).$$

We claim that there is a scheduler $\mathfrak{S}$ with $\mathbb{LP}^{\mathfrak{S}}_{\mathcal{K},s_{init}}(\mathcal{D}) > \vartheta'$ if and only if there is a scheduler $\mathfrak{T}$ such that $G^{>\vartheta}_{\inf}(\varphi)$ holds with probability 1 under $\mathfrak{T}$ in $\mathcal{K}$.

Suppose there is a scheduler with $\mathfrak{S}$ with $\mathbb{LP}^{\mathfrak{S}}_{\mathcal{K}}(\mathcal{D}) > \vartheta'$. By Lemma 4.14, we can assume that $\mathfrak{S}$ is a finite-memory scheduler as the maximal long-run probability can be approximated by finite-memory schedulers. As $\mathcal{K}$ is strongly connected, we can further assume that $\mathfrak{S}$ induces only one BSCC. We claim that under this scheduler $\mathfrak{S}$ also $G^{>\vartheta}_{\inf}(\varphi)$ holds with probability 1. For finite-memory schedulers, it is easy to check that the expected long-run probability equals the expected long-run frequency as we obtain a finite-state Markov chain: Let $x_{\mathfrak{s}}$ be the steady state probability of states $\mathfrak{s}$ enriched with memory modes in the single BSCC $\mathcal{B}^{\mathfrak{S}}$ induced by $\mathfrak{S}$. Further, let $p_{\mathfrak{s}}$ be the probability that a run starting in $\mathfrak{s}$ under $\mathfrak{S}$ satisfies φ. Then, $\mathbb{LP}^{\mathfrak{S}}_{\mathcal{K}}(\mathcal{D}) = \sum_{\mathfrak{s}\in\mathcal{B}^{\mathfrak{S}}} x_{\mathfrak{s}} \cdot p_{\mathfrak{s}}$ (see Section 4.2.1). But the same expression also computes the expected frequency with which φ holds on suffixes as shown in [FK15]. Furthermore, in a strongly connected Markov chain, the frequency of φ along almost all paths agrees with the expected frequency (see [FK15]). So,

$$\liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{1}_{\zeta[i\ldots]\models\varphi} > \vartheta$$

holds on almost all paths ζ.

Conversely, if there is a scheduler $\mathfrak{T}$ such that $G^{>\vartheta}_{\text{inf}}(\varphi)$ holds with probability 1 under $\mathfrak{T}$ in $\mathcal{K}$, the expected value $\mathbb{E}^{\mathfrak{S}}_{\mathcal{K}}(\liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} \mathbb{1}_{\varsigma[i\ldots]\models\varphi}) > \vartheta$. By an argument using Fatou's lemma analogously to the proof of Lemma 4.14, we can find a finite memory scheduler with expected long-run frequency, and hence long-run probability, greater than ϑ. □

5.2.3 One-counter MDPs, energy objectives, cost problems, and quantiles

In the introduction (Chapter 1), we discussed several problems and notions related to stochastic shortest path problems – namely, one-counter MDPs, energy objectives, cost problems, and quantiles. The core decision problem that arises for these notions is the threshold problem for the maximal or minimal probability that the accumulated weight satisfies an inequality constraint. More formally, given an MDP $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}}, \mathit{wgt}, \mathit{Goal})$, an integer w, a probability value $p \in [0,1]$, and two inequality operators $\bowtie, \bowtie' \in \{<, \leq, \geq, >\}$ these questions come in a number of natural variants where the maximum can also be replaced by a minimum:

1. Is $\Pr^{\max}_{\mathcal{M},s_{\mathit{init}}}(\Diamond(\text{accumulated weight} \bowtie w)) \bowtie' p$?

2. Is $\Pr^{\max}_{\mathcal{M},s_{\mathit{init}}}(\diamondplus \mathit{Goal} \bowtie w) \bowtie' p$?

3. Is $\Pr^{\max}_{\mathcal{M},s_{\mathit{init}}}(\text{total accumulated weight} \bowtie w) \bowtie' p$?

In question 3, we encounter the problem that the total accumulated weight of a path is only defined if all transitions have weight 0 from some point on. If we require schedulers to make sure that this is the case on almost all paths, we can alternatively add a state *goal* together with transitions to this new states from all state in end components in which all weights are 0. Instead of considering the total accumulated weight, we could then consider the random variable $\diamondplus \mathit{goal}$ as in question 2. Hence, we focus on the first two questions. Note that in an MDP $\mathcal{M}$ with non-negative weights in which *Goal* is reached almost surely, the first two questions coincide.

For an overview what is known about these questions and one-counter MDPs, energy objectives, cost problems, and quantiles, we refer to the discussion in Chapter 1. The positive results from the literature concern MDPs with non-negative weights, qualitative thresholds ($p = 0$ or $p = 1$), and the approximability of the optimal probabilities. The general questions 1 and 2 in MDPs with integer weights are open.

Recall that a one-counter MDP is a finite-state MDP equipped with a counter and for each transition it is specified whether it increases or decreases the counter or leaves it unchanged (see [BBE+10,BBEK11,BKNW12]). The process starts with counter value 1 is said to terminate as soon as the counter value reaches 0. The change of the counter value behaves exactly like the accumulation of weight in our settings. On the other hand, we can treat a weighted MDP as a one-counter MDP by viewing a transition with weight w as a sequence of $|w|$-many steps in which the counter is increased or decrease, respectively. This comes with the caveat that the size of the one-counter MDP corresponding to a weighted MDP is then polynomial in the size of the weighted MDP and the numeric value of the weights. In other words, the size of the one-counter MDP is roughly the size of the weighted MDP when assuming that weights are encoded in unary. In our setting, the termination of a one-counter MDP corresponds to the event

$$\Diamond(\text{accumulated weight} < 0)$$

that the accumulated weight of a prefix of a path is below 0. The threshold problem for the optimal termination probability of a one-counter MDP is hence an instance of question 1. Besides the optimal termination probability, also the optimal expected time before termination in one-counter MDPs that terminate almost surely has been addressed in the literature. The threshold problem for the optimal expected termination time is open as well. The complement of the termination of a one-counter MDP is described by the condition

$$\Box(\text{accumulated weight} \geq 0)$$

that the weight of all prefixes of a run is non-negative. This is precisely the energy objective (see [CD11, BKN16, MSTW17]).

Question 2 is addressed in [HK15, HKL17, BBD+18] and called the cost problem (with atomic cost formula) in [HK15]. The maximal or minimal value w such that a constraint as in question 2 is satisfied is called a quantile. The computation of such quantiles is addressed in [UB13, BDD+14, RRS17]. Of course, an algorithm computing quantiles can directly be employed to solve the corresponding threshold problem.

In this section, we will show that the threshold problem for the optimal termination probability of a one-counter MDP and hence question 1 is Positivity-hard. Our proof will allow us to conclude that also question 2 is Positivity-hard and hence also an algorithm for the computation of quantiles of the accumulated weight before reaching the goal would imply the decidability of the Positivity problem. Furthermore, we will show that the threshold problem for the optimal expected termination time of almost surely terminating one-counter MDPs is Positivity-hard, too.

Termination probability of one-counter MDPs. First, we will use the above formulation of termination in terms of weighted MDPs to show the result for the termination probability of one-counter MDPs. We reuse the gadget $\mathcal{G}_{\bar{\alpha}}$ which is possible because the maximal termination probabilities $p(s,w)$ in terms of the current state s and counter value (accumulated weight) w satisfy the same optimality equation $(*)$ from Section 5.1 as maximal partial expectations if we set $p(s,w)=1$ for all $w<0$. The missing ingredient is again a gadget to encode the initial values of a linear recurrence sequence. We first address the maximal termination probability and line out the necessary adjustments to show Positivity-hardness also for the threshold problem for minimal termination probabilities afterwards.

Theorem 5.10. *The Positivity problem is reducible in polynomial time to the following problem: Given an MDP $\mathcal{M}$ and a rational $\vartheta \in (0,1)$, decide whether*

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\Diamond(\textit{accumulated weight} < 0)) > \vartheta.$$

Proof. Let $k \geq 2$ be a natural number, $\alpha_1,\dots,\alpha_k$ and $\beta_0,\dots,\beta_{k-1}$ rationals. Let $(u_n)_{n\geq 0}$ be the linear recurrence sequence defined by the given k initial values and the linear recurrence relation $u_{i+k} = \alpha_1 \cdot u_{i+k-1} + \dots + \alpha_k \cdot u_i$ for all $i \geq 0$. W.l.o.g., we can assume that $\sum_{i=1}^{k} |\alpha_i| < 1/(k+1)$ and that $0 \leq \beta_j < 1/(k+1)$ for all $0 \leq j \leq k-1$. We are going to construct an MDP $\mathcal{M}$ and a rational $\vartheta \in (0,1)$ such that

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\Diamond(\text{ accumulated weight} < 0)) > \vartheta \text{ iff } u_n < 0 \text{ for some } n \geq 0.$$

The MDP $\mathcal{M}$ is depicted in Figure 5.9 for $k=2$ and assuming that $\alpha_1 \geq 0$ while $\alpha_2 < 0$. The weight function is called c. The initial gadget and the gadget $\mathcal{G}_{\bar{\alpha}}$ are included as before while *goal* is replaced by an absorbing state *trap*. The new gadget $\mathcal{O}_{\bar{\beta}}$ encoding the initial values $\bar{\beta}$ works as follows: For $0 \leq j \leq k-1$, the action γ_j enabled in t leads to state x_j with probability $\frac{k-j}{k+1} + \beta_j$. By assumption on β_j, this probability is less than $\frac{k-j+1}{k+1}$. The remaining probability leads to *trap*. In state s, the action δ_j leads to y_j with probability $\frac{k-j}{k+1}$ and to *trap* with the remaining probability. For $0 \leq j \leq k-1$, one reaches *trap* from x_j and y_j with probability 1 and a counter change of $-(j+1)$.

In order to terminate, the accumulated weight has to drop below 0 before reaching *trap*. As soon as the trap state is reached with non-negative accumulated weight, the process cannot terminate anymore. The optimal decision in order to maximize the termination probability in state t is now easy to determine. Let ℓ be the current weight. If $0 \leq \ell \leq k-1$, choosing action γ leads to termination with probability less than $1/(k+1)$

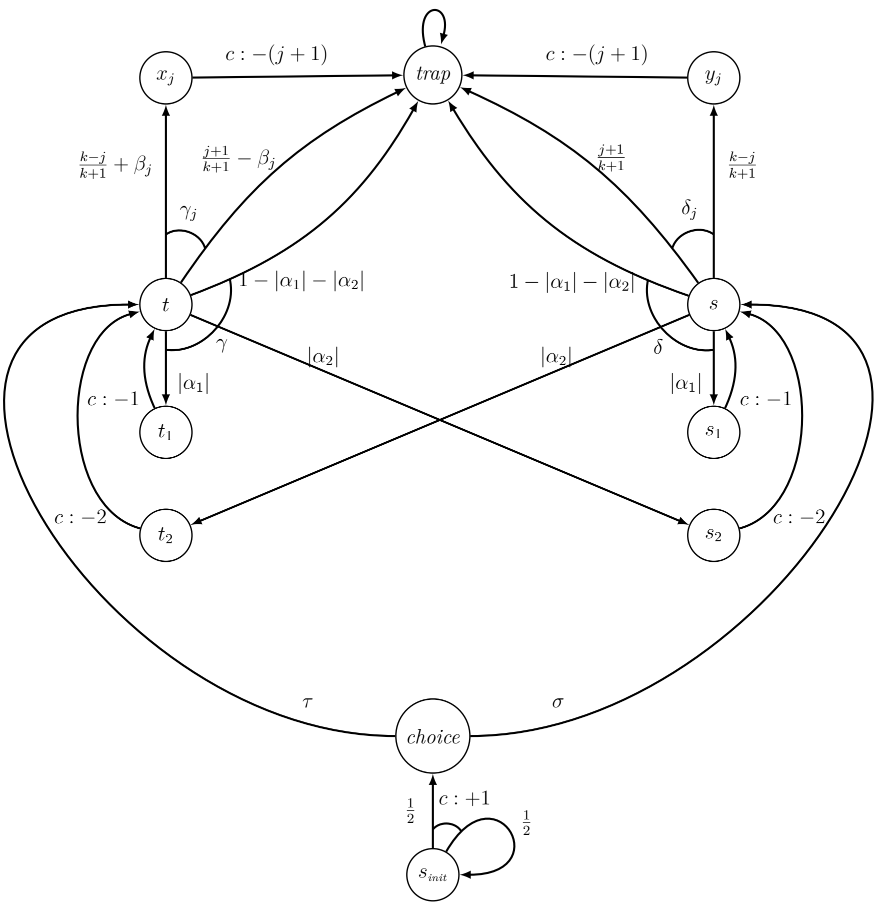


Figure 5.9: Full MDP for the reduction to the threshold problem for termination probabilities of one-counter MDPs. The MDP contains the upper part for all $0 \leq j \leq k-1$. The middle part is depicted for $k = 2$, $\alpha_1 \geq 0$, and $\alpha_2 < 0$.

as *trap* is reached immediately with probability at least $k/(k+1)$. Choosing action γ_j makes it impossible to terminate if $\ell > j$. If $\ell \leq j$, then choosing γ_j lets the process terminate if x_j is reached. This happens with probability $\frac{k-j}{k+1} + \beta_j$. As $\beta_j < 1/(k+1)$ for all j, the maximal termination probability is reached when choosing γ_ℓ. If $\ell \geq k$, then γ_j leads to termination with probability 0 for all j. Hence, action γ is optimal. Analogously, we see that the optimal choice in state s with weight ℓ is δ_ℓ if $\ell \leq k-1$ and δ otherwise.

