Certificates and Witnesses for Probabilistic Model Checking

Dissertation

eingereicht zur Erlangung des akademischen Grades

Doctor Rerum Naturalium (Dr. rer. nat.)

vorgelegt an der

Technischen Universität Dresden

Fakultät Informatik

eingereicht von

Simon Jantsch

geboren am 23. September 1992 in Wien

verteidigt am 6.7.2022

Begutachtet von

Prof. Dr. Christel Baier

Technische Universität Dresden

Prof. Dr. Nils Jansen

Radboud University Nijmegen

Abstract

The ability to provide succinct information about why a property does, or does not, hold in a given system is a key feature in the context of formal verification and model checking. It can be used both to explain the behavior of the system to a user of verification software, and as a tool to aid automated abstraction and synthesis procedures. Counterexample traces, which are executions of the system that do not satisfy the desired specification, are a classical example. Specifications of systems with probabilistic behavior usually require that an event happens with sufficiently high (or low) probability. In general, single executions of the system are not enough to demonstrate that such a specification holds. Rather, standard witnesses in this setting are sets of executions which in sum exceed the required probability bound.

In this thesis we consider methods to certify and witness that probabilistic reachability constraints hold in Markov decision processes (MDPs) and probabilistic timed automata (PTA). Probabilistic reachability constraints are threshold conditions on the maximal or minimal probability of reaching a set of target-states in the system. The threshold condition may represent an upper or lower bound and be strict or non-strict. We show that the model-checking problem for each type of constraint can be formulated as a satisfiability problem of a system of linear inequalities. These inequalities correspond closely to the probabilistic transition matrix of the MDP. Solutions of the inequalities are called *Farkas certificates* for the corresponding property, as they can indeed be used to easily validate that the property holds.

By themselves, Farkas certificates do not explain *why* the corresponding probabilistic reachability constraint holds in the considered MDP. To demonstrate that the maximal reachability probability in an MDP is above a certain threshold, a commonly used notion are *witnessing subsystems*. A subsystem is a witness if the MDP satisfies the lower bound on the optimal reachability probability even if all states not included in the subsystem are made rejecting trap states. Hence, a subsystem is a part of the MDP which by itself satisfies the lower-bounded threshold constraint on the optimal probability of reaching the target-states. We consider witnessing subsystems for lower bounds on both the maximal and minimal reachability probabilities, and show that Farkas certificates and witnessing subsystems are related. More precisely, the support (i.e., the indices with a non-zero entry) of a Farkas certificate induces the state-space of a witnessing subsystem for the corresponding property. Vice versa, given a witnessing subsystem one can compute a Farkas certificate whose support corresponds to the state-space of the witness.

This insight yields novel algorithms and heuristics to compute small and minimal witnessing subsystems. To compute minimal witnesses, we propose mixed-integer linear programming formulations whose solutions are Farkas certificates with minimal support. We show that the corresponding decision problem is NP-complete even for acyclic Markov chains, which supports the use of integer programs to solve it. As this approach does not scale well to large instances, we introduce the quotient-sum heuristic, which is based on iteratively solving a sequence of linear programs. The solutions of these linear programs are also Farkas certificates. In an experimental evaluation we show that the quotient-sum heuristic is competitive with state-of-the-art methods. A large part of the algorithms proposed in this thesis are implemented in the tool Storm.

We study the complexity of computing minimal witnessing subsystems for probabilistic systems that are similar to trees or paths. Formally, this is captured by the notions of *tree width* and *path width*. Our main result here is that the problem of computing minimal witnessing subsystems remains NP-complete even for Markov chains with bounded path width. The

hardness proof identifies a new source of combinatorial hardness in the corresponding decision problem.

Probabilistic timed automata generalize MDPs by including a set of clocks whose values determine which transitions are enabled. They are widely used to model and verify real-time systems. Due to the continuously-valued clocks, their underlying state-space is inherently uncountable. Hence, the methods that we describe for finite-state MDPs do not carry over directly to PTA. Furthermore, a good notion of witness for PTA should also take into account timing aspects. We define two kinds of subsystems for PTA, one for maximal and one for minimal reachability probabilities, respectively. As for MDPs, a subsystem of a PTA is called a witness for a lower-bounded constraint on the (maximal or minimal) reachability probability, if it itself satisfies this constraint. Then, we show that witnessing subsystems of PTA induce Farkas certificates in certain finite-state quotients of the PTA. Vice versa, Farkas certificates of such a quotient induce witnesses of the PTA. Again, the support of the Farkas certificates corresponds to the states included in the subsystem. These insights are used to describe algorithms for the computation of minimal witnessing subsystems for PTA, with respect to three different notions of size. One of them counts the number of locations in the subsystem, while the other two take into account the possible clock valuations in the subsystem.

Acknowledgements

First and foremost I would like to thank my supervisor *Christel Baier*. Your experience and expertise have been invaluable for my work towards this thesis throughout the last years. I am grateful for your guidance and support in all parts of academic life. Your commitment to research and teaching is an inspiration for me.

I would like to thank all of my colleagues at the chair for Algebraic and Logic Foundations of Computer Science in Dresden for the friendly and lively atmosphere, which I enjoyed a lot. It is a real pity that office life changed so drastically half way into my PhD. In particular, many thanks to *Clemens Dubslaff*, *Nikolai Käfer*, *Jakob Piribauer* and *Patrick Wienhöft* for your valuable comments after reading parts of this thesis, and to *Max Korn* for your help and support with the server setup for the experiments!

In the last years, I was lucky to get the chance to work with many inspiring people. I would like to thank my coauthors for the fruitful collaborations from which I have learned a lot. Special thanks go to *Florian Funke* for the energy and skill that you bring to every project, and to *Hans Harder* for your work on the implementation of Switss.

I am indebted to my friends and colleagues from QuantLA, for all the seminars, workshops, discussions and trips. The same thanks goes out to my friends and colleagues in CPEC, including for your hospitality when we visited in Saarbrücken. I am also grateful to the DFG for supporting me financially through these projects.

Of course, there is life outside of uni. I am extremely grateful for the support and love that I have always received from my friends and family.

Contents

Chapter 1

Introduction

The question of how to establish whether or not a program satisfies fundamental properties such as *correctness, safety* and *termination* has accompanied the field of computer science since its birth. Today, several large research areas including *testing* [Kin76, GKS05], *(higher-order) theorem proving* [NPW02, BC04] and *model checking* [BK08, CES09, CHVB18] approach it from very different angles. It is clear that a diverse toolkit is essential to tackle the problems at hand, given the fast and enormous growth in complexity, size, scope and diversity that computing technology has gone through in the last decades.

Model checking arose in the 1980's in pioneering work by Clarke and Emerson [EC82] and Queille and Sifakis [QS82]. It is a *fully automated* verification methodology, which is based on modeling programs as *transition systems* and properties they should satisfy using *temporal logics* such as *linear temporal logic* (LTL) and *computation tree logic* (CTL). This framework allows precise mathematical definitions and builds on the long tradition of logic-based methods in computer science. Prominent innovations that have shaped the field of model checking since include the *automata-theoretic* approach to model checking [VW86], *abstraction-based* methods [CGL94, CGJ+03], *symbolic* model checking [BCM+92] and the introduction of methods from Boolean *satisfiability-* [BCCZ99] and *satisfiability-modulo-theories* checking [AMP09]. A number of software tools implementing various model checking algorithms, both for software and hardware systems, have been built [LPY97, BHJM07, Ben08, BLR11, KT14]. Especially in the area of hardware design, model checking is by now an established and mature technology widely used in the industry.

An important line of research in the area of model checking has been to extend its scope to new kinds of systems and programs. In this spirit, a theory of *timed automata* [ACD93, AD94] was developed, which allows modeling and verifying real-time systems. Pushing this idea further yields *hybrid automata* [Hen96], which include continuously-valued variables whose behavior is described using differential equations. These models have been proven to be extremely useful for applying automated verification techniques to dynamical and cyber-physical systems.

At the same time, the study of model checking for *probabilistic systems* [Var85, Han91, CY95] was initiated. Such systems arise in various contexts, where probabilities may represent *failure probabilities* of physical components or unreliable communication channels, *assumptions or*

knowledge on the likelihood of events (e.g., input sequences) or *intrinsic probabilistic behavior* of programs or protocols as utilized by, e.g., randomized consensus algorithms [AH90]. The theory covers continuous-time [BHHK03] and parametrized [LMT07] probabilistic models. Several successful tools have been developed for probabilistic model checking, including Prism [KNP11], Storm [DJKV17] and others [CB06, KZH+11, HLS+14].

The standard discrete-time models that are used for probabilistic model checking are *Markov chains* and *Markov decision processes*. A Markov chain consists of a set of states and a probabilistic transition function. This function assigns to each state a probability distribution over the possible successor states. While in classical transition systems a linear-time property is either satisfied or not, in Markov chains it is satisfied with some probability. Markov decision processes (MDPs) add a layer of complexity by including *nondeterministic* branching. The execution of an MDP proceeds as follows. In a given state, one out of a set of possible probability distributions over the successor states is picked. Then, this distribution determines the probabilities of the next state in the same way as for Markov chains. The probability of a property in an MDP depends on how the choice of distributions is resolved. This is done by a *scheduler*, which maps each finite path of the MDP to one of the possible probability distributions. The nondeterminism in MDPs is crucial to model asynchronous concurrent systems, where the order of executions of different participating processes is not known beforehand.

1.1 Counterexamples, witnesses and certificates

One feature of many classical model checking algorithms is that if they establish that the system at hand does not satisfy the property, then they also provide a *counterexample*. For linear-time properties, a counterexample is typically an execution of the system which violates the property. In practice, counterexamples are extremely useful because they provide succinct information about *why* the property is violated. What exactly constitutes a counterexample depends on the kind of systems and properties that are considered [CJLV02, CV03].

Apart from providing useful information to the user of a verification tool, counterexamples are a key ingredient in the *counterexample-guided abstraction refinement* method [CGJ+03]. The idea is to start with a coarse abstraction of the system, which is iteratively refined until it is fine enough to provide a proof that the considered property holds in the system. The refinement is guided by *spurious counterexamples*, which exist in the current abstraction but not in the actual system.

Counterexamples serve as *explications* or *witnesses* for the violation of a property in a system. It is as desirable, but usually more difficult, to provide explications in case the property is satisfied. For example, for model checking of linear-time properties this amounts to showing that *all* executions satisfy the property. Common techniques to achieve this include deriving deductive proofs for positive model checking results [Nam01, PPZ01, BMS+17] or using rank-based certificates [KV04].

Witnesses in probabilistic model checking

In probabilistic model checking, individual executions often carry low probability and hence, in general, do not by themselves serve as an explication for most properties. To show that linear-time property φ holds with probability *at least* λ in a Markov chain $\mathcal{M}$, an explication has to demonstrate (intuitively speaking) that *a set of executions* of $\mathcal{M}$ satisfying φ exists, whose

total probability is at least λ. Dually, φ holds with probability *at most* λ if a set of executions *violating* φ exists with total probability exceeding $1-\lambda$. This shows that the conceptual difference between explications for positive and negative model checking results is not as pronounced in the probabilistic case.

For this reason we will henceforth use the positive terms witness, explication or certificate and say explicitly which property is meant. These notions will be distinguished as follows. A *witness* should carry information about *why* a given property holds in a system, in terms of the given system description. For example, execution traces, individual components or subsystems are possible witnesses. On the other hand, a *certificate* is any (mathematical) token which can be used to *easily validate* that the property holds. Here, "easily" means that an independent and "simple" computer program should be able to check the certificate. In particular, this check should be simpler than verifying the property in the first place. This meaning of certificate is proposed and used by the acknowledged theory of *certifying algorithms* [MMNS11].

The term *explication* includes witnesses and certificates and in general any object which carries information about whether or why a given property holds in a system. We use the term explication rather than *explanation*, because our emphasis lies on the mathematical objects which can be used to describe the behavior of a system, and the algorithmic questions of how to compute them. To provide a useful *explanation* to a user of a verification tool, usually further processing steps are required such as visualization and the selective presentation of parts of the explication [KNVG22].

We now give an overview of the state-of-the-art on witnesses (usually called counterexamples) in the context of probabilistic model checking. The distinction between counterexamples and witnesses is only a matter of terminology, as counterexamples are just witnesses for the negated property. A survey discussing many of these works can be found in [ÁBD+14]. More specific comparisons of the work presented in this thesis and existing literature will be given in the beginning of each chapter.

Path-based witnesses. The first notion of witnesses for probabilistic properties were *path-based*. As observed above, the fact that the probability of a reachability property exceeds λ in a Markov chain can be witnessed by a set of finite paths satisfying the property with total probability larger than λ [AHL05, AL06, HK07a]. Informative witnesses in this context are those which include few paths. The problem of finding a minimal witness in this sense has been addressed using a reformulation of the problem in terms of finding shortest paths in a weighted graph [HK07a, HKD09]. The work covers bounded and unbounded until properties, and bounds on the expected total reward. As the size of a minimal witness can be arbitrarily large with growing threshold λ, techniques to succinctly represent path-based witnesses using regular expressions have been proposed [DHK08, HKD09].

Heuristic approaches to compute small path-based witnesses were studied in [AL06, AL10] and applied to continuous-time Markov chains [AHL05, HK07b] and MDPs [AL09]. For MDPs, witnesses for lower-bounded (resp. upper-bounded) threshold constraints on the maximal (resp. minimal) reachability probabilities were considered. These properties can be witnessed by a scheduler whose induced reachability probability satisfies the threshold constraint. Two strategies were proposed to compute such witnesses. One is to first compute an optimal scheduler for the property, and then to apply known methods for Markov chains. The second is based on directly enumerating paths of the MDP. Notably, the algorithms described in [AL09, AL10] return so called *diagnostic subgraphs*, which are subsystems including all states that appear on some path

in the computed path-based witness. However, the optimization objective of these algorithms is not to produce small subgraphs, but rather to compute small (i.e., including few paths) path-based witnesses. These approaches have been implemented in the tool DIPRO [ALLS11].

To tackle the problem that the number of paths in a minimal path-based witness may be huge, symbolic methods based on bounded model checking using binary decision diagrams (BDDs) [WBB09] and satisfiability-modulo-theories (SMT) [BWB+11] have been proposed. A different approach to deal with large numbers of paths is to collapse strongly connected components and compute witnesses in the resulting acyclic model [ADvR09, ÁJW+10].

Witnessing subsystems. While earlier work had already represented path-based witnesses using (a kind of) subsystems, it was proposed later to define witnesses as subsystems in the first place [JÁK+11]. In particular, this changed the optimization criterion. For subsystems, natural criteria include the number of included states or transitions (or both), but not the number of paths it includes.

Several heuristics to compute small witnessing subsystems for Markov chains were introduced [JÁK+11] and implemented in the tool COMICS [JÁV+12]. They are based on iteratively extending subsystems until they satisfy the required lower bound on the reachability probability. BDD-based symbolic algorithms for these heuristics were developed in [JÁZ+13, JWÁ+14]. The problem of computing *minimal* subsystems for Markov chains and MDPs was addressed in [WJÁ+12], using *mixed-integer linear programs* (MILPs) and SMT-based methods. The properties considered in these works are lower bounds on the reachability probability (of the maximizing scheduler, for MDPs). A related definition of witnessing subsystem for MDPs has been introduced using simulation relations [CV10]. Computing minimal witnesses for MDPs is NP-complete for both notions [CV10, WJÁ+12].

High-level witnesses. Probabilistic systems are typically not specified directly as MDPs, but rather using some higher-level formalism. A standard modeling language is the one of PRISM [KNP11], which is based on the *reactive modules* paradigm [AH99].

In this setting, witnessing subsystems of the concrete state-based models may not be very informative, or simply prohibitively large. For this reason, notions of witnesses on the level of PRISM code in terms of *critical command sets* have been studied [DJW+14, KÁJW15]. To compute minimal critical command sets, the notion of *label-minimal witnessing subsystems* for MDPs was introduced [KÁJW15]. Rather than minimizing the number of states occurring in a witnessing subsystem, the idea is to minimize the number of participating labels in a witnessing subsystem of a transition-labeled MDP.

Computing the minimal critical command set of a PRISM program reduces to finding label-minimal witnesses in the corresponding labeled MDP, where the labels of a concrete transition correspond to the high-level commands that participate to form it. As computing minimal critical command sets using MILPs does not scale very well [KÁJW15], an approach which iteratively extends the command set until the induced MDP satisfies the lower bound on the maximal reachability probability has been proposed [DJW+14]. It leverages dependencies between commands which can be deduced from the PRISM-modules, and uses MAX-SAT [FM06] to find minimal command sets satisfying these dependencies.