Let $p(r, w)$ denote the optimal termination probability when starting in state $r \in \{t, s\}$ with accumulated weight $w \geq 0$. The linear recurrence sequence $(u_n)_{n \geq 0}$ now can be found in terms of these optimal values: Consider again the difference $d(w) = p(t, w) - p(s, w)$. For counter value $w \leq k-1$, we have seen that γ_w and δ_w, respectively, are the optimal actions. Hence, $d(w) = u_w$ in this case as we have just seen that the optimal termination probability when starting with weight $w \leq k-1$ is $\frac{k-w}{k+1} + \beta_w$ in t and $\frac{k-w}{k+1}$ in s. Furthermore, for $w > k-1$, actions γ and δ are optimal. So, by the discussion in Section 5.1, the sequence of differences satisfies the linear recurrence relation given by $\alpha_1, \ldots, \alpha_k$. Therefore, $d(w) = u_w$ for all $w \geq 0$. The state *choice* is reached with any positive accumulated weight with positive probability. For the optimal choices in the state *choice* with accumulated weight w, we observe that again choosing τ is optimal iff $d(w) \geq 0$. This in turn holds if and only if $u_w \geq 0$. Consider the scheduler $\mathfrak{S}$ which always chooses τ in state *choice* and afterwards behaves according to the optimal choices as described above. This scheduler $\mathfrak{S}$ is optimal if and only if the sequence $(u_n)_{n\geq 0}$ is non-negative. To complete the reduction, we compute the value

$$\vartheta \overset{\text{def}}{=} \Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Diamond(\text{accumulated weight } < 0)).$$

We will see that ϑ is a rational computable in polynomial time and we know that $\Pr^{\max}_{\mathcal{M},s_{init}}(\Diamond(\text{accumulated weight } < 0)) \leq \vartheta$ if and only if the scheduler $\mathfrak{S}$ is optimal which is the case iff $(u_n)_{n\geq 0}$ is non-negative.

In order to be able to obtain an explicit representation of the optimal termination probabilities $p(t,w)$ and $p(s,w)$, we group these values into segments of k consecutive weight values. We consider the Markov chain $\mathcal{C}$ (see Figure 5.5) again to determine how the values for accumulated weights $(n+1)\cdot k, \ldots, (n+1)\cdot k+k-1$ depend on the values for accumulated weights $n\cdot k, \ldots, n\cdot k+k-1$. Similar to before, for $n \geq 1$ and $0 \leq i < k$, we have

$$\begin{aligned} p(t, nk+i) = \sum_{j=1}^{k} \Big(& \Pr_{\mathcal{C},t_{+i}}(\Diamond t_{-k+j}) \cdot p(t,(n{-}1)k+j) \\ & + \Pr_{\mathcal{C},t_{+i}}(\Diamond t_{-k+j}) \cdot p(s,(n{-}1)k+j) \Big) \end{aligned} \tag{5.2}$$

and analogously for $p(s, nk{+}i)$. We now group the optimal values together in the following vectors

$$v_n = (p(t, nk+k-1), p(t, nk+k-2), \ldots, p(t, nk), p(s, nk+k-1), \ldots, p(s, nk))^t$$

for $n \in \mathbb{N}$. In other words, this vector contains the optimal values for the termination probability when starting in t or s with an accumulated weight from $\{nk, \ldots, nk{+}k{-}1\}$. The vector v_0 is

$$(p(t,k-1), \ldots, p(t,0), \ldots p(s,k-1), \ldots, p(s,0))^t$$

and these values occur as transition probabilities in $\mathcal{M}$ under the actions $\gamma_{k-1}, \ldots, \gamma_0$ and $\delta_{k-1}, \ldots, \delta_0$.

As the reachability probabilities in $\mathcal{C}$ are rational and computable in polynomial time, we conclude from equation (5.2) that there is a matrix $A \in \mathbb{Q}^{2k\times 2k}$ computable in polynomial time such that $v_{n+1} = Av_n$ for all $n \in \mathbb{N}$. So, $v_n = A^n v_0$ for all $n \in \mathbb{N}$.

As state *choice* is reached with weight w with probability $(1/2)^w$ for all $w \geq 1$, the value $\vartheta = \sum_{w=1}^{\infty}(1/2)^w p(t,w)$. Let $c = (\frac{1}{2^k}, \frac{1}{2^{k-1}}, \dots, \frac{1}{2^1}, 0, \dots, 0)$. Observe that for all $n \in \mathbb{N}$,

$$\left(\frac{1}{2^k}\right)^n \cdot c \cdot v_n = \sum_{i=1}^{k} \frac{1}{2^{nk+i}} p(t, nk+i).$$

Hence, we can write

$$\begin{aligned}\vartheta &= \sum_{n=0}^{\infty}\left(\frac{1}{2^k}\right)^n \cdot c\cdot v_n - p(t,0) = c\cdot \sum_{n=0}^{\infty}\left(\frac{1}{2^k}\right)^n \cdot v_n - p(t,0)\\ &= c\cdot \sum_{n=0}^{\infty}\left(\frac{1}{2^k}\right)^n \cdot A^n \cdot v_0 - p(t,0) = c\cdot \left(\sum_{n=0}^{\infty}\left(\frac{1}{2^k}\cdot A\right)^n\right)\cdot v_0 - p(t,0).\end{aligned}$$

We have to subtract $p(t,0)$ as the state *choice* cannot be reached with weight 0, but the summand $1 \cdot p(t,0)$ occurs in the sum. As $p(t,0) = \frac{k}{k+1} + \beta_0$, this does not cause a problem. We claim that the matrix series involved converges to a rational matrix. As before $\|\frac{1}{2^k}A\|_\infty < 1$ and hence the Neumann series $\sum_{n=0}^{\infty}(\frac{1}{2^k}A)^n$ converges to $(I_{2k} - \frac{1}{2^k}A)^{-1}$ where I_{2k} is the identity matrix of size $2k\times 2k$. So,

$$\vartheta = c\cdot (I_{2k} - \frac{1}{2^k}A)^{-1}\cdot v_0 - p(t,0)$$

is computable in polynomial time and

$$\mathrm{Pr}^{\max}_{\mathcal{M},s_{init}}(\Diamond(\text{accumulated weight } < 0)) \leq \vartheta$$

if and only if the given linear recurrence sequence $(u_n)_{n\geq 0}$ is non-negative. The construction can be carried out in time polynomial in k and in the size of the representations of $\alpha_1, \dots, \alpha_k$ and $\beta_0, \dots, \beta_{k-1}$. □

Note that the absolute value of the weights in the constructed MDP are at most k. Hence, the one-counter MDP corresponding to the constructed MDP is only polynomially larger after we replace the transitions with a weight by a sequence of states decreasing or increasing the counter value.

The construction shows that the threshold problem for the *maximal* termination probability of one-counter MDPs is Positivity-hard. Using exactly the same ideas, we can show that the threshold problem for the *minimal* termination probability is Positivity-hard as well. Let us describe the necessary changes in the construction that are also depicted in Figure 5.10. We rename the state *trap* to *trap*′ and add a transition with weight $-k$

to a new absorbing state *trap*. From the states x_j and y_j now state *trap* is reached directly with probability 1 and weight $-j$. Furthermore, the probability to reach x_j when choosing γ_j in t is changed to $\frac{j+1}{k+1} + \beta_j$ and the probability to reach *trap*′ is adjusted accordingly. The analogous change is performed for δ_j. Now, it is easy to check that the optimal choice to minimize the termination probability in state t is to choose γ if the accumulated weight is $\geq k$. In this case the probability of termination is less than $\frac{1}{k+1}$. If the accumulated weight is $0 \leq \ell < k$, the optimal choice is γ_ℓ. The analogous result holds in state s. From then on the proof goes as we have just seen for the maximal termination probability with the change that we have to consider the scheduler $\mathfrak{S}$ always choosing σ in the state *choice* this time. This scheduler is optimal to minimize the termination probability if and only if the given linear recurrence sequence is non-negative. With these adjustments, we conclude:

Corollary 5.11. *The Positivity problem is reducible in polynomial time to the following problem: Given an MDP $\mathcal{M}$ and a rational $\vartheta \in (0,1)$, decide whether*

$$\Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond(\textit{accumulated weight} < 0)) < \vartheta.$$

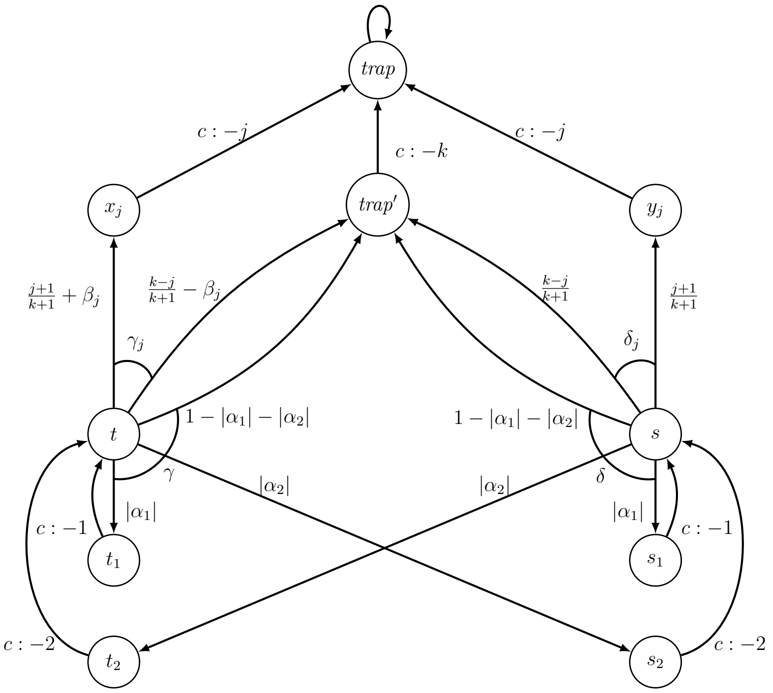


Figure 5.10: Necessary changes to the construction for the result for minimal termination probabilities. The initial component of the MDP is omitted here and stays unchanged.

Energy objectives. As the energy objective $\Box$(accumulated weight ≥ 0) is satisfied if and only if $\Diamond$(accumulated weight < 0) does not hold, the Positivity-hardness of the threshold problem of the optimal satisfaction probability of an energy objective follows easily. As

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\Box(\text{accumulated weight} \geq 0)) = 1 - \Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond(\text{accumulated weight} < 0)),$$

we conclude:

Corollary 5.12. *The Positivity problem is reducible in polynomial time to the following problems: Given an MDP $\mathcal{M}$ and a rational $\vartheta \in (0,1)$, decide whether*

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\Box(\textit{accumulated weight} \geq 0)) > \vartheta$$

and decide whether

$$\Pr^{\min}_{\mathcal{M},s_{init}}(\Box(\textit{accumulated weight} \geq 0)) < \vartheta.$$

Cost problems and quantiles. The proof of the Positivity-hardness of the threshold problem for the termination probability of one-counter MDPs in fact also serves as a proof that cost problems and the computation of quantiles of the accumulated weight before reaching a goal state are Positivity-hard. Observe that in the MDP constructed for Theorem 5.10 and Corollary 5.11, almost all paths ζ under any scheduler satisfy $\Diamond$(accumulated weight < 0) iff they satisfy $\mathrlap{\Diamond}\,{+}\mathit{trap}(\zeta) < 0$ iff their total accumulated weight is less than 0. Thus, we obtain the following corollary:

Corollary 5.13. *The Positivity problem is reducible in polynomial time to the following problems: Given an MDP $\mathcal{M}$ with a designated set of trap states Goal and a rational $\vartheta \in (0,1)$, decide whether*

$$\Pr^{\max}_{\mathcal{M},s_{init}}(\mathrlap{\Diamond}\,{+}\mathit{Goal} < 0) > \vartheta$$

and decide whether

$$\Pr^{\min}_{\mathcal{M},s_{init}}(\mathrlap{\Diamond}\,{+}\mathit{Goal} < 0) < \vartheta.$$

The analogous result also holds for the total accumulated weight.

Termination times of one-counter MDPs. To conclude the section, we show that not only the threshold problems for optimal termination probabilities, but also for the optimal expected termination times in one-counter MDPs that terminate almost surely is Positivity-hard. We again work with weighted MDPs. Let T be the random variable that assigns to each path in a weighted MDP $\mathcal{M}$ the length of the shortest prefix such that the accumulated weight is < 0 after the prefix. To reflect precisely the behavior of a

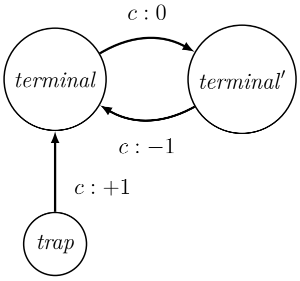


Figure 5.11: Necessary changes to the construction for the result for for maximal expected termination times.

one-counter MDP, we now will work with MDPs where the weight is reduced or increased by at most 1 in each step. We make a small change to the MDP constructed for the proof of Corollary 5.11 that is depicted in Figure 5.10. The initial component (that is not depicted) stays unchanged. For the remaining transitions, all transition reduce the weight or leave it unchanged. The transitions with weight 0 do not occur directly after each other except for the loop at the state *trap* that we adjust in a moment. Hence, we can add additional auxiliary states such that along each path starting from s or t not reaching the state *trap*, the weight is left unchanged and reduced by 1 in an alternating fashion. So, if a path starts in state s or t with accumulated weight w and terminates (i.e. reaches accumulated weight -1) before reaching the state *trap* this takes $2(w+1)$ steps. Now, we replace the loop at the state *trap* by the gadget depicted in Figure 5.11 and let us call the resulting MDP $\mathcal{N}$. So, when reaching *trap* the accumulated weight is increased by 1 before it is reduced in every other step until termination. That means that if a path starting in state s or t with weight w does not terminate before reaching *trap*, the termination time is $2(w+1)+3$ steps.

Now, let $\mathfrak{S}$ be a scheduler and denote the probability not to terminate before reaching *trap* under $\mathfrak{S}$ by $p^{\mathfrak{S}}$. For the expected termination time T in $\mathcal{N}$, we now have

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{N},s_{init}} = \left(\sum_{i=1}^{\infty}(1/2)^i(i+2(i+1))\right) + 3\cdot p^{\mathfrak{S}} = 7 + 3\cdot p^{\mathfrak{S}}.$$

The summands $(1/2)^i(i+2(i+1))$ correspond to the probability to accumulated weight i in the initial component which takes i steps and the $2(i+1)$ steps needed to terminate by alternatingly leaving the weight unchanged and reducing it by 1. The three additional steps after *trap* occur precisely with probability $p^{\mathfrak{S}}$.