Applications of witnesses in probabilistic model checking. As for non-probabilistic model checking, an important application of witnesses is to provide information of *why* certain proper-

ties hold or fail to hold. For example, a case study describing how probabilistic counterexamples can be used in the industrial design of an airbag controller is described in [AFG+09]. In this work, special visualization methods for probabilistic counterexamples are used [AL08]. Another approach to represent the information included in a witness for a probabilistic property is to compile it into a *fault tree* [KLL11].

A second important use case of witnesses are automated approaches such as *counterexample-guided abstraction refinement* (CEGAR) and *counterexample-guided inductive synthesis* (CEGIS). Both have been applied successfully in the context of probabilistic model checking. CEGAR for probabilistic systems has been proposed in [CHJM05], which also extends to two-player stochastic games under partial information, and in [HWZ08], which is based on predicate abstraction and uses path-based witnesses for lower bounds on the reachability probability to guide the abstraction-refinement. Another incarnation of CEGAR is proposed in [CV10], which considers the safety fragment of probabilistic computation tree logic (PCTL) and uses subsystems of MDPs as witnesses. A CEGAR-based approach has also been developed for probabilistic *hybrid* automata [LP19].

The goal of *counter-example guided inductive synthesis* (CEGIS) is to automatically synthesize (i.e., construct) a program satisfying a given specification out of a syntactically specified family of programs. In [ČHJK19], witnessing subsystems for Markov chains are used in a CEGIS-loop to prune the search space of possible program instances. Smaller witnessing subsystems are preferable as they witness the violation of the property in a larger part of the family. The approach is developed further in [AČJK21], where in particular novel heuristics for computing witnessing subsystems with few labels in Markov chains are proposed. They are based on greedily adding labels to (not-yet-witnessing) subsystems and using information of the program family to get smaller witnesses. This CEGIS framework forms a part of the tool Paynt [AČJ+21].

Beyond reachability. The problem of computing minimal witnessing subsystems for lower bounds on the maximal probability of satisfying an ω-regular property given as a *deterministic Rabin automaton* (DRA) was solved using a MILP-based approach in [WJÁ+14]. This work enables the computation of witnesses for LTL specifications by first applying a translation from LTL to DRA. Witnesses for a safety fragment of PCTL are considered in [CV10]. They show that computing state-minimal witnesses is NP-hard, and therefore present a polynomial-time algorithm which computes a witness such that removing any further state would break the witness property. Heuristics to compute small witnesses for PCTL properties have not been considered, to the best of our knowledge.

Another important class of properties measure the (optimal) *expected utility* or *cost* which can be achieved in a probabilistic system. In this setting, typically states (or state-action pairs) are paired with a numerical gain or cost in that state. One can now consider the expected value of several random variables such as the total accumulated value before reaching a designated state, the long-run average value, etc. Witnessing subsystems for the expected total (accumulated) reward have been studied in [QJD+15].

Other kinds of witnesses. A notable exception to the path-based and subsystem-based witnesses described above is proposed in [SVV09]. Here, witnesses for lower bounds on the probability of satisfying an LTL formula in a Markov chain are presented as a pair of two sets of path fragments (W, R). Intuitively, W consists of a set of finite initial paths whose total probability exceeds the lower bound, and R consists of path fragments such that any path,

which extends a path in W and sees a path fragment in R infinitely often, satisfies the LTL formula. Using these sets, a mechanism of validating witnesses is proposed as an interactive game between with the model checker and a user [SVV09]. In [BCC+15] a witness is defined to be a scheduler which satisfies the desired property, and the paper proposes strategies to represent such schedulers concisely, and to compute them using learning algorithms.

1.2 Outline and contributions

The topic of this thesis is the generation of certificates and witnesses for probabilistic model checking, in particular for *probabilistic reachability constraints* in Markov decision processes and probabilistic timed automata. A probabilistic reachability constraint is a bound (lower or upper, strict or non-strict) on the optimal (that is, maximal or minimal) probability of reaching a dedicated state from the initial state of the system. Our emphasis is on defining appropriate notions of explications for these properties and describing algorithms to compute them precisely and heuristically with respect to different optimization criteria. Furthermore, we study the complexity of the associated decision problems. We now give an overview of the structure of the thesis, and mention specific contributions made in each chapter.

In Chapter 2 we introduce our notation and the models that we work with throughout the thesis. Additionally, standard results from the literature which will be used are presented using our notations.

Farkas certificates. Chapter 3 is concerned with *certifying algorithms* for model checking of probabilistic reachability constraints in MDPs. The main contribution is the following.

Contribution 1. **We establish certificate conditions and certifying algorithms for all types of probabilistic reachability constraints in MDPs.**

The introduced certificates are named *Farkas certificates*, as they are derived using *Farkas' Lemma* [Far02]. This is a standard result in linear algebra which provides for each system of linear inequalities Γ another system Γ' such that Γ is unsatisfiable if and only if Γ' is satisfiable. Hence, solutions of Γ' are *certificates* for the unsatisfiability of Γ.

Our starting point is the well-known characterization of optimal reachability probabilities in MDPs using *linear programming* [Kal83]. We observe that this characterization yields certificates for those probabilistic reachability constraints which state that *all schedulers* satisfy a given bound. These certificates are solutions of certain systems of linear inequalities. To derive certificate conditions for the remaining cases (which state that *some scheduler* satisfies a threshold condition), we apply variants of Farkas' lemma.

Then, we study how Farkas certificates which certify *the existence* of a scheduler satisfying a threshold condition can be transformed into a witnessing scheduler, and vice versa. An important insight here is that these Farkas certificates are related to the *expected number of visits* of state-action pairs in the MDP under some scheduler.

We consider the general case of MDPs with *proper end components*, which are parts of the MDP in which a scheduler may remain forever[1]. To construct the systems of linear inequalities defining Farkas certificates for bounds on the minimal reachability probabilities, one has to

[1] Usually, we consider MDPs with dedicated absorbing states "target" and "exit", and we distinguish whether the MDP has proper end components *apart from these two*.

compute the proper end components beforehand.

Contribution 2. **We describe a method to certify the result of algorithms for the maximal end component decomposition**.

As discussed above, some of the proposed certificate conditions depend on knowing which state is included in a proper end component. Hence, the certificate condition is only correct if these states have been computed correctly. We introduce techniques to certify that a given set of sub-MDPs indeed equals the maximal end components of the MDP. These techniques are based on certificates for strong connectedness of directed graphs, and a Farkas-like certificate which certifies that the corresponding quotient MDP has no proper end components.

Finally, we show that Farkas certificates for threshold constraints on the *expected total reward* can be defined in a very similar fashion to those for probabilistic reachability constraints. Here we allow arbitrary integer rewards, but restrict the MDPs to have no proper end components apart from one absorbing state, in which the reward is collected.

New techniques for witnessing subsystems. Chapter 4 considers the notion of witnessing subsystem, as introduced in [JÁK+11] (where they were called *critical subsystems*). In contrast to other works on subsystems for MDPs, we define them both for lower bounds on the *maximal* and *minimal* reachability probabilities. Technically, the same definition of witnessing subsystem can be used for both properties. The definition must ensure that the *set of enabled actions* remains the same for all states in subsystems, however, which is not the case if one only considers lower bounds on the maximal probability.

Our first contribution on witnessing subsystems is to show that the (associated decision-) problem of computing *minimal* witnessing subsystems is NP-complete for acyclic Markov chains. While membership in NP is clear (this holds for all related problems that we will consider), NP-hardness for Markov chains was left as an open problem in [WJÁ+12, WJÁ+14]. NP-hardness for the full class of MDPs was established in [CV10, WJÁ+14].

Contribution 3. **Computing minimal witnessing subsystems for lower-bounded probabilistic reachability constraints in acyclic Markov chains is NP-complete**.

We use this result to show NP-hardness of finding a minimal *ϵ-core* [KM20] in Markov chains. An ϵ-core is a kind of subsystem, with the property that the maximal probability of ever leaving it is at most ϵ. Finding minimal ϵ-cores was shown to be NP-hard for MDPs, but it was left open whether the same holds for the subclass of Markov chains [KM20, Remark 3.7]. The main contribution of Chapter 4 is the following.

Contribution 4. **We show that Farkas certificates and witnessing subsystems for the same property are strongly related.**

More precisely, the *support* (i.e., the non-zero entries) of a Farkas certificate induces a witnessing subsystem for the corresponding property, and, vice versa, for every witnessing subsystem we find a Farkas certificate whose support corresponds to the states (or state-action pairs) included in the subsystem. Moreover, a Farkas certificate also certifies that the corresponding subsystem is a witness.

This insight yields new algorithms to compute minimal and small witnessing subsystems. We describe *mixed-integer linear programs* (MILPs) whose solutions correspond to minimal-support Farkas certificates, both for lower-bounded threshold constraints on the minimal and

maximal reachability probabilities. These results carry over to the case of expected total reward properties, and, with minor extensions, can also be used to find *label-minimal* and *weight-minimal* witnesses. Furthermore, we show how witnessing subsystems for lower bounds on the optimal probability of satisfying an *invariance condition* in MDPs can be computed using methods based on Farkas certificates. Such properties correspond to probabilistic reachability constraints with upper-bounded threshold conditions.

Contribution 5. **We introduce the quotient-sum heuristic which computes small witnessing subsystems by iteratively solving a sequence of LPs**.

The *k-step quotient-sum* heuristic computes Farkas certificates with small support by solving a sequence of *linear programs* (LPs), where the solution of the i-th LP is used to define the objective function of the i+1-th LP. In principle, this heuristic can be applied to compute vectors with small support in any polytope included in the nonnegative orthant. By applying the relation between Farkas certificates and witnessing subsystems, the heuristic can be used to compute small witnessing subsystems.

Experimental studies show that this heuristic is competitive with state-of-the-art methods to generate witnessing subsystems. Usually, it finds small witnesses after few (typically between two and three) iterations, which means that the same number of LPs have to be solved. Hence, the overhead of running the quotient-sum heuristic as opposed to computing the optimal reachability probabilities (which can be done by solving a single LP) is a small constant multiplicative factor. The heuristic can also be applied to find witnesses with few labels.

We want to emphasize that all the considered algorithms (including both heuristic and exact approaches) produce Farkas certificates, which means that they are *certifying*. This is not obvious, as to check whether a subsystem is indeed witnessing is, in general, as hard as checking whether the corresponding constraint holds in the original system. On the other hand, to check the validity of a Farkas certificate one merely has to verify that the certificate (which is a vector) is a solution of a linear system of inequalities. This can be done in linear time.

Contribution 6. **An implementation of most presented algorithms on Farkas certificates and witnessing subsystems in the tool Switss**.

All experiments we report on were conducted using the tool Switss, which implements most of the algorithms described in this thesis. It is written in python and uses modern mathematical optimization solvers in the back end to solve (MI)LPs. Furthermore, it includes modules to compute and validate Farkas certificates, visualize subsystems and methods supporting the automated execution and evaluation of experiments. Many of the examples described in this thesis have been implemented in a Jupyter notebook[1] powered by Switss [Jan22a]. With it, one can validate the calculations made in the examples and experiment with them by changing the models or parameters that were used.

Probabilistic systems with low tree width. In Chapter 5 we study probabilistic systems with *low tree width*. The tree width of a graph is a number which, intuitively, quantifies how close the graph is to being a tree. Considering classes of graphs with bounded tree width is a widely used restriction in graph theory and often makes computational problems easier to solve [Bod97]. In particular, a number of NP-complete problems on graphs become tractable if the tree width is considered to be constant (one example is three-colorability). These

[1] `https://jupyter.org/`

results rely on *dynamic programming* techniques and make use of the underlying tree structure. Several works have considered algorithmic questions under restrictions on the tree width for probabilistic systems [CŁ13, CIP15, ACG+20].

We say that a Markov chain is *tree structured* if its underlying graph is a tree. An indication that restrictions on the tree width could be useful for the problem of computing minimal witnessing subsystems is the following result.

Contribution 7. **Computing witnessing subsystems with minimal weight can be done in pseudo-polynomial time in tree structured Markov chains.**

In the above statement, the size of a subsystem is taken to be the sum of weights of the included states, for some predefined weight function. It follows that the problem of computing witnessing subsystems with a minimal number of states is solvable in polynomial time for tree structured Markov chains. However, we also show that the problem becomes NP-hard for tree structured Markov chains in the binary representation of weights and if the goal is to find label-minimal witnessing subsystems.

Inspired by the above result, we go on to study whether suitable restrictions on the tree width of the underlying graph of Markov chains can be exploited to compute minimal witnessing subsystems. To this end, we introduce a novel notion of tree width for directed graphs.

Contribution 8. **We introduce the directed tree- and path-partition width for directed graphs.**

The directed tree-partition width of a directed graph is at most K if its vertices can be partitioned into sets of size at most K such that the induced quotient under the partition is a (directed) tree. For directed path-partition width the definition is analogous. We show that deciding whether a graph has directed path- (or tree-) partition width at most K is NP-complete. This notion of tree width is rather strong, in the sense that classes of bounded directed tree-partition width have bounded width with respect to all notions of tree width that we are aware of in the literature. The main contribution of Chapter 5 is the proof of the following result.

Contribution 9. **Computing minimal witnessing subsystems is NP-complete in Markov chains with bounded directed path-partition width.**

Hence, computing minimal witnesses is NP-hard even for classes of Markov chains which are (asymptotically) very close to paths. The proof is fundamentally different from the other NP-hardness proofs in the thesis, and exhibits a new source of hardness in the problem of computing minimal witnesses. To prove it, we introduce an auxiliary matrix problem, called the *d-dimensional matrix-pair chain problem*. It takes as input a starting vector and n pairs of $d \times d$ matrices, for some fixed d. The question is whether one can reach a specified halfspace in n steps, where in the i-th step one can multiply one of the two matrices of the i-th pair to the current vector. In a chain of polynomial reductions, we first show NP-hardness for the above problem with $d = 2$ and arbitrary matrices and then for $d = 3$ and nonnegative matrices. Finally, this problem is reduced to the problem of computing minimal witnesses in Markov chains with path-partition width at most six.

This means that, unfortunately, we cannot hope to find algorithms that compute minimal witnesses in polynomial time in arbitrary Markov chains with bounded path or tree width. Nevertheless, in Chapter 5 we also develop an algorithm which utilizes the tree structure of a probabilistic system to find a witnessing subsystem. In an experimental study we show that it outperforms the MILP-based approach in certain benchmarks. An example where such tree

structure appears is if the system includes a counter which is never decremented. Then, the possible values of the counter induce a path partition of the state space.

Explications for probabilistic timed automata. Chapter 6 considers explications for probabilistic reachability constraints in PTA. The definition of *subsystems* along with timing-sensitive notions of size for subsystems are the main contributions of this chapter.

Contribution 10. **Timing-sensitive notions of minimal witnessing subsystems for probabilistic timed automata.**

The types of minimality for witnesses we consider are *location-minimality*, *invariant-minimality* and *volume-minimality*. Whereas location-minimality counts the number of participating locations and hence corresponds to state-minimality for MDPs, the other two notions take the timing constraints into consideration. Invariant-minimality measures the logical strength of the location invariants, which are given as clock constraints. If no location invariant can be strengthened without breaking the property, the witnessing subsystem is called inv-minimal. As many subsystems may be incomparable with respect to the inv-order, we introduce volume-minimality, which measures the combined volume of location invariants viewed as polytopes.

We describe algorithms for computing minimal witnessing subsystems under these three notions. They are based on a relation between minimal witnesses of PTA and label-minimal MDP-subsystems in quotient-MDPs induced by *probabilistic time-abstracting bisimulations* of the PTA. Hence, the algorithms and heuristics described in Chapter 4 for computing label-minimal witnessing subsystems for MDPs can be applied here.

Related publications

This thesis is mainly based on the following peer-reviewed publications.

[FJB20] Florian Funke, Simon Jantsch, and Christel Baier. Farkas Certificates and Minimal Witnesses for Probabilistic Reachability Constraints. In *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference* (TACAS), Lecture Notes in Computer Science, pages 324-245. 2020.