Not terminating before *trap* corresponds exactly to not terminating at all in the MDP constructed for Corollary 5.11. The termination probability there is hence $1-p^{\mathfrak{S}}$ for any scheduler. It is hence possible to terminate with a probability less than ϑ in that MDP if and only if it is possible to reach an expected termination time of more than $10-3\vartheta$ in

$\mathcal{N}$. By Corollary 5.11 and the fact that termination is reached almost surely in $\mathcal{N}$ under any scheduler, we hence conclude:

Corollary 5.14. *Let $\mathcal{M}$ be a one-counter MDP with initial state s_{init} that terminates almost surely under any scheduler, let ϑ be a rational, and let T be the random variable assigning the termination time to runs. The Positivity problem is polynomial-time reducible to the problem whether*

$$\mathbb{E}^{\max}_{\mathcal{M},s_{\mathit{init}}}(T) > \vartheta.$$

By similar changes to the MDP used in the proof of Theorem 5.10, we can show the same result for the problem whether $\mathbb{E}^{\min}_{\mathcal{M},s_{\mathit{init}}}(T) < \vartheta$.

5.2.4 Conditional value-at-risk for accumulated weights

Lastly, we aim to prove the Positivity-hardness of the threshold problem for the conditional value-at-risk in this section. Paths with low weight are considered to be the bad outcomes in the sequel denoted by the $\downarrow$ in the index. The main result of the section is the following:

Theorem 5.15. *The Positivity problem is polynomial-time reducible to the following problem: Given an MDP $\mathcal{M}$ and rationals ϑ and $p \in (0,1)$, decide whether*

$$\mathit{CVaR}^{\max}_{\downarrow,p}(\oplus \mathit{goal}) > \vartheta.$$

We will use an auxiliary optimization problem to prove this result. We begin with the following consideration: Given an MDP $\mathcal{M}$ with initial state s_{init}, we construct a new MDP $\mathcal{N}$. We add a new initial state s'_{init}. In s'_{init}, there is only one action with weight 0 enabled leading to s_{init} with probability $\frac{1}{3}$ and to *goal* with probability $\frac{2}{3}$. So, at least two thirds of the paths accumulate weight 0 before reaching the goal. Hence, we can already say that $\mathit{VaR}^{\mathfrak{S}}_{1/2}(\oplus \mathit{goal}) = 0$ in $\mathcal{N}$ under any scheduler $\mathfrak{S}$. Note that schedulers for $\mathcal{M}$ can be seen as schedulers for $\mathcal{N}$ and vice versa. This considerably simplifies the computation of the conditional value-at-risk in $\mathcal{N}$. Define the random variable $\ominus \mathit{goal}(\zeta)$ to be $\oplus \mathit{goal}(\zeta)$ if $\oplus \mathit{goal} \leq 0$ and to be 0 otherwise. Now, the conditional value-at-risk for the probability value 1/2 under a scheduler $\mathfrak{S}$ in $\mathcal{N}$ is given by $\mathit{CVaR}^{\mathfrak{S}}_{1/2}(\oplus \mathit{goal}) = 2 \cdot \mathbb{E}^{\mathfrak{S}}_{\mathcal{N},s_{\mathit{init}}}(\ominus \mathit{goal}) = \frac{2}{3} \cdot \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}(\ominus \mathit{goal})$. So, the result follows from the following lemma:

Lemma 5.16. *The Positivity problem is polynomial-time reducible to the following problem: Given an MDP $\mathcal{M}$ and a rational ϑ, decide whether $\mathbb{E}^{\max}_{\mathcal{M},s_{\mathit{init}}}(\ominus \mathit{goal}) > \vartheta$.*

Proof. The first important observation is that the optimal expectation $e(q,w)$ of $\ominus \mathit{goal}$ for different starting states q and starting weights w satisfies Equation $(*)$ from Section 5.1, i.e., $e(q,w) = \sum_{r\in S} P(q,\alpha,r)\cdot e(r, w{+}wgt(q,\alpha))$ if an optimal scheduler chooses actions α in state $q \neq goal$ when the accumulated weight is w. The value $e(goal, w)$ is w if $w \leq 0$ and 0 otherwise. This allows us to reuse the gadget $\mathcal{G}_{\bar{\alpha}}$ to encode a linear recurrence relation.

We again adjust the gadget encoding the initial values of a linear recurrence sequence. So, let k be a natural number, $\alpha_1,\dots,\alpha_k$ be rational coefficients of a linear recurrence sequence, and $\beta_0,\dots,\beta_{k-1} \geq 0$ the rational initial values. W.l.o.g. we again assume these values to be small, namely: $\sum_{1\leq i\leq k} |\alpha_i| \leq \frac{1}{5(k+1)}$ and for all j, $\beta_j \leq \frac{1}{3}\alpha$ where $\alpha = \sum_{1\leq i\leq k} |\alpha_i|$.

The new gadget that encodes the initial values of a linear recurrence sequence is depicted in Figure 5.12. In states t and s, there is a choice between actions γ_j and δ_j, respectively, for $0 \leq j \leq k-1$. After glueing together this gadget with the gadget $\mathcal{G}_{\bar{\alpha}}$ at states t, s, and *goal*, we prove that the interplay between the gadgets is correct: Let $0 \leq j \leq k-1$. Starting with accumulated weight $-k{+}j$ in state t, the action γ_j maximizes the partial expectation among the actions $\gamma_0,\dots,\gamma_{k-1}$. Likewise, δ_j is optimal when starting in s with weight $-k{+}j$. If the accumulated weight is non-negative in state s or t, then γ or δ are optimal. The idea is that for positive starting weights, the tail loss of γ_i and δ_i is relatively high while for weights just below 0, the chance to reach *goal* with positive weight again outweighs this tail loss.

First, we estimate the expectation of $\oplus goal$ when choosing δ_i and δ while the accumulated weight is $-k{+}j$ in s. If $i > j$, then δ_i and δ lead to *goal* directly with probability $1{-}\alpha$ and weight ≤ -1. So, the expectation is less than $-(1-\alpha) \leq -1{+}\frac{1}{5(k+1)}$.

If $i \leq j$, then with probability $1{-}\alpha$ *goal* is reached with positive weight, hence $\oplus goal$ is 0 on these paths. With probability β_i, goal is reached via y_j'. In this case all runs reach *goal* with negative weight. On the way to y_j' weight $2k$ is added, but afterwards subtracted again at least once. In expectation weight $2k$ is subtracted $\frac{k+1}{k}$ many times. Furthermore, $-2k{+}i$ is added to the starting weight of $-k{+}j$. So, these paths contribute $\beta_i\cdot(2k{-}2k\frac{k+1}{k}{-}3k{+}j{+}i) = (-3k{+}j{+}i{-}2)\cdot\beta_i$ to the expectation of $\oplus goal$. With analogous reasoning, we see that the remaining paths contribute $(-3k{+}j{+}i{-}1)\cdot(\alpha-\beta_i)$. So, all in all the expectation of $\oplus goal$ in this situation is $\alpha\cdot(-3k{+}j{+}i{-}1){-}\beta_i$. Now, as $\alpha \leq \frac{1}{5(k+1)}$ and $\beta_i \leq \frac{\alpha}{3}$ for all i, we see that $\alpha\cdot(-3k{+}j{+}i{-}1){-}\beta_i \geq -(3k+2)\alpha \geq -1{+}\frac{1}{5(k+1)}$. The optimum with $i \leq j$ is obtained for $i = j$ as $\beta_i \leq \alpha/3$ for all i. Hence indeed δ_j is the optimal action. For γ_j the same proof with $\beta_i = 0$ for all i leads to the same result.

Now assume that the accumulated weight in t or s is $\ell \geq 0$. Then, all actions lead to *goal* with a positive weight with probability $1-\alpha$. In this case $\oplus goal$ is 0. However, a scheduler $\mathfrak{S}$ which always chooses γ and δ is better than a scheduler choosing γ_j or δ_j for any $j \leq k{-}1$. Under scheduler $\mathfrak{S}$ starting from s or t a run returns to $\{s,t\}$ with probability α while accumulating weight $\geq -k$ and the process is repeated. After choosing γ_j or δ_j the run moves to x_j, y_j or y_j' while accumulating a negative weight.

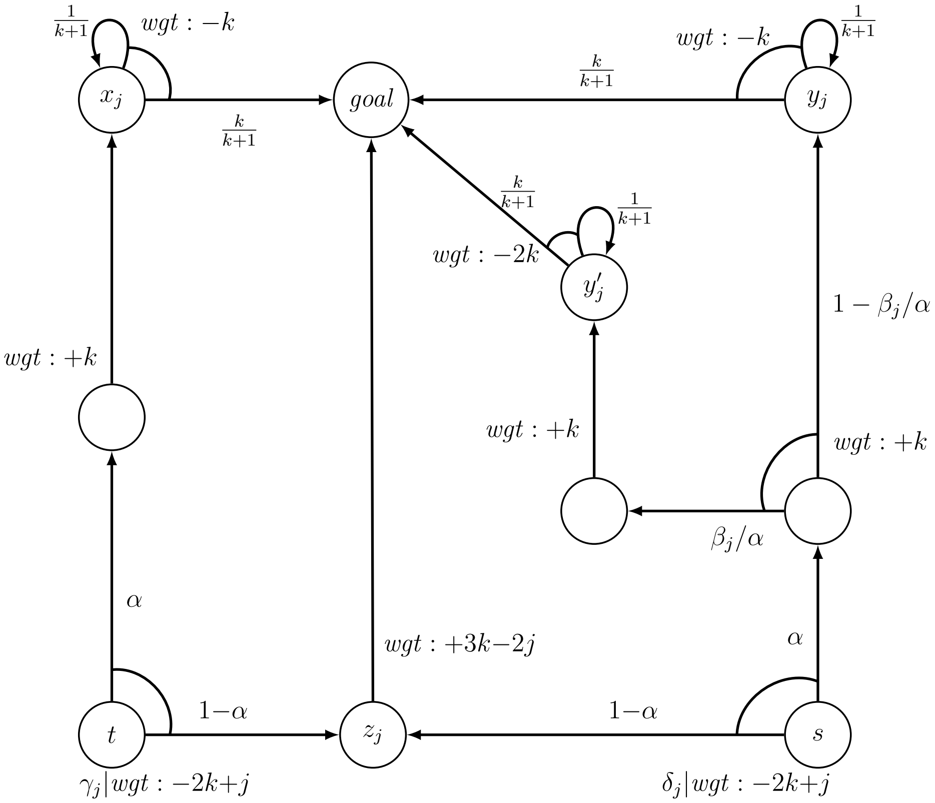


Figure 5.12: The gadget encoding initial values for the reduction to the threshold problem for the conditional value-at-risk. The gadget contains the depicted states and actions for each $0 \leq j \leq k-1$. The probability α is $\sum_{1\leq i\leq k} |\alpha_i|$.

From then on, in each step it will stay in that state with probability greater than α and accumulate weight $\leq -k$. Hence, the expectation of $\oplus goal$ is lower under γ_j or δ_j than under $\mathfrak{S}$. Therefore indeed γ and δ are the best actions for non-negative accumulated weight in states s and t.

Let now $e(t,w)$ and $e(s,w)$ denote the optimal expectations of $\oplus goal$ when starting in t or s with weight w. Further, let $d(w) = e(t,w) - e(s,w)$. From the argument above, we also learn that the difference $d(-k+j)$ is equal to β_j, for $0 \leq j \leq k-1$. Put together with the linear recurrence encoded in $\mathcal{G}_{\bar{\alpha}}$ this shows that $d(-k+w) = u_w$ for all w where $(u_n)_{n\in\mathbb{N}}$ is the linear recurrence sequence specified by the α_i, β_j, $1 \leq i \leq k$, and $0 \leq j \leq k-1$.

Finally, we add the same initial component as in the previous section to obtain an MDP $\mathcal{M}$. Let $\mathfrak{S}$ be the scheduler always choosing τ in state c and afterwards following the optimal actions as described above is optimal iff the linear recurrence sequence stays non-negative. The remaining argument goes completely analogously to the proof of Theorem 5.10. Grouping together the optimal values in vectors v_n with $2k$ entries as done there, we can use the same Markov chain as in that proof to obtain a matrix A such that

$v_{n+1} = Av_n$. This allows us to compute the rational value $\vartheta = \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\diamondminus goal)$ via a matrix series in polynomial time and $\mathbb{E}^{\max}_{\mathcal{M},s_{init}}(\diamondminus goal) > \vartheta$ if and only if the given linear recurrence sequence is eventually negative. □

By the discussion above, this lemma directly implies Theorem 5.15. With adaptions similar to the previous section, it is possible to obtain the analogous result for the minimal expectation of $\diamondminus goal$. This implies that also the threshold problem whether the minimal conditional value-at-risk is less than a threshold ϑ, $CVaR^{\min}_{\downarrow,p}(\diamondplus goal) < \vartheta$, is Positivity-hard.

CHAPTER

SIX

APPROXIMATION ALGORITHMS

The Positivity-hardness results of the previous chapter show that we cannot expect to be able to compute the optimal values in the non-classical stochastic shortest path problems we investigated with known techniques. Therefore, we now turn our attention to the approximability of these values. For practical purposes, efficient approximation algorithms are usually sufficient for the development of useful tools.

But also from a theoretical point of view, the approximability results we present here are interesting. After proving Positivity-hardness, the question whether the two non-classical stochastic shortest path problems are undecidable remains open. The arguably most famous undecidability result in a similar setting is the undecidability of the emptiness problem for probabilistic finite automata (PFA) [Paz71]. In a PFA, the successor state in each step is chosen according to a probability distribution that depends on the current state and the current letter of an input word. The emptiness problem asks whether there is a finite input word w such that the probability to be in an accepting state of the PFA after reading the input word w is at least ϑ for some given threshold ϑ. In the terminology of MDPs, input words can be regarded as deterministic schedulers that cannot take any information on the current state or the history of a run into account, but have to schedule a sequence of actions without further information. From this perspective, PFAs can be seen to be *unobservable* MDPs, sometimes called *blind partially observable* MDPs (see, e.g., [PT87, MHC99, CCT16]). This emptiness problem for PFAs has been shown to be undecidable by Paz [Paz71]. Furthermore, Condon and Lipton [CL89] showed that it is impossible to approximate the value of a PFA, i.e., the supremum over the probabilities with which finite words are accepted. In this chapter, we will prove that the optimal partial and conditional expectations can be approximated. This is a strong indication that the problems are fundamentally different from the emptiness problem for PFAs. More precisely, the goal of this section is the proof of the following theorem.