[JHFB20] Simon Jantsch, Hans Harder, Florian Funke, and Christel Baier. SWITSS: Computing Small Witnessing Subsystems. In *Proceedings of the 20th Conference on Formal Methods in Computer-Aided Design* (FMCAD), Tu Wien Academic Press, pages 236-244. 2020.

[JFB20] Simon Jantsch, Florian Funke, and Christel Baier. Minimal Witnesses for Probabilistic Timed Automata. In *Automated Technology for Verification and Analysis - 18th International Symposium* (ATVA), Lecture Notes in Computer Science, pages 501-517. 2020.

[JPB21] Simon Jantsch, Jakob Piribauer, and Christel Baier. Witnessing Subsystems for Probabilistic Systems with Low Tree Width. In *Proceedings of the 12th International Symposium on Games, Automata, Logics, and Formal Verification* (GandALF), Electronic Proceedings in Theoretical Computer Science, pages 35-51. 2021.

Further, the thesis extends the work published in these papers by

- considering MDPs with proper end components throughout,
- providing details on the relationship between Farkas certificates and the expected number of visits of state-action pairs under certain schedulers of an MDP,
- considering certification of algorithms that compute the maximal end components,
- transferring the results on Farkas certificates and witnessing subsystems to threshold constraints on the expected total reward,
- considering witnessing subsystems for lower-bounded threshold constraints on the probability of satisfying an invariant property, and
- generalizing many results to the computation of label- and weight-minimal witnessing subsystems.

The relation to published work is stated more precisely in the beginning of each chapter.

Chapter 2

Preliminaries

This chapter collects all preliminary definitions that are needed in the sequel, introduces the notions that are used and states standard results from the literature.

2.1 Linear algebra and linear programming

Vectors of the form $\mathbf{x} \in \mathbb{R}^n$ are written in lowercase and boldface, matrices $\mathbf{M} \in \mathbb{R}^{m \times n}$ are written in uppercase and boldface, and scalars $a \in \mathbb{R}$ are written in lowercase. When working with vectors we will often use finite index sets I and write $\mathbf{x} \in \mathbb{R}^I$ instead of $\mathbf{x} \in \mathbb{R}^{|I|}$. To represent a vector $\mathbf{x} \in \mathbb{R}^I$ we will sometimes use the notation

$$\mathbf{x} = \big(i_1 \mapsto a_1,\ i_2 \mapsto a_2,\ \ldots,\ i_n \mapsto a_n \big),$$

with $I = \{i_1, \ldots, i_n\}$. In this case $\mathbf{x}(i_k)$ denotes the corresponding entry a_k of $\mathbf{x}$. The *support* of a vector $\mathbf{x} \in \mathbb{R}^I$ is the set $\operatorname{supp}(\mathbf{x}) = \{i \in I \mid \mathbf{x}(i) \neq 0\}$. For a given $i \in I$ we define the *dirac vector* $\delta_i \in \mathbb{R}^I$ to be $\delta_i(i) = 1$ and $\delta_i(i') = 0$ for all $i' \in I \setminus \{i\}$. The vectors $\mathbf{0}^n$ and $\mathbf{1}^n$ denote the constant zero and one vectors of dimension n, and if the dimension is clear from the context we simply write $\mathbf{0}$ and $\mathbf{1}$. Let $\mathbf{M} \in \mathbb{R}^{I_1 \times I_2}$ be a matrix of dimension $|I_1| \times |I_2|$, where again I_1, I_2 are used as index sets. For any $I_1' \subseteq I_1$ and $I_2' \subseteq I_2$, we define the *restriction* of $\mathbf{M}$ to $I_1' \times I_2'$, denoted by $\mathbf{M}|_{I_1' \times I_2'} \in \mathbb{R}^{I_1' \times I_2'}$, to be: $\mathbf{M}|_{I_1' \times I_2'}(a_1, a_2) = \mathbf{M}(a_1, a_2)$ for all $(a_1, a_2) \in I_1' \times I_2'$. The restriction of vectors is defined analogously.

Polyhedra and linear inequalities. A *halfspace* in $\mathbb{R}^n$ is a set $\{\mathbf{v} \in \mathbb{R}^n \mid \mathbf{a} \cdot \mathbf{v} \leq b\}$ for some $\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}$ such that $\mathbf{a} \neq \mathbf{0}$ and $b \neq 0$. The intersection of finitely many halfspaces is called a *polyhedron*, and a *polytope* is a bounded polyhedron. A *face* of a polyhedron P is a subset $F \subseteq P$ of the form $F = \{\mathbf{x} \in P \mid \mathbf{a} \cdot \mathbf{x} = \max\{\mathbf{a} \cdot \mathbf{y} \mid \mathbf{y} \in P\}\}$ for some $\mathbf{a} \in \mathbb{R}^n$, i.e., it is the "boundary" of P in some direction. Faces consisting of only one point are called *vertices*.

Vectors of *variables* will be used to define (systems of) linear inequalities. If $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector of variables of dimension n, then $\mathbf{a} \cdot \mathbf{x} \leq b$, where $\mathbf{a} \in \mathbb{R}^n, b \in \mathbb{R}$, represents the *linear inequality* $\sum_{1 \leq i \leq n} a_i x_i \leq b$. Similarly, a *system of linear inequalities* can be represented using a

matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and vector $\mathbf{b} \in \mathbb{R}^m$ by writing $\mathbf{M} \cdot \mathbf{x} \leq \mathbf{b}$. This system consists of m linear inequalities, which are formed using the m rows of $\mathbf{M}$ together with the corresponding entries of $\mathbf{b}$. The set of solutions of a linear inequality forms a halfspace, and the set of solutions of a system of linear inequalities forms a polyhedron. We will often use the same name for a variable vector $\mathbf{x} = (x_1, \ldots, x_n)$ and for some concrete solution $\mathbf{x} \in \mathbb{R}^n$ of a system of linear inequalities.

(Mixed-integer) linear programming. A *linear program* (LP) is formed by a system of linear inequalities and a linear optimization function. If $\mathbf{x}$ is a variable vector of dimension n, $\mathbf{o} \in \mathbb{R}^n, \mathbf{M} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, then the corresponding linear program is written as: *maximize* $\mathbf{o} \cdot \mathbf{x}$ *such that* $\mathbf{M}\mathbf{x} \leq \mathbf{b}$. We say that $\mathbf{x} \in \mathbb{R}^n$ is a *feasible solution* of this LP if it is a solution of $\mathbf{M} \cdot \mathbf{x} \leq \mathbf{b}$ and it is an *optimal solution* if $\mathbf{o} \cdot \mathbf{x} = \max\{\mathbf{o} \cdot \mathbf{x}' \mid \mathbf{x}' \in \mathbb{R}^n, \mathbf{M}\mathbf{x}' \leq \mathbf{b}\}$. If this maximum does not exist, we say that the LP is *unbounded*. In this case, no optimal solution exists.

Solving a system of linear inequalities is a special case of computing the optimal solutions of a linear program (by setting $\mathbf{o} = \mathbf{0}$). Furthermore, given a linear program one can compute in linear time a system of linear inequalities such that the optimal solutions of the former correspond to the solutions of the latter [Sch99, Theorem 10.4]. Both problems are hence very closely connected, and it turns out that both can be solved in polynomial time [Kha79, GL81], see also [Sch99, Chapter 13].

A *mixed-integer linear program* is a linear program in which a subset of the variables are declared to be integer- (or binary-) variables. That is, the feasible solutions are restricted to those in which the specified variables take integer values. In contrast to standard linear programming, solving mixed-integer linear programs is NP-complete [Sch99, Theorem 18.1].

Farkas' Lemma. Farkas' Lemma [Far02] is a fundamental *duality theorem* in polyhedra theory and linear programming. It shows that for each system of linear inequalities one can compute a dual system such that the former is satisfiable if and only if the latter is unsatisfiable. Hence, a solution of one system certifies the unsatisfiability of the other, and vice versa. We will use it in the following two versions.

Lemma 2.1 (Farkas' Lemma, cf. [Sch99, Corollary 7.1e]). *Let* $\mathbf{M} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} \in \mathbb{R}^m$. *Then there exists* $\mathbf{z} \in \mathbb{R}^n$ *with* $\mathbf{M}\mathbf{z} \leq \mathbf{b}$ *if and only if there does* not *exist* $\mathbf{y} \in \mathbb{R}^m_{\geq 0}$ *with* $\mathbf{y}\mathbf{M} = \mathbf{0} \wedge \mathbf{y}\mathbf{b} < 0$.