Theorem 6.1. *Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$ and let $\varepsilon > 0$. The maximal partial expectation $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ can be approximated up to an absolute error of ε in time exponential in the size of $\mathcal{M}$ and polynomial in* $\log(1/\varepsilon)$.

If further $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$, also $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$ can be approximated up to an absolute error of ε in time exponential in the size of $\mathcal{M}$ and polynomial in $\log(1/\varepsilon)$.

Afterwards, we prove a hardness result for approximations. Namely, we show that there are no polynomial-time approximation algorithms if P $\neq$ PSPACE.

Outline. We first show how we can estimate bounds on the growth of the accumulated weights in an MDP (Section 6.1). Then, we show that there are ε-optimal schedulers for the partial expectation which become memoryless as soon as the accumulated weight leaves a sufficiently large weight window around 0. This result uses a weak form of saturation points. This insight can be used to compute approximations of the optimal partial expectation (Section 6.2). The result can be extended to conditional expectations via an approximate binary search (Section 6.3). We conclude by presenting the hardness result (Section 6.4).

Related Work. For some of the problems that we have shown to be Positivity-hard in the previous chapter, approximation algorithms have been provided in the literature: The optimal termination probability of one-counter MDPs is shown to be approximable in [BBEK11] while an analogous result for the expected termination time of almost surely terminating one-counter MDPs is shown in [BKNW12]. Conceptually, these approximation algorithms are similar to our approximation algorithms as they estimate counter values from which on ϵ-optimal schedulers can behave memorylessly.

In MDPs with an energy objective, it has been shown that the maximal expected mean payoff among schedulers satisfying the energy objective with probability 1 can be approximated [BKN16]. For cost problems, [HKL17] proves that the probability with which the accumulated cost satisfies a Boolean combination of inequality constraints when entering a target state can be approximated in Markov chains.

Note on the publication of the results. The approximation algorithms for partial and conditional expectations have been presented in [PB19]. Here, we additionally provide the hardness result (Section 6.4).

6.1 Bounding the growth of weights

In this section, we will start by providing estimations to be able to bound the possible growth of weights in an MDP. As a result, we are able to provide upper bounds on the optimal partial and conditional expectation that we need in the subsequent sections. Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$. We have seen in Chapter 3 that finiteness of $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ can be checked in polynomial time. After a successful check of finiteness and the usual pre-processing, we can assume that all end components have negative maximal expected mean payoff and that *Goal* consists of an absorbing state *goal* that can be reached from all states except for one absorbing state *fail*. Let δ be the minimal non-zero transition probability in $\mathcal{M}$ and $W := \max_{s\in S,\alpha\in Act(s)} |wgt(s,\alpha)|$. To compute an upper bound on the partial expectation in $\mathcal{M}$, we are going to compute a weight value $c_{\mathcal{M}}$ and a probability $\lambda_{\mathcal{M}}$ such that the probability to accumulated weight above $c_{\mathcal{M}}$ from any state in $\mathcal{M}$ is at most $\lambda_{\mathcal{M}}$. Considering the MEC-quotient $MEC(\mathcal{M})$, there are two ways to accumulate weight: Weights can be accumulated while taking transitions in the MEC-quotient moving between the MECs and weights can be accumulated by moving around inside a MEC. For the first way to accumulate weight, we can make an easy estimation: Moving through the MEC-quotient, the probability to reach an accumulated weight of $|S|\cdot W$ is bounded by $1-\delta^{|S|}$ as *goal* or *fail* is reached within S steps with probability at least $1-\delta^{|S|}$. It remains to show similar bounds inside an end component.

We will use the characterization of the maximal expected mean payoff in terms of super-harmonic vectors due to Hordijk and Kallenberg [HK79] to define a super-martingale controlling the growth of the accumulated weight in an end component under any scheduler. We will, however, not dive into the definition of super-martingales here because the arguments in the sequel can be carried out without the definition. For the reader familiar with the notion, we will briefly point out the super-martingale. As the value vector for the maximal mean payoff in an end component is constant and negative in our case, the results of [HK79] that were also briefly discussed in Chapter 2 yield:

Proposition 6.2 (Hordijk, Kallenberg [HK79])**.** *Let $\mathcal{E}=(S,Act)$ be an end component of $\mathcal{M}$ with maximal mean payoff $-t$ for some $t>0$. Then there is a vector $(u_s)_{s\in S}$ such that $-t+u_s\geq wgt(s,\alpha)+\sum_{s'\in S}P(s,\alpha,s')\cdot u_{s'}$ for all $s\in S$ and $\alpha\in Act(s)$.*

Furthermore, let v be the vector (-t,…,-t) in $\mathbb{R}^S$. Then, (v,u) is the solution to a linear program with $2|S|$ variables, $2|S||Act|$ inequalities, and coefficients formed from the transition probabilities and weights in $\mathcal{E}$.

We will call the vector u a *super-potential* because the expected accumulated weight after i steps is at most $u_s-\min_{t\in S}u_t-i\cdot t$ when starting in state s.

Let now $\mathfrak{S}$ be a scheduler for $\mathcal{E}$ starting in some state s. We define the following random variables on $\mathfrak{S}$-runs in $\mathcal{E}$: let $s(i)\in S$ be the state after i steps, let $\alpha(i)$ be the action chosen after i steps, let $w(i)$ be the accumulated weight after i steps, and let $\pi(i)$ be the history, i.e. the finite path after i steps. To keep the notation simple, we do not include the scheduler $\mathfrak{S}$ to the notation although these random variables of course depend on $\mathfrak{S}$.

Lemma 6.3. *The sequence $m(i)\overset{def}{=}w(i)+u_{s(i)}$ satisfies*

$$\mathbb{E}(m(i+1)|\pi(0),\dots,\pi(i))\leq m(i)-t$$

for all i.

Proof. By Proposition 6.2, $\mathbb{E}(m(i+1)|\pi(0),\dots,\pi(i)) - m(i) = wgt(s(i),\mathfrak{S}(\pi(i))) + \sum_{s'\in S} P(s(i),\mathfrak{S}(\pi(i)),s')\cdot u_{s'} - u_{s(i)} \leq -t$. □

The statement of the lemma means that $m(i)+i\cdot t$ is a super-martingale with respect to the histories $\pi(i)$. We can now estimate the growth of the weight inside an end component by applying the following theorem by Blackwell [Bla54].

Theorem 6.4 (Blackwell [Bla54]). *Let $X_1, X_2, \dots$ be random variables and define the random variable $S_n \overset{def}{=} \sum_{k=1}^{n} X_k$. Assume that $|X_i| \leq 1$ for all i and that there is a $u > 0$ such that $\mathbb{E}(X_{n+1}|X_1,\dots,X_n) \leq -u$. Then,*

$$\Pr(\sup_{n\in\mathbb{N}} S_n \geq t) \leq \left(\frac{1-u}{1+u}\right)^t .$$

We denote $\max_{s'\in S} u_{s'} - \min_{s'\in S} u_{s'}$ by $\|u\|$. Observe that

$$|m(i+1) - m(i)| \leq \|u\| + W =: c_{\mathcal{E}}.$$

We can rescale the sequence $m(i)$ by defining $m'(i) \overset{\text{def}}{=} (m(i) - m(0))/c_{\mathcal{E}}$. This ensures that $m'(0) = 0$, $|m'(i+1) - m'(i)| \leq 1$ and $\mathbb{E}(m'(i+1)|m'(0),\dots,m'(i)) \leq -t/c_{\mathcal{E}}$ for all i. In this way, we arrive at the following conclusion, putting $\lambda_{\mathcal{E}} := \frac{1-t/c_{\mathcal{E}}}{1+t/c_{\mathcal{E}}}$.

Corollary 6.5. *For any scheduler $\mathfrak{S}$ and any starting state s in $\mathcal{E}$, we have*

$$\Pr_s^{\mathfrak{S}}(\Diamond wgt \geq (k{+}1)\cdot c_{\mathcal{E}}) \leq \lambda_{\mathcal{E}}^k.$$

Proof. By Theorem 6.4,

$$\begin{aligned}\Pr_s^{\mathfrak{S}}(\Diamond wgt \geq (k+1)\cdot c_{\mathcal{E}}) &\leq \Pr_s^{\mathfrak{S}}(\Diamond wgt \geq \|u\| + k\cdot c_{\mathcal{E}})\\ &\leq \Pr_s^{\mathfrak{S}}(\exists i : m(i) - m(0) \geq k\cdot c_{\mathcal{E}})\\ &\leq \Pr_s^{\mathfrak{S}}(\sup_{i\in\mathbb{N}} m'(i) \geq k)\\ &\leq \left(\frac{1-t/c_{\mathcal{E}}}{1+t/c_{\mathcal{E}}}\right)^k .\end{aligned}$$

□

Let *MEC* be the set of maximal end components in $\mathcal{M}$. For each $\mathcal{E} \in MEC$, let $\lambda_{\mathcal{E}}$ and $c_{\mathcal{E}}$ be as in Corollary 6.5. Define

$$\lambda_{\mathcal{M}} \overset{\text{def}}{=} 1 - (\delta^{|S|}\cdot \prod_{\mathcal{E}\in MEC} (1-\lambda_{\mathcal{E}})),$$

and

$$c_{\mathcal{M}} \overset{\text{def}}{=} |S| \cdot W + \sum_{\mathcal{E} \in MEC} c_{\mathcal{E}}.$$

Then an accumulated weight of $c_{\mathcal{M}}$ cannot be reached with a probability greater than $\lambda_{\mathcal{M}}$ because reaching accumulated weight $c_{\mathcal{M}}$ would require reaching weight $c_{\mathcal{E}}$ in some end component $\mathcal{E}$ or reaching weight $|S| \cdot W$ in the MEC-quotient and $1 - \lambda_{\mathcal{M}}$ is a lower bound on the probability that none of this happens (under any scheduler).

After bounding the growth of the accumulated weight in $\mathcal{M}$, we are now in the position to provide upper bounds on the partial and conditional expectation.

Proposition 6.6. *Let $\mathcal{M}$ be as above. There is an upper bound $\mathbb{PE}^{ub}$ for the partial expectation in $\mathcal{M}$ computable in polynomial time.*

Proof. In any end component $\mathcal{E}$, the maximal mean payoff $-t$ and the super-potential u are computable in polynomial time. Hence, $c_{\mathcal{E}}$ and $\lambda_{\mathcal{E}}$, and in turn also $c_{\mathcal{M}}$ and $\lambda_{\mathcal{M}}$ are also computable in polynomial time. When we reach accumulated weight $c_{\mathcal{M}}$ for the first time, the actual accumulated weight is at most $c_{\mathcal{M}} + W$. So, we conclude that $\Pr^{\max}_{\mathcal{M},s}(\Diamond wgt \geq k \cdot (c_{\mathcal{M}} + W)) \leq \lambda_{\mathcal{M}}^k$ for all $s \in S$. The partial expectation can now be bounded by $\sum_{k=0}^{\infty}(k+1) \cdot (c_{\mathcal{M}} + W) \cdot \lambda_{\mathcal{M}}^k = \frac{c_{\mathcal{M}}+W}{(1-\lambda_{\mathcal{M}})^2}$. □

Corollary 6.7. *Let $\mathcal{M}$ be as before and assume that $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$. There is an upper bound $\mathbb{CE}^{ub}$ for the conditional expectation in $\mathcal{M}$ computable in polynomial time.*

Proof. By the pre-processing described in Chapter 3, we can construct an MDP $\mathcal{N}$ in which *goal* is reached with positive probability under any scheduler in polynomial time with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} = \mathbb{CE}^{\max}_{\mathcal{N},s_{init}}$. As $q = \Pr^{\min}_{\mathcal{N},s_{init}}(\Diamond goal)$ is computable in polynomial time, the bound $\mathbb{CE}^{ub} \overset{\text{def}}{=} \mathbb{PE}^{ub}/q$ is an upper bound for the conditional expectation in $\mathcal{M}$ computable in polynomial time. □

These upper bounds serve as an ingredient for the computation of a weak form of "saturation point" for MDPs with integer weights in the next section.

6.2 Approximating optimal partial expectations

Recall that there are no saturation points in MDPs with integer weights that provide a bound on the accumulated weight above which optimal schedulers can switch to memoryless behavior (see Section 3.2.3). Nevertheless, we can compute a bound below which optimal schedulers can only choose actions that make it possible to reach *goal* with the minimal possible probability. This will allow us to approximate the maximal partial expectation by finite-memory schedulers that switch to memoryless behavior once a sufficiently large weight window around 0 is left.

Let $\mathcal{M}$ be as above with $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} < \infty$ and let $\mathbb{PE}^{ub}$ be the upper bound on the partial expectation that we just computed. For each state $s \in S$, we define $p^{\max}_s = \Pr^{\max}_{\mathcal{M},s}(\Diamond goal)$ and $p^{\min}_s = \Pr^{\min}_{\mathcal{M},s}(\Diamond goal)$. Further, for opt $\in \{\max, \min\}$ and each state-action pair (s,α), let $p^{\text{opt}}_{s,\alpha} = \sum_{t\in S} P(s,\alpha,t) \cdot p^{\text{opt}}_t$ and let $Act^{\text{opt}}(s) = \{\alpha \in Act(s) | p^{\text{opt}}_{s,\alpha} = p^{\text{opt}}_s\}$.

As we have also seen in Chapter 3, we can compute a memoryless deterministic schedulers $\mathfrak{Max}$ in polynomial time that maximizes the partial expectation among all schedulers reaching *goal* with maximal probability. To compute this scheduler, we can solve a classical stochastic shortest path problem in the MDP $\mathcal{M}^{\max}$ in which only actions in $Act^{\max}$ are enabled. We scale down the weight of a state-action pair (s,α) to $wgt(s,\alpha) \cdot p^{\max}_s$ and maximize the accumulated weight before reaching *goal* or *fail* according to this weight function. Likewise, we can compute a memoryless scheduler $\mathfrak{Min}$ maximizing the partial expectation among all scheduler reaching *goal* with minimal probability.

The idea for the approximation is to use the approximate value $\mathbb{PE}^{\mathfrak{Max}}_{\mathcal{M},s} + w \cdot p^{\max}_s$ for the maximal partial expectation if a high weight w has been accumulated in state s. Similarly, for small weights w', we use the value $\mathbb{PE}^{\mathfrak{Min}}_{\mathcal{M},s} + w \cdot p^{\min}_s$. We will first provide a lower "saturation point" making sure that only actions minimizing the probability to reach the goal are used by an optimal scheduler as soon as the accumulated weight drops below this saturation point. The argument is similar to the proof of the existence of a saturation point in the setting with non-negative weights as in Proposition 3.26. In contrast to the setting with non-negative weights, it is not the case that below the "saturation point" there is a fixed memoryless scheduler providing the optimal decisions – hence the quotation marks.

Proposition 6.8. *Let $\mathcal{M}$ be as above. Let $s \in S$ and let*

$$\mathfrak{q}_s \overset{def}{=} \frac{\mathbb{PE}^{ub} - \mathbb{PE}^{\mathfrak{Min}}_s}{p^{\min}_s - \min\limits_{\alpha \notin Act^{\min}(s)} p^{\min}_{s,\alpha}}.$$

Then any weight-based deterministic scheduler $\mathfrak{S}$ maximizing the partial expectation in $\mathcal{M}$ satisfies $\mathfrak{S}(s,w) \in Act^{\min}(s)$ if $w \leq \mathfrak{q}_s$.

Proof. Suppose a weight-based deterministic scheduler $\mathfrak{S}$ chooses an action $\alpha \notin Act^{\min}(s)$ at state s when the accumulated weight is $w < \mathfrak{q}_s$. The partial expectation from s on starting with weight w is then bounded from above by

$$\mathbb{PE}^{ub} + w \cdot p^{\min}_{s,\alpha}.$$

The scheduler $\mathfrak{Min}$ achieves a partial expectation from this situation of

$$\mathbb{PE}^{\mathfrak{Min}}_s + w \cdot p^{\min}_s.$$

The value $\mathfrak{q}_s$ is chosen such that it follows that $\mathfrak{Min}$ achieves a higher partial expectation from this situation on for $w < \mathfrak{q}_s$. For an optimal scheduler, we can hence indeed assume that it only chooses actions from $Act^{\min}(s)$ for weights below $\mathfrak{q}_s$. □

Let $\mathfrak{q} \overset{\text{def}}{=} \min_{s\in S} \mathfrak{q}_s$ and let $D \overset{\text{def}}{=} \mathbb{PE}^{ub} - \min\{\mathbb{PE}_s^{\mathfrak{Max}}, \mathbb{PE}_s^{\mathfrak{Min}} | s \in S\}$. Given $\varepsilon > 0$, we define $R_\varepsilon^+ \overset{\text{def}}{=} (c_{\mathcal{M}} + W) \cdot \left\lceil \frac{\log(2D)+\log(1/\varepsilon)}{\log(1/\lambda_{\mathcal{M}})} \right\rceil$ and $R_\varepsilon^- \overset{\text{def}}{=} \mathfrak{q} - R_\varepsilon^+$.

Theorem 6.9. *With all notation as above, there is a weight-based deterministic scheduler $\mathfrak{S}$ such that the scheduler $\mathfrak{T}$ defined by*

$$\mathfrak{T}(\pi) = \begin{cases} \mathfrak{S}(\pi) & \textit{if any prefix } \pi' \textit{ of } \pi \textit{ satisfies } R_\varepsilon^- \leq wgt(\pi') \leq R_\varepsilon^+, \\ \mathfrak{Max}(\pi) & \textit{if the shortest prefix } \pi' \textit{ of } \pi \textit{ with } wgt(\pi') \notin [R_\varepsilon^-, R_\varepsilon^+] \\ & \textit{satisfies } wgt(\pi') > R_\varepsilon^+, \\ \mathfrak{Min}(\pi) & \textit{otherwise,} \end{cases}$$

satisfies $\mathbb{PE}_{s_{init}}^{\mathfrak{T}} \geq \mathbb{PE}_{s_{init}}^{\max} - \varepsilon$.

Proof. Let $\mathfrak{S}$ be a weight-based deterministic scheduler with $\mathbb{PE}_{s_{init}}^{\mathfrak{S}} = \mathbb{PE}_{s_{init}}^{\max}$. Define

$$\mathfrak{T}(\pi) = \begin{cases} \mathfrak{S}(\pi) & \text{, if any prefix } \pi' \text{ of } \pi \text{ satisfies } R_\varepsilon^- \leq wgt(\pi) \leq R_\varepsilon^+, \\ \mathfrak{Max}(\pi) & \text{, if the shortest prefix } \pi' \text{ of } \pi \text{ with } wgt(\pi') \notin [R_\varepsilon^-, R_\varepsilon^+] \\ & \text{satisfies } wgt(\pi') > R_\varepsilon^+, \\ \mathfrak{Min}(\pi) & \text{, otherwise.} \end{cases}$$

We give an estimation for the difference $\mathbb{PE}_{s_{init}}^{\max} - \mathbb{PE}_{s_{init}}^{\mathfrak{T}}$. In order to do so, we define the following two sets:

$$\begin{aligned} \Pi_\varepsilon^+ &:= \{\pi \text{ finite } \mathfrak{S}\text{-path } | wgt(\pi) \geq R_\varepsilon^+ \\ &\qquad \text{and for any proper prefix } \pi' \text{ of } \pi,\ R_\varepsilon^- \leq wgt(\pi') \leq R_\varepsilon^+\}, \\ \Pi_\varepsilon^- &:= \{\pi \text{ finite } \mathfrak{S}\text{-path } | wgt(\pi) \leq R_\varepsilon^- \\ &\qquad \text{and for any proper prefix } \pi' \text{ of } \pi,\ R_\varepsilon^- \leq wgt(\pi') \leq R_\varepsilon^+\}. \end{aligned}$$

Denote by $\mathbb{PE}_s^{\max}[w]$ the maximal partial expectation when starting in state s with weight w. The schedulers $\mathfrak{S}$ and $\mathfrak{T}$ agree on all paths not in Π_ε^+ or Π_ε^-. Hence,

$$\begin{aligned} & \mathbb{PE}_{s_{init}}^{\max} - \mathbb{PE}_{s_{init}}^{\mathfrak{T}} \\ = & \sum_{\pi\in\Pi_\varepsilon^+} \Pr_{s_{init}}^{\mathfrak{S}}(\pi) \cdot (\mathbb{PE}_{last(\pi)}^{\max}[wgt(\pi)] - \mathbb{PE}_{last(\pi)}^{\mathfrak{Max}} - p_{last(\pi)}^{\max} \cdot wgt(\pi)) + \\ & \sum_{\pi\in\Pi_\varepsilon^-} \Pr_{s_{init}}^{\mathfrak{S}}(\pi) \cdot (\mathbb{PE}_{last(\pi)}^{\max}[wgt(\pi)] - \mathbb{PE}_{last(\pi)}^{\mathfrak{Min}} - p_{last(\pi)}^{\min} \cdot wgt(\pi)). \end{aligned}$$

For the first sum, we have the following estimation:

$$\begin{aligned} & \sum_{\pi\in\Pi_\varepsilon^+} \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\pi)\cdot(\mathbb{PE}^{\max}_{last(\pi)}[wgt(\pi)] - \mathbb{PE}^{\mathfrak{Max}}_{last(\pi)} - p^{\max}_{last(\pi)}\cdot wgt(\pi)) \\ \leq\ & \sum_{\pi\in\Pi_\varepsilon^+} \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\pi)\cdot(\mathbb{PE}^{\max}_{last(\pi)} - \mathbb{PE}^{\mathfrak{Max}}_{last(\pi)}) \\ \leq\ & \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\Pi_\varepsilon^+)\cdot D \leq \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\Diamond wgt \geq R_\varepsilon^+)\cdot D \\ \leq\ & \lambda_{\mathcal{M}}^{\frac{\log(2D)+\log(1/\varepsilon)}{\log(1/\lambda_{\mathcal{M}})}}\cdot D = 2^{\log(\lambda_{\mathcal{M}})\cdot\frac{\log(2D)+\log(1/\varepsilon)}{\log(1/\lambda_{\mathcal{M}})}}\cdot D = 2^{\log(\varepsilon)-\log(2D)}\cdot D = \varepsilon/2. \end{aligned}$$

For the second sum, consider the following scheduler. On extensions of paths in Π_ε^-, let $\mathfrak{S}'$ be the scheduler which behaves like $\mathfrak{S}$ until the accumulated weight is at least $\mathfrak{q}$ again and then switches to the choices of $\mathfrak{Min}$. We know that $\mathfrak{S}$ only chooses actions in $Act^{\min}(s)$ when in a state s with accumulated weight below $\mathfrak{q}$. On the other hand, $\mathfrak{Min}$ is optimal among these schedulers. So, $\mathfrak{Min}$ is at least as good as $\mathfrak{S}'$ on extensions of paths in Π_ε^- with respect to maximizing the partial expectation. Further, starting at a path in Π_ε^- we reach an accumulated weight of at least $\mathfrak{q}$ only if we accumulate a weight of at least R_ε^+. Afterwards, we can bound the advantage of $\mathfrak{S}$ over $\mathfrak{Min}$ by D. So, we get the following estimation:

$$\begin{aligned} & \sum_{\pi\in\Pi-_\varepsilon} \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\pi)\cdot(\mathbb{PE}^{\max}_{last(\pi)}[wgt(\pi)] - \mathbb{PE}^{\mathfrak{Min}}_{last(\pi)} - p^{\min}_{last(\pi)}\cdot wgt(\pi)) \\ \leq\ & \sum_{\pi\in\Pi_\varepsilon^-} \mathrm{Pr}^{\mathfrak{S}}_{s_{init}}(\pi)\cdot(\mathrm{Pr}^{\max}_{last(\pi)}(\Diamond wgt \geq R_\varepsilon^+)\cdot D) \leq \varepsilon/2. \end{aligned}$$

So, $\mathbb{PE}^{\max}_{s_{init}} - \mathbb{PE}^{\mathfrak{T}}_{s_{init}} \leq \varepsilon$. □

This result now allows us to compute an ε-approximation and an ε-optimal scheduler with finite memory by linear programming, similar to the case of non-negative weights, in a linear program with $R_\varepsilon^+ + R_\varepsilon^-$ many variables and $|Act|$-times as many inequalities.

Theorem 6.10. $\mathbb{PE}^{\max}_{s_{init}}$ *can be approximated up to an absolute error of* ε *in time exponential in the size of* $\mathcal{M}$ *and polynomial in* $\log(1/\varepsilon)$.

Proof. We have seen that R_ε^- and R_ε^+ can be computed in time polynomial in the size of $\mathcal{M}$. Their numeric values are hence at most exponential in the size of $\mathcal{M}$. Furthermore, these numeric values are linear in $\log(1/\varepsilon)$. Consider the following linear program with one variable $x_{s,w}$ for each $s\in S$ and $R_\varepsilon^- - W \leq w \leq R_\varepsilon^+ + W$:

Minimize $\sum_{s,w} x_{s,w}$ under the following constraints:

$$x_{goal,w} = w, \text{ and } x_{fail,w} = 0,$$

for $w \geq R_\varepsilon^+$ and $s \in S \setminus \{goal, fail\}$,

$$x_{s,w} = \mathbb{PE}_s^{\mathfrak{Max}} + p_s^{\max} \cdot w,$$

for $w \leq R_\varepsilon^-$ and $s \in S \setminus \{goal, fail\}$,

$$x_{s,w} = \mathbb{PE}_s^{\mathfrak{Min}} + p_s^{\min} \cdot w,$$

and for $R_\varepsilon^- < w < R_\varepsilon^+$, $s \in S \setminus \{goal, fail\}$, and $\alpha \in Act(s)$,

$$x_{s,w} \geq \sum_{t \in S} P(s, \alpha, t) \cdot x_{t,w+wgt(s,\alpha)}.$$

We can interpret the linear program as a linear program for weighted reachability on an MDP with state space $S \times \{R_\varepsilon^- - W, \ldots, R_\varepsilon^+\}$ and the transitions induced by $\mathcal{M}$. This MDP now has no end components. Hence, the linear program has a unique solution. This solution corresponds to the optimal value among the schedulers of the form of $\mathfrak{T}$ in the previous theorem. □

6.3 Transfer to conditional expectations

To approximate the optimal conditional expectation, we will use the reduction from the threshold problem for conditional expectations to the threshold problem for partial expectations. Via the approximation algorithm for the optimal partial expectation, we can conduct an approximate binary search for the optimal conditional expectation. Let us recall the reduction: Given an MDP $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$ and a rational ϑ, we add a new initial state s'_{init} from which s_{init} is reached with probability 1 and weight $-\vartheta$ to obtain an MDP $\mathcal{M}_\vartheta$. Then, $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} > \vartheta$ if and only if $\mathbb{PE}^{\max}_{\mathcal{M}_\vartheta,s'_{init}} > 0$. Let us denote the value $\mathbb{PE}^{\max}_{\mathcal{M}_\vartheta,s'_{init}}$ by $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}[-\vartheta]$, i.e., the parameter $-\vartheta$ indicates the starting weight in the initial state s_{init}.