Lemma 2.2 (Farkas' Lemma (variant), cf. [Sch99, Corollary 7.1f]). *Let* $\mathbf{M} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} \in \mathbb{R}^m$. *Then there exists* $\mathbf{z} \in \mathbb{R}^n_{\geq 0}$ *with* $\mathbf{M}\mathbf{z} \leq \mathbf{b}$ *if and only if there does* not *exist* $\mathbf{y} \in \mathbb{R}^m_{\geq 0}$ *with* $\mathbf{y}\mathbf{M} \geq \mathbf{0} \wedge \mathbf{y}\mathbf{b} < 0$.

2.2 Markov decision processes

The most important model which is considered in this thesis are *Markov decision processes*. They are used to analyze probabilistic systems which appear in a wide range of contexts, from operations research over biological models to verification. We introduce our notation along with standard results which we will use throughout the thesis. For more details, we refer to [Put94, BK08].

2.2.1 Definitions

Definition 2.3. A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (S, \text{Act}, \iota, P)$, where

- S is a countable set of *states*,
- Act is a finite set of *actions*,
- $\iota : S \to [0,1]$ is the *initial distribution* where we require $\sum_{s \in S} \iota(s) = 1$,
- $P \colon S \times \text{Act} \times S \to [0,1]$ is the *transition probability function*, satisfying $\sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$ for all $s \in S$ and $\alpha \in \text{Act}$.

When considering properties based on *rewards*, we extend the tuple by

- a *reward function* $\text{rew} : S \times \text{Act} \to \mathbb{Z}$.

We say that action $\alpha \in \text{Act}$ is *enabled* in state $s \in S$ if $\sum_{s' \in S} P(s, \alpha, s') = 1$, and denote by $\text{Act}(s)$ the set of such actions. The set of *enabled state-action pairs* is defined to be $\mathcal{E} = \{(s, \alpha) \in S \times \text{Act} \mid \alpha \in \text{Act}(s)\}$. We require that $\text{Act}(s) \neq \emptyset$ for all $s \in S$.

Size of an MDP. In this thesis we will almost exclusively consider *finite* MDPs, in which the set of states S is finite. If not stated otherwise, we will assume MDPs to be finite. For all algorithmic questions we additionally assume that all numbers in the ranges of ι and P are rational. Under these assumptions we define the size of an MDP to be the sum of the cardinalities of S and Act and the lengths of the binary encoding of all numbers in the ranges of ι and P.

The initial state, finite and infinite paths. If the initial distribution is a Dirac distribution on a single state $s_{in} \in S$, then we may also write $\mathcal{M} = (S, \text{Act}, s_{in}, P)$.

An *infinite path* of $\mathcal{M}$ is an infinite sequence $s_1\alpha_1 s_2\alpha_2 \ldots \in (S \times \text{Act})^\omega$ such that $P(s_i, \alpha_i, s_{i+1}) > 0$ for all $i \geq 1$. A *finite path* is a finite sequence $s_1\alpha_1 s_2\alpha_2 \ldots s_n \in (S \times \text{Act})^* S$ satisfying $P(s_i, \alpha_i, s_{i+1}) > 0$ for all $1 \leq i < n$. If π is a finite path, we define $\text{last}(\pi) = s_n$. The set of infinite paths of $\mathcal{M}$ is denoted by $\text{Paths}(\mathcal{M})$, and the set of finite paths is denoted by $\text{Paths}_{\text{fin}}(\mathcal{M})$. Often we will use the set of (in)finite paths restricted to a certain starting state $s \in S$, defined as follows. The set $\text{Paths}(\mathcal{M}, s) = \{s_1\alpha_1 s_2\alpha_2 \ldots \in \text{Paths}(\mathcal{M}) \mid s_1 = s\}$ includes all paths of $\mathcal{M}$ starting in s, and the analogous notation is used finite paths. The length of a path is defined by $\text{len}(s_1\alpha_1 \ldots s_n) = n$ for finite paths, and $\text{len}(\pi) = \infty$ if π is an infinite path. We call a state $s \in S$ *absorbing* if $P(s, \alpha, s) = 1$ for all $\alpha \in \text{Act}(s)$.

When defining sets of paths, we will use notations borrowed from linear temporal logic (LTL). In particular, given a set of states $R \subseteq S$, we let $\Diamond R$ denote the set $\{s_1\alpha_1 s_2\alpha_2 \ldots \in \text{Paths}(\mathcal{M}) \mid s_i \in R \text{ for some } i \geq 1\}$ and $\Box R$ denote the set $\{s_1\alpha_1 s_2\alpha_2 \ldots \in \text{Paths}(\mathcal{M}) \mid s_i \in R \text{ for all } i \geq 1\}$. If $R = \{s\}$ is a singleton, we may write $\Diamond s$ or $\Box s$, and we will also use the notations $\overline{R} = S \setminus R$, $\neg R = \overline{R}$ and $\neg s = \overline{\{s\}}$ for $s \in S$.

Markov chains and the underlying probability measure. A *(discrete-time) Markov chain* (DTMC) is an MDP in which exactly one action is enabled in every state. For DTMCs we will omit the action set in the defining tuple and write $\mathcal{M} = (S, \iota, P)$. In this case the probability transition function is of type $P : S \times S \to [0,1]$ and is required to satisfy $\sum_{s' \in S} P(s, s') = 1$ for all $s \in S$. Consequently, finite paths of the DTMC $\mathcal{M}$ are sequences $s_1 s_2 \ldots s_n \in S^*$ such that $P(s_i, s_{i+1}) > 0$ for all $1 \leq i \leq n$, and analogously for infinite paths.

If $\mathcal{M}$ is a DTMC, then $\mathrm{Paths}(\mathcal{M})$ carries a probability measure. Its associated σ-algebra is generated by the cylinder sets $\mathrm{Cyl}(\tau) = \{\pi \in \mathrm{Paths}(\mathcal{M}) \mid \pi \text{ has prefix } \tau\}$, where τ is a finite path of $\mathcal{M}$. If $\tau = s_1 s_2 \dots s_n$, then the probability of $\mathrm{Cyl}(\tau)$ is given by

$$\Pr(\mathrm{Cyl}(\tau)) = \iota(s_1) \cdot P(s_1, s_2) \cdot P(s_2, s_3) \cdot \ldots \cdot P(s_{n-1}, s_n).$$

For more details, see [BK08, Section 10.1].

We denote by $\Pr_{\mathcal{M}}(\Pi)$ the probability of a measurable set $\Pi \subseteq \mathrm{Paths}(\mathcal{M})$. If we are interested in the probability of an event when starting in some starting state $s \in S$ of $\mathcal{M}$, we consider the probability measure as defined above for the DTMC $\mathcal{M}_s = (S, s, P)$ with unique initial state s. This DTMC corresponds to $\mathcal{M}$ except for the initial distribution, which is now a Dirac distribution on s. We will often write $\Pr_{\mathcal{M},s}(\Pi)$ instead of $\Pr_{\mathcal{M}_s}(\Pi)$. In the following we denote for a finite set X the set of probability distributions on X by $\mathrm{Dist}(X)$.

Remark 2.4 (Measurability of events). Measurability of all ω-regular sets (these are the sets definable by a nondeterministic Büchi automaton) of paths in the described σ-algebra can be proved, and hence this holds in particular for reachability properties and all properties definable in linear temporal logic. Hence, we will not be concerned further with the measurability of events in the thesis and refer to [BK08, Chapter 10] for further details. △

Structural properties of Markov decision processes. The *underlying graph* of MDP $\mathcal{M} = (S, \mathrm{Act}, \iota, P)$ is the directed graph $\mathcal{U}(\mathcal{M}) = (S, E)$ with vertices S and edges defined by: $(s_1, s_2) \in E$ if and only if there exists $\alpha \in \mathrm{Act}(s_1)$ such that $P(s_1, \alpha, s_2) > 0$. Let $\mathcal{U} = (S, E)$ be the underlying graph of a finite DTMC $\mathcal{M} = (S, \iota, P)$ and $S_1, \dots, S_n$ be the strongly connected components (SCCs) of $\mathcal{U}$. If all states contained in an SCC S_i have only edges to other states in S_i, we say that S_i is a *bottom strongly connected component* (BSCC). We call states which belong to a BSCC *recurrent*, and all others *transient*. In case $\mathcal{M}$ consists of only one SCC, we call $\mathcal{M}$ *irreducible*.