The approximation algorithm works as follows: Let $\mathcal{M} = (S, Act, P, s_{init}, wgt, Goal)$ be an MDP with $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} < \infty$ and let $\varepsilon > 0$. After our pre-processing procedure, we can assume that $p \overset{\text{def}}{=} \Pr^{\min}_{\mathcal{M},s_{init}}(\Diamond goal)$ is positive. For the optimal conditional expectation, we know that

$$\mathbb{CE}^{\max}_{\mathcal{M},s_{init}} \in [\mathbb{CE}^{\mathfrak{Max}}_{\mathcal{M},s_{init}}, \mathbb{CE}^{ub}].$$

We perform a binary search to approximate $\mathbb{CE}^{\max}_{\mathcal{M},s_{init}}$: We put $A_0 := \mathbb{CE}^{\mathfrak{Max}}_{\mathcal{M},s_{init}}$ and $B_0 := \mathbb{CE}^{ub}$. Given A_i and B_i, let $\vartheta_i := (A_i + B_i)/2$. Then, we approximate $\mathbb{PE}^{\max}_{s_{init}}[-\vartheta_i]$ up to an absolute error of $p \cdot \varepsilon$. Let E_i be the value of this approximation. If $E_i \in [-2p \cdot \varepsilon, 2p \cdot \varepsilon]$, terminate and return θ_i as the approximation for $\mathbb{CE}^{\max}_{s_{init}}$. If $E_i < -2p \cdot \varepsilon$, put $A_{i+1} := A_i$ and $B_{i+1} := \vartheta_i$, and repeat. If $E_i > 2p \cdot \varepsilon$, put $A_{i+1} := \vartheta_i$ and $B_{i+1} := B_i$, and repeat.

Theorem 6.11. *The procedure terminates after at most* $\lceil \log((A_0 - B_0)/(p\cdot\varepsilon))\rceil$ *iterations and returns an* 3ε*-approximation of* $\mathbb{CE}^{\max}_{s_{init}}$ *in time exponential in the size of* $\mathcal{M}$ *and polynomial in* $\log(1/\varepsilon)$*.*

Proof. We begin by showing that the algorithm terminates after at most $\lceil \log((A_0 - B_0)/(\varepsilon\cdot p))\rceil$ many iterations, i.e. when $|A_i - B_i| \leq \varepsilon\cdot p$. We know that $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i] < 0$ if $E_i < -2p\varepsilon$ and $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i] > 0$ if $E_i > 2p\varepsilon$. By the reduction between the threshold problems, we conclude that $\mathbb{CE}^{\max}_{s_{init}} \in [A_{i+1}, B_{i+1}]$ at any time. So, after at most $log((A_0 - B_0)/(\varepsilon\cdot p))$ many iteration, we have that $|A_i - B_i| \leq \varepsilon\cdot p$ and hence $\mathbb{CE}^{\max}_{s_{init}} - \varepsilon\cdot p \leq \theta_i \leq \mathbb{CE}^{\max}_{s_{init}} + \varepsilon\cdot p$. We claim that then $E_i \in [-2p\varepsilon, 2p\varepsilon]$. Suppose $E_i < -2p\varepsilon$. Then $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i] < -p\varepsilon$. But we have

$$0 = \mathbb{PE}^{\max}_{s_{init}}[-\mathbb{CE}^{\max}_{s_{init}}] \leq \mathbb{PE}^{\max}_{s_{init}}[-\theta_i + p\varepsilon] \leq \mathbb{PE}^{\max}_{s_{init}}[-\theta_i] + p\varepsilon$$

contradicting the supposition. Analogously, we show that E_i cannot be greater than $2p\varepsilon$.

Next, we show that the algorithm returns an 3ε-approximation of $\mathbb{CE}^{\max}_{s_{init}}$. As soon as the algorithm terminates, we have that $E_i \in [-2p\varepsilon, 2p\varepsilon]$. So, $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i] \in [-3p\varepsilon, 3p\varepsilon]$. So there is a scheduler $\mathfrak{S}$ with

$$\mathbb{PE}^{\mathfrak{S}}_{s_{init}}[-\theta_i] = \mathbb{PE}^{\mathfrak{S}}_{s_{init}} - \theta_i \cdot \Pr^{\mathfrak{S}}_{s_{init}}(\Diamond goal) \geq -3p\varepsilon.$$

As $\Pr^{\mathfrak{S}}_{s_{init}}(\Diamond goal) \geq p$, this implies $\mathbb{CE}^{\max}_{s_{init}} \geq \mathbb{CE}^{\mathfrak{S}}_{s_{init}} \geq \theta_i - 3\varepsilon$. On the other hand, suppose that $\mathbb{CE}^{\max}_{s_{init}} > \theta_i + 3\varepsilon$. Then there is a scheduler $\mathfrak{T}$ with $\mathbb{CE}^{\mathfrak{T}}_{s_{init}} > \theta_i + 3\varepsilon$. For this scheduler, we have

$$0 < \mathbb{PE}^{\mathfrak{T}}_{s_{init}}[-\theta_i - 3\varepsilon] = \mathbb{PE}^{\mathfrak{T}}_{s_{init}}[-\theta_i] - 3\varepsilon\cdot \Pr^{\mathfrak{T}}_{s_{init}}(\Diamond goal) \leq \mathbb{PE}^{\mathfrak{T}}_{s_{init}}[-\theta_i] - 3\varepsilon\cdot p.$$

This contradicts $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i] \leq 3p\varepsilon$. Therefore, the algorithm indeed returns a 3ε-approximation of $\mathbb{CE}^{\max}_{s_{init}}$.

Finally, we show that the claimed running time is correct: The algorithm stops after at most $\lceil \log((A_0 - B_0)/(\varepsilon\cdot p))\rceil$ iterations. As all values involved can be computed in polynomial time, this is polynomial in the size of $\mathcal{M}$ and linear in $\log(1/\varepsilon)$. In each iteration, we have to approximate the maximal partial expectation $\mathbb{PE}^{\max}_{s_{init}}[-\theta_i]$ up to an absolute error of $p\cdot\varepsilon$. As the logarithmic lengths of θ_i and p are polynomial in the size of $\mathcal{M}$ as well, this can be done in time exponential in the size of $\mathcal{M}$ and polynomial in $\log(1/\varepsilon)$. □

So, we have now shown that maximal partial and conditional expectations can be approximated as stated in Theorem 6.1. For the minimal values, we can simply first multiply all weights with -1.

6.4 Hardness of approximations

The approximation algorithms presented above require exponential time. This naturally raises the question whether optimal values could be approximated more efficiently. We conclude this chapter by proving that we cannot expect a polynomial-time approximation algorithm.

Theorem 6.12. *If P≠PSPACE, then there is no algorithm approximating the optimal partial or conditional expectation in an MDP $\mathcal{M}$ up to an absolute error of ε that runs in time polynomial in $\mathcal{M}$ and $\log(1/\varepsilon)$. This holds even for acyclic MDPs with non-negative weights.*

Proof. The threshold problem for maximal partial expectations is PSPACE-hard even for acyclic MDPs (see Theorem 3.32). More precisely, given an acyclic MDP $\mathcal{M}$ with non-negative weights weights and initial state s_{init}, a designated target state *goal*, and a rational ϑ, the problem to decide whether $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}} \geq \vartheta$ is PSPACE-hard. The transition probabilities in $\mathcal{M}$ are given as fractions of co-prime integers. Let D be the product of the denominators of all transition probabilities. Note that there are only polynomially many transitions and that the binary length of the denominators is part of the size of $\mathcal{M}$. Hence, the binary representation of D is polynomial in the size of $\mathcal{M}$.

Furthermore, we know that the maximal partial expectation is obtained by a deterministic scheduler $\mathfrak{S}$ (see Theorem 3.23). Let $\mathfrak{S}$ be an optimal deterministic scheduler. All $\mathfrak{S}$-paths from s_{init} to *goal* have a probability that is an integer multiple of $1/D$. Hence, also $\mathbb{PE}^{\mathfrak{S}}_{\mathcal{M},s_{init}} = \mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ is an integer multiple of $1/D$.

With an algorithm that approximates $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ up to an absolute error of ε in time polynomial in the size of $\mathcal{M}$ and in $\log(1/\varepsilon)$, we could approximate $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ up to an absolute error of $1/3D$ in time polynomial in the size of $\mathcal{M}$. Rounding the result to the closest integer multiple of $1/D$ would return the exact value $\mathbb{PE}^{\max}_{\mathcal{M},s_{init}}$ and allow us to solve the threshold problem in polynomial time.

The proof for the hardness of approximating optimal conditional expectations works analogously. □

CHAPTER SEVEN

CONCLUSIONS AND OUTLOOK

We conclude with brief summaries, final remarks, and hints at possible directions for future work regarding three topics that played an important role in this thesis: Positivity-hardness, saturation points, and approximations.

Positivity-hardness. Our investigations of two non-classical variants of the stochastic shortest path problems, the partial and the conditional stochastic shortest path problem, showed that these variants are most likely not solvable with currently known techniques. The decision versions of both variants are at least as hard as the Positivity problem for linear recurrence sequences, a problem that has been open for many decades and to which known number-theoretic techniques do not seem to be applicable. A decidability result for these non-classical stochastic shortest path problems and hence for the Positivity problem would lead to a major breakthrough in analytic number theory.

For the proof of these Positivity-hardness results, we constructed MDP-gadgets that encode a linear recurrence relation and the initial values of a linear recurrence sequence, respectively. These gadgets allow for great flexibility and the proof idea can easily be adapted to a series of further problems on MDPs – most of which have been studied and left open in the literature – by exchanging the gadget encoding the initial values. In this way, we proved that problems addressing the optimal termination probability and the optimal expected termination time of one-counter MDPs, the optimal satisfaction probability of energy objectives, the optimal probability that the accumulated weight satisfies an inequality constraint (cost problems, quantile queries), the optimal conditional value-at-risk for accumulated weights, and the optimal long-run probability of regular co-safety properties as well as the model-checking problem for frequency-LTL are Positivity-hard. This series of results shows that we developed a powerful technique to prove the inherent mathematical difficulty of optimization problems on (finite-state) MDPs.

We expect that the proof technique is applicable to further threshold problems associated to optimization problems on MDPs. A main requirement for the direct applicability of the technique is that the optimal values $V(s,w)$ in terms of the current state s and the weight w accumulated so far, or a similar quantity that can be increased and decreased, satisfy an optimality equation of the form

$$V(s,w) = \max_{\alpha \in Act(s)} \sum_{t \in S} P(s,\alpha,t) \cdot V(t, w + wgt(s,\alpha)).$$

In addition, the optimum must not be achievable with memoryless schedulers but the optimal decisions have to depend on the accumulated weight to make it possible to encode initial values of a linear recurrence sequence. This combination of conditions is quite common as we have seen. Furthermore, our and possible future Positivity-hardness results might be transferrable to further notions resulting from taking long-run averages (as in the case of long-run probabilities) or conditioning (as in the case of conditional expectations and conditional values-at-risk).

For optimization problems on MDPs, determining the structure of optimal schedulers is often a key step for the solution. We were able to prove that schedulers optimizing the partial or conditional expectation can be chosen to be weight-based and deterministic. A finer restriction on the necessary structure for optimal schedulers, however, is linked to deep questions surrounding the Positivity problem: In the MDPs constructed from the mentioned gadgets, the structure of the optimal scheduler is directly related to the negativity set $\{n \in \mathbb{N} \mid u_n < 0\}$ of the given linear recurrence sequence $(u_n)_{n \geq 0}$. The famous Skolem-Mahler-Lech theorem [Sko34, Mah35, Lec53] states that the set of zeros $\{n \in \mathbb{N} \mid u_n = 0\}$ is ultimately periodic, a.k.a. semi-linear, for any linear recurrence sequence over a field of characteristic 0. In [BG07], it is shown that this is not the case in general for the negativity sets of *real* linear recurrence sequences. To the best of our knowledge, it is not known whether the negativity set of *rational* linear recurrence sequences is always semi-linear. A proof that the optimal schedulers for any of the problems we have shown to be Positivity-hard can be chosen to be ultimately periodic with respect to the accumulated weight (cf. Section 3.2.3) would imply the analogue of the Skolem-Mahler-Lech theorem for the negativity set of rational linear recurrence sequences (and hence also for the positivity set).

Besides the Positivity-hardness that we established, the problems under consideration exhibit further complications that stand in the way of a proof of inter-reducibility with the Positivity problem. The MDPs constructed for the Positivity-hardness consist of an initial component in which positive weights are accumulated and afterwards the accumulated weight only decreases. General MDPs have a much more complicated structure. In this vein, it is also remarkable that the threshold problem for the probability that the accumulated cost when entering a goal state satisfies a Boolean combination of inequality constraints (cost problem) in finite-state Markov chains is open [HKL17]. For partial and conditional expectations, on the other hand, the computation is easy in Markov chains.

All in all, this leaves the possibility open that some or all of the problems we studied are in fact harder than the Positivity problem. In particular, it could be the case that the problems are undecidable and that a proof of the undecidability would yield no implications for the Positivity problem. For this reason, investigating whether some or all of the threshold problems are reducible to the Positivity problem constitutes a very interesting – and challenging – direction for future work. Such an inter-reducibility result would show that studying any of the discussed optimization problems on MDPs could be a worthwhile direction of research to settle the decidability status of the Positivity-problem. Some hope for an inter-reducibility result can be drawn from the fact that the optimal values are approximable for several of the problems – for termination probabilities and expected termination times of one-counter MDPs, this was shown in [BBEK11, BKNW12] and we proved this result for partial and conditional expectations in this thesis. This indicates that there is at least a major difference to undecidable problems in a similar context such as the emptiness problem for probabilistic finite automata [Paz71, CL89].

Saturation points. In MDPs with non-negative weights, the partial and conditional stochastic shortest path problems are solvable in exponential time. These results were established in [CFK$^+$13a] and [BKKW17], respectively. The key insight is the existence of saturation points, i.e., bounds on the accumulated weight after which optimal schedulers can behave memorylessly. So, optimal schedulers for partial and conditional expectations in MDPs with non-negative weights are not only weight-based and deterministic, but in addition they have to keep record of the accumulated weight only up to the saturation point. In particular, this means that there are optimal finite-memory schedulers. The saturation point provided in [CFK$^+$13a] in the context of stochastic multiplayer games relies on upper bounds for the optimal partial expectation. For MDPs, we showed that the least possible saturation point is computable in polynomial time without first computing an upper bound. For practical purposes, this least saturation point might significantly speed-up computations as the runtime of algorithms to compute the optimal partial expectation depends directly on the size of the computed saturation point. The saturation point for conditional expectations provided in [BKKW17] relies on upper bounds as well. Here, it is not clear whether an efficient computation of a lower or even the least possible saturation point similar to the computation of our saturation point for partial expectations is possible due to the more intricate inter-play between the accumulated weight and the probability to reach a goal state. The close connections between the two problems that we established by simple reductions between the threshold problems, however, might be useful to obtain a way to compute smaller saturation points for conditional expectations.

The concept of saturation points turns out to be very useful as it can be applied to further problems. First, we provided a simple saturation point for the computation of the optimal conditional value-at-risk of the accumulated weight before reaching a goal state in MDPs with non-negative weights. The saturation point simply provides a bound w on the accumulated weight such that paths exceeding this bound certainly belong to the p worst (or $1-p$ best) outcomes for a given probability value p. This allowed us to reduce the problem to the computation of the conditional value-at-risk for a weighted reachability problem in an exponentially large MDP. In [KM18], the optimal conditional value-at-risk for weighted reachability has been shown to be computable in polynomial time. The constructed MDP should allow us to also solve problems that address the simultaneous satisfaction of constraints on the conditional value-at-risk, the value-at-risk, and the expected value of the accumulated weight before reaching a goal state as such problems have been solved in [KM18] for weighted reachability objectives.

Second, we showed that saturation points can also be used to solve problems that are not as obviously related to stochastic shortest path problems. We investigated notions addressing the long-run satisfaction of path properties. In the non-probabilistic setting, long-run frequencies quantify how often a path property is satisfied on the suffixes of a run in a transition system. We proved that optimal long-run frequencies for regular co-safety properties given by an NFA can be computed in time polynomial in the size of the transition system and exponential in the size of the NFA. For the probabilistic setting, we introduced the notion of long-run probability quantifying the long-run average probability that a suffix satisfies a path property. The situation becomes much more complicated in the probabilistic setting and the threshold problem for optimal long-run probabilities of regular co-safety properties is Positivity-hard as mentioned above. For the restricted class of constrained reachability properties ($a \,\mathrm{U}\, b$), however, the existence of saturation points lead us to a solution. Here, saturation points are bounds on the number of consecutive visits to states labeled with a. For the proof for the existence of an efficiently computable saturation point, we used ideas similar to the proofs of the existence of saturation points for partial and conditional expectations. This allowed us to reduce the computation of the optimal long-run probability to the computation of the optimal expected mean payoff in an exponentially large MDP. For weighted MDPs, we furthermore introduced the notion of long-run expectation that quantifies the average expected value of the weight that will be accumulated before the next visit to a goal state. Also here, we proved the existence of a saturation point, even in MDPs with integer weights. Again, the saturation point provides a bound on the number of consecutive visits to certain states before optimal schedulers can switch to memoryless behavior.

In all mentioned cases, saturation points are computable in polynomial time and allow to solve the respective problems in exponential time. For conditional expectations, it has been shown in [BKKW17] that the threshold problem is PSPACE-hard in acyclic MDPs with non-negative weights. We transferred this result to partial expectations. Furthermore, we showed that the threshold problems for optimal long-run probabilities of constrained reachability properties and for optimal long-run expectations are NP-hard.

Closing the resulting complexity gaps and providing lower bounds (or a polynomial-time algorithm) for the conditional value-at-risk of accumulated weights before reaching a goal state remain as future work. Further, investigating to which extend the least possible saturation points for these problems can be computed as in the case of partial expectations could be fruitful, in particular for practical applications.

Approximation. While saturation points for the non-classical stochastic shortest path problems in MDPs with arbitrary integer weights do not exist, we established a weak analogue that can be seen as a precise version of the following simple idea: If the accumulated weight along a path is very high or very low, it is close to optimal to maximize or minimize the probability to reach a goal state in order to maximize partial and conditional expectations, respectively. Based on this idea, we showed that the optimal values for both non-classical stochastic shortest path problems in an MDP $\mathcal{M}$ can be approximated up to an absolute error of ε in time exponential in the size of $\mathcal{M}$ and polynomial in the accuracy $\log(1/\varepsilon)$. Approximation algorithms for termination probabilities and times of one-counter MDPs using a similar idea have been presented in [BBEK11, BKNW12].

We expect that this simple idea can be used to obtain approximation algorithms for further quantities such as optimal conditional values-at-risk for accumulated weights or optimal long-run probabilities of regular co-safety properties. If a co-safety property is given by a DFA, the optimization of long-run probabilities implicitly requires a trade-off analysis between the probabilities that runs starting in different states of the DFA are accepted. Here, approximate Pareto curves for the satisfaction probabilities of multiple ω-regular objectives that were shown to be efficiently computable in [EKVY07] might be helpful. Furthermore, this could lead into the direction of approximation algorithms for long-run probabilities of more general ω-regular properties although further complications have to be expected here.

We showed that there is no polynomial-time approximation algorithm for optimal partial or conditional expectations if P $\neq$ PSPACE. Nevertheless, there are some possible improvements for our exponential-time approximation algorithm: Our approximation algorithms rely on the *exact* solution to an exponentially large weighted reachability problem. An *approximate* solution to this exponentially large problem, however, would be sufficient to approximate optimal partial and conditional expectations. Standard procedures to obtain approximate solutions such as value iteration (see, e.g., [Put94]) might lead to much faster approximation algorithms in practice. For the classical stochastic shortest path problem, [YB13] presents an involved approximation algorithm combining Q-learning and policy iteration. It is worth investigating to which extend a similar approach to the non-classical stochastic shortest path problems is fruitful. Of course, such approximation approaches are also interesting for partial and conditional expectations in MDPs with non-negative weights.

BIBLIOGRAPHY

[AAGT15] Manindra Agrawal, Sundararaman Akshay, Blaise Genest, and P. S. Thiagarajan. Approximate verification of the symbolic dynamics of Markov chains. *Journal of the ACM*, 62(1):1–34, 2015.

[AAOW15] S. Akshay, Timos Antonopoulos, Joël Ouaknine, and James Worrell. Reachability problems for Markov chains. *Information Processing Letters*, 115(2):155–158, 2015.

[AHK03] Suzana Andova, Holger Hermanns, and Joost-Pieter Katoen. Discrete-time rewards model-checked. In Kim Guldstrand Larsen and Peter Niebert, editors, *First International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 2791 of *Lecture Notes in Computer Science*, pages 88–104. Springer, 2003.

[AT02] Carlo Acerbi and Dirk Tasche. Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes*, 31(2):379–388, 2002.

[BAGM12] Amir M. Ben-Amram, Samir Genaim, and Abu Naser Masud. On the termination of integer loops. *ACM Transactions on Programming Languages and Systems*, 34(4):1–24, 2012.

[BBC$^+$14] Tomás Brázdil, Václav Brozek, Krishnendu Chatterjee, Vojtech Forejt, and Antonín Kucera. Two views on multiple mean-payoff objectives in Markov decision processes. *Logical Methods in Computer Science*, 10(1), 2014.

[BBD$^+$18] Christel Baier, Nathalie Bertrand, Clemens Dubslaff, Daniel Gburek, and Ocan Sankur. Stochastic shortest paths and weight-bounded properties in Markov decision processes. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 86–94. ACM, 2018.

[BBE$^+$10] Tomás Brázdil, Václav Brožek, Kousha Etessami, Antonín Kučera, and Dominik Wojtczak. One-counter Markov decision processes. In *21st Annual*

ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 863–874. SIAM, 2010.

[BBEK11] Tomáš Brázdil, Václav Brožek, Kousha Etessami, and Antonín Kučera. Approximating the termination value of one-counter MDPs and stochastic games. In Luca Aceto, Monika Henzinger, and Jiří Sgall, editors, *38th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 6755 of *Theoretical Computer Science and General Issues*, pages 332–343. Springer, 2011.

[BBPS19] Christel Baier, Nathalie Bertrand, Jakob Piribauer, and Ocan Sankur. Long-run satisfaction of path properties. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–14. IEEE, 2019.

[BCHK11] Udi Boker, Krishnendu Chatterjee, Thomas A. Henzinger, and Orna Kupferman. Temporal specifications with accumulative values. In *26th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 43–52. IEEE, 2011.

[BDD$^+$14] Christel Baier, Marcus Daum, Clemens Dubslaff, Joachim Klein, and Sascha Klüppelholz. Energy-utility quantiles. In Julia M. Badger and Kristin Yvonne Rozier, editors, *6th NASA Formal Methods Symposium (NFM)*, volume 8430 of *Programming and Software Engineering*, pages 285–299. Springer, 2014.

[BDK$^+$14] Christel Baier, Clemens Dubslaff, Joachim Klein, Sascha Klüppelholz, and Sascha Wunderlich. Probabilistic model checking for energy-utility analysis. In Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten, editors, *Horizons of the Mind. A Tribute to Prakash Panangaden*, volume 8464 of *Lecture Notes in Computer Science*, pages 96–123. Springer, 2014.

[BDL12] Benedikt Bollig, Normann Decker, and Martin Leucker. Frequency linear-time temporal logic. In *6th International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 85–92. IEEE, 2012.

[BEFH16] Gilles Barthe, Thomas Espitau, Luis María Ferrer Fioriti, and Justin Hsu. Synthesizing probabilistic invariants via Doob's decomposition. In Swarat Chaudhuri and Azadeh Farzan, editors, *28th International Conference on Computer Aided Verification (CAV), Part I*, volume 9779 of *Lecture Notes in Computer Science*, pages 43–61. Springer, 2016.

[BG07] Jason P. Bell and Stefan Gerhold. On the positivity set of a linear recurrence sequence. *Israel Journal of Mathematics*, 157(1):333–345, 2007.

[BGB12] Christel Baier, Marcus Größer, and Nathalie Bertrand. Probabilistic ω-automata. *Journal of the ACM*, 59(1):1:1–1:52, 2012.

[BGC09] Christel Baier, Marcus Größer, and Frank Ciesinski. Quantitative analysis under fairness constraints. In Zhiming Liu and Anders P. Ravn, editors, *7th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 5799 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2009.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking.* MIT Press, 2008.

[BKKW14] Christel Baier, Joachim Klein, Sascha Klüppelholz, and Sascha Wunderlich. Weight monitoring with linear temporal logic: Complexity and decidability. In *29th Symposium on Logic In Computer Science (LICS)*, pages 11:1–11:10. ACM, 2014.

[BKKW17] Christel Baier, Joachim Klein, Sascha Klüppelholz, and Sascha Wunderlich. Maximizing the conditional expected reward for reaching the goal. In Axel Legay and Tiziana Margaria, editors, *23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 10206 of *Lecture Notes in Computer Science*, pages 269–285. Springer, 2017.

[BKM16] Michael Blondin, Andreas Krebs, and Pierre McKenzie. The complexity of intersecting finite automata having few final states. *Computational Complexity*, 25(4):775–814, 2016.

[BKN16] Tomás Brázdil, Antonín Kucera, and Petr Novotný. Optimizing the expected mean payoff in energy Markov decision processes. In Cyrille Artho, Axel Legay, and Doron Peled, editors, *14th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, volume 9938 of *Lecture Notes in Computer Science*, pages 32–49. Springer, 2016.

[BKNW12] Tomáš Brázdil, Antonín Kučera, Petr Novotnỳ, and Dominik Wojtczak. Minimizing expected termination time in one-counter Markov decision processes. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *39th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 7392 of *Theoretical Computer Science and General Issues*, pages 141–152. Springer, 2012.

[BKSV08] Noam Berger, Nevin Kapur, Leonard Schulman, and Vijay Vazirani. Solvency games. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *28th international conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 2 of *LIPIcs*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008.

[Bla54] David Blackwell. On optimal systems. *The Annals of Mathematical Statistics*, 25:394–397, 1954.

[BMM14] Patricia Bouyer, Nicolas Markey, and Raj Mohan Matteplackel. Averaging in LTL. In Paolo Baldan, editor, *25th International Conference on Concurrency Theory (CONCUR)*, volume 8704 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 2014.

[Bra06] Mark Braverman. Termination of integer linear programs. In Thomas Ball and Robert B. Jones, editors, *18th International Conference on Computer Aided Verification (CAV)*, volume 4144 of *Theoretical Computer Science and General Issues*, pages 372–385. Springer, 2006.

[BRS06] Daniele Beauquier, Alexander Rabinovich, and Anatol Slissenko. A logic of probability with decidable model checking. *Journal of Logic and Computation*, 16(4):461–487, 2006.

[BT91] Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[Can88] John Canny. Some algebraic and geometric computations in pspace. In *20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 460–467. ACM, 1988.

[CBGK08] Frank Ciesinski, Christel Baier, Marcus Größer, and Joachim Klein. Reduction techniques for model checking Markov decision processes. In *5th International Conference on Quantitative Evaluation of Systems (QEST)*, pages 45–54. IEEE, 2008.

[CCT16] Krishnendu Chatterjee, Martin Chmelík, and Mathieu Tracol. What is decidable about partially observable Markov decision processes with ω-regular objectives. *Journal of Computer and System Sciences*, 82(5):878–911, 2016.

[CD11] Krishnendu Chatterjee and Laurent Doyen. Energy and mean-payoff parity Markov decision processes. In Filip Murlak and Piotr Sankowski, editors, *36th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6907 of *Lecture Notes in Computer Science*, pages 206–218. Springer, 2011.

[CE81] Edmund M Clarke and E Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Dexter Kozen, editor, *Workshop on Logic of Programs*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[CFG16] Krishnendu Chatterjee, Hongfei Fu, and Amir Kafshdar Goharshady. Termination analysis of probabilistic programs through Positivstellensatz's. In Swarat Chaudhuri and Azadeh Farzan, editors, *28th International Conference on Computer Aided Verification (CAV), Part I*, volume 9779 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2016.

[CFK$^+$13a] Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, David Parker, and Aistis Simaitis. Automatic verification of competitive stochastic systems. *Formal Methods in System Design*, 43(1):61–92, 2013.

[CFK$^+$13b] Taolue Chen, Vojtěch Forejt, Marta Kwiatkowska, David Parker, and Aistis Simaitis. Prism-games: A model checker for stochastic multi-player games. In Nir Piterman and Scott A. Smolka, editors, *19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 7795 of *Lecture Notes in Computer Science*, pages 185–191. Springer, 2013.