A *sub-MDP* of $\mathcal{M}$ is a pair (E, A), where $E \subseteq S$ is a set of states and $A : E \to 2^{\mathrm{Act}}$ is a function such that $\{t \in S \mid P(s, \alpha, t) > 0\} \subseteq E$ holds for all $s \in E, \alpha \in A(s)$. The *induced graph* of a sub-MDP (E, A) is the directed graph with vertices E and edges

$$\{(s, t) \in E \times E \mid \text{there exists } \alpha \in A(s) \text{ such that } P(s, \alpha, t) > 0\}.$$

An *end component* of $\mathcal{M}$ is a sub-MDP whose induced graph is strongly connected. We say that an end component (E, A) is *maximal* if there is no other end component (E', A') such that $E \subseteq E'$ and $A(s) \subseteq A'(s)$ for all $s \in E$. The end component (E, A) is called *proper* if there exists $s \in E$ such that $A(s) \neq \emptyset$. If (E, A) is not proper, then it can only contain a single state. Each state s is included in a unique (possibly non-proper) maximal end component.

Schedulers. A *scheduler* for $\mathcal{M}$ is a function $\mathfrak{S} : \mathrm{Paths}_{\mathrm{fin}}(\mathcal{M}) \to \mathrm{Dist}(\mathrm{Act})$ assigning to each finite path π of $\mathcal{M}$ a probability distribution over the actions, where we require that $\mathrm{supp}(\mathfrak{S}(\pi)) \subseteq \mathrm{Act}(\mathrm{last}(\pi))$ holds for each finite path π. In other words, $\mathfrak{S}$ is only allowed to assign non-zero values to actions that are enabled in the last state of π. If the chosen probability distribution is always a Dirac distribution we call the scheduler *deterministic*, and otherwise *randomized*. Given a scheduler $\mathfrak{S}$ for $\mathcal{M}$ and a path (finite or infinite) $\pi = s_1 \alpha_1 s_2 \alpha_2 \dots$ of $\mathcal{M}$ we say that π is an $\mathfrak{S}$-path if $\alpha_i \in \mathrm{supp}(\mathfrak{S}(s_1 \alpha_1 \dots s_i))$ for all $1 \leq i < \mathrm{len}(\pi)$. The scheduler

$\mathfrak{S}$ is called *memoryless* if for all finite paths π we have $\mathfrak{S}(\pi) = \mathfrak{S}(\text{last}(\pi))$. In this case the decision of $\mathfrak{S}$ always depends only on the last state of the given finite path. A scheduler is called *finite-memory* if it can be realized by a finite state machine. Memoryless and deterministic schedulers will be abbreviated as MD-schedulers, and memoryless randomized schedulers will be called MR-schedulers.

A scheduler $\mathfrak{S}$ for $\mathcal{M}$ induces an infinite Markov chain $\mathcal{M}_{\mathfrak{S}} = (\text{Paths}_{\text{fin}}(\mathcal{M}), \iota, P_{\mathfrak{S}})$ defined as follows. For all $\pi, \pi' \in \text{Paths}_{\text{fin}}(\mathcal{M})$ we have:

$$P_{\mathfrak{S}}(\pi, \pi') = \begin{cases} \mathfrak{S}(\pi, \alpha) \cdot P(s, \alpha, s') & \text{if } \text{last}(\pi) = s \text{ and } \pi' = \pi\alpha s', \\ 0 & \text{otherwise.} \end{cases}$$

This construction can be used as a basis to associate probabilities to events in the MDP $\mathcal{M}$ under a given scheduler $\mathfrak{S}$. The set of infinite paths of $\mathcal{M}_{\mathfrak{S}}$ is in bijection with the set of infinite $\mathfrak{S}$-paths of $\mathcal{M}$. For a set of infinite $\mathfrak{S}$-paths $\Pi \subseteq \text{Paths}(\mathcal{M})$, let Π' be the set of corresponding infinite paths of $\mathcal{M}_{\mathfrak{S}}$. Then, we define the probability $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Pi)$ of Π under scheduler $\mathfrak{S}$ as

$$\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Pi) = \Pr_{\mathcal{M}_{\mathfrak{S}}}(\Pi'),$$

provided that Π' is measurable in the probability measure of $\mathcal{M}_{\mathfrak{S}}$. As for DTMCs, we will also consider the probabilities when starting in some state $s \in S$, denoted by $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Pi)$.

If $\mathfrak{S}$ is a memoryless scheduler we may use a simpler and finite (provided that $\mathcal{M}$ is finite) induced Markov chain $\mathcal{M}'_{\mathfrak{S}} = (S, \iota, P'_{\mathfrak{S}})$ where $P'_{\mathfrak{S}}(s,t) = \sum_{\alpha \in \text{Act}(s)} \mathfrak{S}(s,\alpha) \cdot P(s,\alpha,t)$ for all $s, t \in S$. As the behavior of all states πs of $\mathcal{M}_{\mathfrak{S}}$ that agree on their last state s is identical to the behavior of state s in $\mathcal{M}'_{\mathfrak{S}}$, the probability measure of any path property in $\mathcal{M}'_{\mathfrak{S}}$ and (the corresponding path property in) $\mathcal{M}_{\mathfrak{S}}$ agree. When considering memoryless schedulers $\mathfrak{S}$ we will always refer to $\mathcal{M}'_{\mathfrak{S}}$ as *the induced Markov chain* and simply write $\mathcal{M}_{\mathfrak{S}}$.

When considering MDPs one is often interested in *best-case* or *worst-case* outcomes when ranging over the possible resolutions of the nondeterminism. The nondeterminism is resolved by a scheduler, and hence the task corresponds to finding schedulers which maximize or minimize the value of some outcome. For a given ω-regular set $\Pi \subseteq \text{Paths}(\mathcal{M})$ we define

$$\mathbf{Pr}^{\max}_{\mathcal{M}}(\Pi) = \sup_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Pi) \qquad \text{and} \qquad \mathbf{Pr}^{\min}_{\mathcal{M}}(\Pi) = \inf_{\mathfrak{S}} \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Pi),$$

where $\mathfrak{S}$ ranges over all schedulers of $\mathcal{M}$. The notation max and min is supported by the classical result that for ω-regular properties there exist (even finite-memory) schedulers which achieve the supremum, respectively infimum [Var85]. As before, if we are interested in the optimal probabilities from some state $s \in S$, we write $\mathbf{Pr}^{\max}_{\mathcal{M},s}(\Pi)$ and $\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Pi)$. If $\mathcal{M}$ is clear from the context we will write $\mathbf{Pr}^{\max}(\Pi)$ and $\mathbf{Pr}^{\min}(\Pi)$.

2.2.2 Reachability probabilities

For an MDP $\mathcal{M} = (S, \text{Act}, \iota, P)$ and a set of states $T \subseteq S$ we denote the set of infinite paths of $\mathcal{M}$ which reach T by $\Diamond T$. The vectors $\left(\mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond T)\right)_{s \in S}$ and $\left(\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond T)\right)_{s \in S}$ containing the optimal reachability probabilities in each state can be characterized as the solutions of certain linear programs, and hence computed in polynomial time [CY90, BdeA95, deA97]. As these linear programs and the underlying systems of linear inequalities will be used extensively throughout the thesis, we will now explain how they are constructed.

Linear program for maximal reachability probabilities. To compute the maximal reachability probabilities for target set T we first compute the set of states $S_{\max=0} \subseteq S$ whose maximal probability to reach T is zero. This holds for state s exactly if there is no path from s to any state in T in the underlying graph of $\mathcal{M}$. Hence, $S_{\max=0}$ can be computed in linear time using standard graph algorithms. For each remaining state s in $S_? = S \setminus S_{\max=0}$ we introduce a variable x_s and consider the linear program: *minimize* $\sum_{s \in S_?} x_s$ *such that*

$$\begin{aligned} x_s &\geq \sum_{s' \in S_?} P(s, \alpha, s') \cdot x_{s'} && \text{for all} \quad s \in S_? \setminus T,\ \alpha \in \mathrm{Act}(s), \\ x_s &= 1 && \text{for all} \quad s \in T. \end{aligned}$$

Intuitively, the linear inequalities require that the value in each state s should be at least as high as the value which is achievable by choosing any of the enabled actions. The minimization makes sure that the value in x_s actually corresponds to the value achieved by one of its actions, and is not artificially high. The vector $\left(\mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond T)\right)_{s \in S_?}$ is the unique optimal solution of this linear program [Kal83, BK08].