[CGK13] Krishnendu Chatterjee, Andreas Gaiser, and Jan Kretínský. Automata with generalized Rabin pairs for probabilistic model checking and LTL synthesis. In Natasha Sharygina and Helmut Veith, editors, *25th International Conference on Computer Aided Verification (CAV)*, volume 8044 of *Lecture Notes in Computer Science*, pages 559–575. Springer, 2013.

[CH11] Krishnendu Chatterjee and Monika Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1318–1336. SIAM, 2011.

[CHR12] Pavol Cerný, Thomas A. Henzinger, and Arjun Radhakrishna. Simulation distances. *Theoretical Computer Science*, 413(1):21–35, 2012.

[CKV06] Hana Chockler, Orna Kupferman, and Moshe Y. Vardi. Coverage metrics for temporal logic model checking. *Formal Methods in System Design*, 28(3):189–212, 2006.

[CL89] Anne Condon and Richard J. Lipton. On the complexity of space bounded interactive proofs. In *30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 462–467. IEEE, 1989.

[COW16] Ventsislav Chonev, Joël Ouaknine, and James Worrell. On the Skolem problem for continuous linear dynamical systems. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 55 of *LIPIcs*, pages 100:1–100:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[CS09] Hana Chockler and Ofer Strichman. Before and after vacuity. *Formal Methods in System Design*, 34(1):37–58, 2009.

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[dA97] Luca de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[dA98] Luca de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *13th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 454–465. IEEE, 1998.

[dA99] Luca de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In Jos C. M. Baeten and Sjouke Mauw, editors, *10th International Conference on Concurrency Theory (CONCUR)*, volume 1664 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1999.

[Der70] Cyrus Derman. *Finite state Markovian decision processes*. Academic Press, 1970.

[DJKV17] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In Rupak Majumdar and Viktor Kunčak, editors, *29th International Conference on Computer Aided Verification (CAV)*, volume 10427 of *Lecture Notes in Computer Science*, pages 592–600. Springer, 2017.

[EKVY07] K. Etessami, M. Kwiatkowska, M. Y. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. In Orna Grumberg and Michael Huth, editors, *13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 4424 of *Lecture Notes in Computer Science*, pages 50–65. Springer, 2007.

[EvdPSW03] Graham Everest, Alfred Jacobus van der Poorten, Igor Shparlinski, and Thomas Ward. *Recurrence sequences.* ACM, 2003.

[EY05] Kousha Etessami and Mihalis Yannakakis. Recursive Markov decision processes and recursive stochastic games. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 891–903. Springer, 2005.

[EY15] Kousha Etessami and Mihalis Yannakakis. Recursive Markov decision processes and recursive stochastic games. *Journal of the ACM*, 62(2):1–69, 2015.

[EZ62] J.H. Eaton and L.A. Zadeh. Optimal pursuit strategies in discrete-state probabilistic systems. *Transactions of the ASME, Series D, Journal of Basic Engineering*, 84(1):23–29, 1962.

[FK15] Vojtech Forejt and Jan Krcál. On frequency LTL in probabilistic systems. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory (CONCUR)*, volume 42 of *LIPIcs*, pages 184–197. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[FKK15] Vojtech Forejt, Jan Krcál, and Jan Kretínský. Controller synthesis for MDPs and frequency $\mathrm{LTL}_{\setminus\mathrm{GU}}$. In Martin Davis, Ansgar Fehnker, Annabelle McIver, and Andrei Voronkov, editors, *20th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 9450 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 2015.

[FSBW13] Dror Fried, Solomon Eyal Shimony, Amit Benbassat, and Cenny Wenner. Complexity of Canadian traveler problem variants. *Theoretical Computer Science*, 487:1–16, 2013.

[GKM14] Friedrich Gretz, Joost-Pieter Katoen, and Annabelle McIver. Operational versus weakest pre-expectation semantics for the probabilistic guarded command language. *Performance Evaluation*, 73:110–132, 2014.

[Hen13] Thomas A. Henzinger. Quantitative reactive modeling and verification. *Computer Science - Research and Development*, 28(4):331–344, 2013.

[HHHK05] Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem's problem–on the border between decidability and undecidability.

Technical report, Technical Report 683, Turku Centre for Computer Science, 2005.

[HJ94] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[HK79] Arie Hordijk and Lodewijk Kallenberg. Linear programming and Markov decision chains. *Management Science*, 25(4):352–362, 1979.

[HK15] Christoph Haase and Stefan Kiefer. The odds of staying on budget. In Magnus M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 9134 of *Theoretical Computer Science and General Issues*, pages 234–246. Springer, 2015.

[HKL17] Christoph Haase, Stefan Kiefer, and Markus Lohrey. Computing quantiles in Markov chains with multi-dimensional costs. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[HO13] Thomas A. Henzinger and Jan Otop. From model checking to model measuring. In Pedro R. D'Argenio and Hernan Melgratti, editors, *24th International Conference on Concurrency Theory (CONCUR)*, volume 8052 of *Lecture Notes in Computer Science*, pages 273–287. Springer, 2013.

[Kal83] Lodewijk Kallenberg. Linear programming and finite Markovian control problems. *Mathematical Center Tracts*, 148, 1983.

[Kal11] Lodewijk Kallenberg. *Markov Decision Processes.* Lecture Notes. University of Leiden, 2011.

[Kar78] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309 – 311, 1978.

[KGJ$^+$15] Joost-Pieter Katoen, Friedrich Gretz, Nils Jansen, Benjamin Lucien Kaminski, and Federico Olmedo. Understanding probabilistic programs. In Roland Meyer, André Platzer, and Heike Wehrheim, editors, *Correct System Design - Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday*, volume 9360 of *Lecture Notes in Computer Science*, pages 15–32. Springer, 2015.

[KK15] Benjamin Lucien Kaminski and Joost-Pieter Katoen. On the hardness of almost–sure termination. In Giuseppe F. Italiano, Giovanni Pighizzini, and

Donald T. Sannella, editors, *40th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 9235 of *Theoretical Computer Science and General Issues*, pages 307–318. Springer, 2015.

[KLS08] Orna Kupferman, Wenchao Li, and Sanjit A. Seshia. A theory of mutations with applications to vacuity, coverage, and fault tolerance. In *Formal Methods in Computer-Aided Design (FMCAD)*, pages 1–9. IEEE, 2008.

[KM18] Jan Kretínský and Tobias Meggendorfer. Conditional value-at-risk for reachability and mean payoff in Markov decision processes. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 609–618. ACM, 2018.

[KNP11] Marta Kwiatkowska, Gethin Norman, and David Parker. Prism 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *23rd International Conference on Computer Aided Verification (CAV)*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011.

[Koz77] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 254–266. IEEE, 1977.

[Kul16] Vidyadhar G. Kulkarni. *Modeling and analysis of stochastic systems.* CRC Press, 2016.

[KV99] Orna Kupferman and Moshe Y. Vardi. Robust satisfaction. In Jos C. M. Baeten and Sjouke Mauw, editors, *10th International Conference on Concurrency Theory (CONCUR)*, volume 1664 of *Lecture Notes in Computer Science*, pages 383–398. Springer, 1999.

[KV03] Orna Kupferman and Moshe Y. Vardi. Vacuity detection in temporal model checking. *International Journal on Software Tools for Technology Transfer*, 4(2):224–233, 2003.

[Lec53] Christer Lech. A note on recurring series. *Arkiv för Matematik*, 2(5):417–421, 1953.

[LPM$^+$15] Yu Lu, Zhaoguang Peng, Alice A Miller, Tingdi Zhao, and Christopher W Johnson. How reliable is satellite navigation for aviation? Checking availability properties with probabilistic verification. *Reliability Engineering & System Safety*, 144:95–116, 2015.

[Mah35] Kurt Mahler. Eine arithmetische Eigenschaft der Taylor-Koeffizienten rationaler Funktionen. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen*, 38:50–60, 1935.

[MHC99] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *16th National Conference on Artificial Intelligence (AAAI)*, pages 541–548. MIT Press, 1999.

[MMM05] Annabelle McIver, Carroll Morgan, and Charles Carroll Morgan. *Abstraction, refinement and proof for probabilistic systems*. Springer, 2005.

[MSS20] Rupak Majumdar, Mahmoud Salamati, and Sadegh Soudjani. On decidability of time-bounded reachability in CTMDPs. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPIcs*, pages 133:1–133:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[MSTW17] Richard Mayr, Sven Schewe, Patrick Totzke, and Dominik Wojtczak. MDPs with energy-parity objectives. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[Odi92] Piergiorgio Odifreddi. *Classical recursion theory: The theory of functions and sets of natural numbers*. Elsevier, 1992.

[OGJ$^+$18] Federico Olmedo, Friedrich Gretz, Nils Jansen, Benjamin Lucien Kaminski, Joost-Pieter Katoen, and Annabelle Mciver. Conditioning in probabilistic programming. *ACM Transactions on Programming Languages and Systems*, 40(1):4:1–4:50, 2018.

[OW12] Joël Ouaknine and James Worrell. Decision problems for linear recurrence sequences. In Alain Finkel, Jerome Leroux, and Igor Potapov, editors, *6th International Workshop on Reachability Problems (RP)*, volume 7550 of *Theoretical Computer Science and General Issues*, pages 21–28. Springer, 2012.

[OW14a] Joël Ouaknine and James Worrell. On the positivity problem for simple linear recurrence sequences. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8573 of *Theoretical Computer Science and General Issues*, pages 318–329. Springer, 2014.

[OW14b] Joël Ouaknine and James Worrell. Positivity problems for low-order linear recurrence sequences. In *25th Annual ACM-SIAM Symposium on Discrete algorithms (SODA)*, pages 366–379. SIAM, 2014.

[OW14c] Joël Ouaknine and James Worrell. Ultimate positivity is decidable for simple linear recurrence sequences. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *41st International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8573 of *Theoretical Computer Science and General Issues*, pages 330–341. Springer, 2014.

[OW15] Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *ACM SIGLOG News*, 2(2):4–13, 2015.

[Pap85] Christos H. Papadimitriou. Games against nature. *Journal of Computer and System Science*, 31(2):288–301, 1985.

[Paz71] Azaria Paz. *Introduction to Probabilistic Automata.* Academic Press, 1971.

[PB19] Jakob Piribauer and Christel Baier. Partial and conditional expectations in Markov decision processes with integer weights. In Mikolaj Bojanczyk and Alex Simpson, editors, *22nd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 11425 of *Lecture Notes in Computer Science*, pages 436–452. Springer, 2019.

[PB20] Jakob Piribauer and Christel Baier. On Skolem-hardness and saturation points in Markov decision processes. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *LIPIcs*, pages 138:1–138:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 46–57. IEEE, 1977.

[PT87] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., 1994.

[QS82] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In M. Dezani-Ciancaglini and U. Montanari, editors, *International Symposium on Programming*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 1982.

[RRS17] Mickael Randour, Jean-François Raskin, and Ocan Sankur. Percentile queries in multi-dimensional Markov decision processes. *Formal Methods in System Design*, 50(2-3):207–248, 2017.

[Sko34] Thoralf Skolem. Ein Verfahren zur Behandlung gewisser exponentialer Gleichungen und diophantischer Gleichungen. *Comptes Rendus du Congrès des Mathématiciens Scandinaves*, 8:163–188, 1934.

[STM84] Tarlok N. Shorey, Robert Tijdeman, and Max Mignotte. The distance between terms of an algebraic recurrence sequence. *Journal für die reine und angewandte Mathematik*, 1984(349):63–76, 1984.

[Tiw04] Ashish Tiwari. Termination of linear programs. In Rajeev Alur and Doron Peled, editors, *16th International Conference on Computer Aided Verification (CAV)*, volume 3114 of *Lecture Notes in Computer Science*, pages 70–82. Springer, 2004.

[TN16] Paulo Tabuada and Daniel Neider. Robust linear temporal logic. In Laurent Regnier and Jean-Marc Talbot, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL)*, volume 62 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[UB13] Michael Ummels and Christel Baier. Computing quantiles in Markov reward models. In Frank Pfenning, editor, *16th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 7794 of *Lecture Notes in Computer Science*, pages 353–368. Springer, 2013.

[Ury00] Stanislav Uryasev. Conditional value-at-risk: optimization algorithms and applications. In *Computational Intelligence and Financial Engineering (CIFEr)*, pages 49–57. IEEE, 2000.

[Var85] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 327–338. IEEE, 1985.

[vEJ11] Christian von Essen and Barbara Jobstmann. Synthesizing systems with optimal average-case behavior for ratio objectives. In *International Workshop on Interactions, Games and Protocols (iWIGP)*, pages 17–32. EPTCS, 2011.

[Ver85] NK Vereshchagin. The problem of appearance of a zero in a linear recurrence sequence. *Mat. Zametki*, 38(2):609–615, 1985.

[VW86] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *First Symposium on Logic in Computer Science (LICS)*, pages 322–331. IEEE, 1986.

[Whi83] Peter Whittle. *Optimization over time (Vol. 2).* John Wiley & Sons, Inc., 1983.

[YB13] Huizhen Yu and Dimitri P. Bertsekas. Q-learning and policy iteration algorithms for stochastic shortest path problems. *Annals of Operations Research*, 208(1):95–132, 2013.

LIST OF FIGURES

ACKNOWLEDGEMENTS

During the research for and the writing of this thesis, I received plenty of support without which this project would not have been possible.

First and foremost, I want to thank my advisor Christel Baier for her relentless help, for introducing me to this great topic of research, and for many productive and inspiring discussions. Your guidance and advice have been invaluable. I have learned a lot in these last years!

It has been a pleasure to work and cooperate with my colleagues at the chair and in QuantLA. The lunch and cake sessions and many discussions in the office as well as the trips to Leipzig and seminars in Dresden played an essential role in creating an enjoyable environment. It's a pity that we have met so rarely in person recently.

Special thanks also go to Nathalie Bertrand and Ocan Sankur! It has been a joy to collaborate with you. Our fruitful meetings in Dresden, Rennes, and online have led to great results and ideas – some of which also play an important role in this thesis.

I am indebted to the DFG for the funding I received through QuantLA.

On a personal note, I am very grateful that my parents and brothers have encouraged and supported me on the way that lead me here throughout my life. I can't thank you enough! To Sophia, I am very glad to have you in my life. Thank you for always being there for me!