Linear program for minimal reachability probabilities. A similar linear program can be given to compute $\left(\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond T)\right)_{s \in S}$. Apart from computing $S_{\max=0}$ we now also compute the set of states whose minimal probability to reach T is zero, which we call $S_{\min=0}$. This set can be computed in linear time using graph algorithms [BK08, Lemma 10.110]. Again, we introduce a variable x_s for all states s in $S_? = S \setminus S_{\max=0}$ and consider the linear program: *maximize* $\sum_{s \in S_?} x_s$ *such that*

$$\begin{aligned} x_s &\leq \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} && \text{for all} \quad s \in S_? \setminus T,\ \alpha \in \mathrm{Act}(s), \\ x_s &= 1 && \text{for all} \quad s \in T, \\ x_s &= 0 && \text{for all} \quad s \in S_{\min=0}. \end{aligned}$$

The vector $\left(\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond T)\right)_{s \in S_?}$ is the unique optimal solution of the above linear program [BK08].

Reachability form. We will now define a standard form which can be used when considering reachability properties in MDPs. It comprises several base assumptions that we will make for MDPs throughout the thesis, and which can always be ensured by a linear time preprocessing. We will assume the existence of a dedicated state called "target", which represents the target set, and a dedicated state called "exit", which represents all states which cannot reach target at all. Hence, once "exit" is reached, the future computation is of no interest for the property of reaching "target". This is formalized in the following definition.

Definition 2.5 (Reachability form). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \mathrm{Act}, \iota, P)$ be an MDP. We say that $\mathcal{M}$ is in *reachability form for* "target" if

- for all $s \in S$ we have $\mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond\{\text{target}\}) > 0$,
- the states "target" and "exit" are absorbing.

When we say that $\mathcal{M}$ is in reachability form we will implicitly assume the existence of states named "target" and "exit" as above. Any MDP $\mathcal{M}$ can be transformed into reachability

form in linear time by first collapsing all target-states of a given reachability objective into a single, absorbing state called "target". Now, the states s satisfying $\mathbf{Pr}_s^{\max}(\Diamond\{\text{target}, \text{exit}\}) = 0$ can be computed in linear time, and their incoming transitions can be redirected to the state "exit". These transformations clearly preserve both $\mathbf{Pr}_s^{\min}(\Diamond\,\text{target})$ and $\mathbf{Pr}_s^{\max}(\Diamond\,\text{target})$ for all remaining states s.

The vectors containing the optimal reachability probabilities will be abbreviated by $\mathbf{pr}^{\max}$ and $\mathbf{pr}^{\min}$. More precisely, we define $\mathbf{pr}_{\mathcal{M}}^{\max} = \big(\mathbf{Pr}_{\mathcal{M},s}^{\max}(\Diamond\,\text{target})\big)_{s\in S}$ and $\mathbf{pr}_{\mathcal{M}}^{\min} = \big(\mathbf{Pr}_{\mathcal{M},s}^{\min}(\Diamond\,\text{target})\big)_{s\in S}$. If the MDP is clear from the context, we will drop the subscript.

Definition 2.6 (End-component-freeness). An MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ is called *free of end components* or *EC-free* if no state in S is included in a proper end component.

Being EC-free is equivalent to satisfying $\mathbf{Pr}_s^{\min}(\Diamond\{\text{target}, \text{exit}\}) = 1$ in each state $s \in S$, and in our setting states "target" and "exit" will always be absorbing.

System matrix and target vector. To work conveniently with the linear programs for optimal reachability probabilities, we now introduce a matrix notation of the underlying systems of inequalities which is tailored for MDPs with dedicated states "target" and "exit". For such MDPs we will consider $\mathcal{E}$ to be the enabled state-action pairs of all states *excluding* these two states.

Definition 2.7 (System matrix and target vector). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, \iota, P)$ be an MDP and $\mathcal{E} \subseteq S \times \text{Act}$ its enabled state-action pairs, excluding those of states "target" and "exit". The matrix $\mathbf{A}_{\mathcal{M}} \in \mathbb{R}^{\mathcal{E}\times S}$, henceforth called the *system matrix* of $\mathcal{M}$, is defined as follows:

$$\mathbf{A}_{\mathcal{M}}((s,\alpha),t) = \begin{cases} 1 - P(s,\alpha,s) & \text{if } s = t, \\ -P(s,\alpha,t) & \text{if } s \neq t. \end{cases}$$

The vector $\mathbf{t}_{\mathcal{M}} = (\mathbf{t}_{\mathcal{M}}(s,\alpha))_{(s,\alpha)\in\mathcal{E}} \in \mathbb{R}^{\mathcal{E}}$, called the *target vector* of $\mathcal{M}$, is defined by: $\mathbf{t}_{\mathcal{M}}(s,\alpha) = P(s,\alpha,\text{target})$. If $\mathcal{M}$ is clear from the context we will omit the subscripts and write $\mathbf{A}$ and $\mathbf{t}$.

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and $\mathbf{A}, \mathbf{t}$ be its system matrix and target-vector. Consider the system of inequalities

$$\mathbf{A}\mathbf{z} \geq \mathbf{t}$$

where $\mathbf{z}$ is a column vector of variables of dimension $|S|$. It contains the following linear inequality for each enabled state-action pair (s,α) :

$$\mathbf{z}(s) \;\geq\; \mathbf{t}(s,\alpha) + \sum_{s'\in S} P(s,\alpha,s') \cdot \mathbf{z}(s').$$

Hence, these inequalities correspond to those of the linear program that characterizes $\mathbf{pr}^{\max}$ (after substituting the variable for the state "target" by one). Here we use that all states in S have positive maximal probability of reaching target, by assumption that $\mathcal{M}$ is in reachability form.

By similar reasoning, the underlying inequalities of the linear program which characterizes the minimal reachability probabilities are $\mathbf{A}\mathbf{z} \leq \mathbf{t}$ together with $\mathbf{z}(s) = 0$ for all $s \in S_{\min=0}$. The following lemma shows that the latter condition is satisfied for all solutions of $\mathbf{A}\mathbf{z} \leq \mathbf{t}$ *if* the MDP is EC-free and in reachability form.

Lemma 2.8. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, \iota, P)$ be an MDP in reachability form and EC-free and let $\mathbf{z} \in \mathbb{R}^S$ satisfy $\mathbf{Az} \leq \mathbf{t}$. Then, for all $s \in S$ such that $\mathbf{Pr}^{\min}_s(\Diamond\,\text{target}) = 0$ we have $\mathbf{z}(s) = 0$.*

Proof. Take any state s satisfying $\mathbf{Pr}^{\min}_s(\Diamond\,\text{target}) = 0$ and let $\mathfrak{S}$ be a deterministic and memoryless scheduler such that $\Pr^{\mathfrak{S}}_s(\Diamond\,\text{target}) = 0$. Such a scheduler exists as optimal reachability probabilities are always attained by some MD-scheduler [BK08]. Let R be the set of states that lie on some $\mathfrak{S}$-path which starts in s, but excluding the state "exit". As $\Pr^{\mathfrak{S}}_s(\Diamond\,\text{target}) = 0$ holds, we have target $\notin R$. Let $\mathcal{E}_R = \{(r, \mathfrak{S}(r)) \mid r \in R\}$ be the actions chosen by $\mathfrak{S}$ in states R, and $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}_R \times R}$ be the restriction of $\mathbf{A}$ to the given dimension. By construction of $\mathbf{A}$, we have $\mathbf{A}' = \mathbf{I} - \mathbf{P}'$ for some substochastic matrix $\mathbf{P}' \in \mathbb{R}^{R \times R}_{\geq 0}$ and the identity matrix $\mathbf{I}$ of matching dimension. The value $(\mathbf{P}')^n(u,t)$ corresponds to the probability of reaching $t \in R$ from $u \in R$ in exactly n steps in $\mathcal{M}$ under $\mathfrak{S}$, for all $n \geq 0$. As $\mathcal{M}$ is EC-free, the probability of eventually reaching the set $\{\text{exit}, \text{target}\}$ from any state is one under any scheduler. This implies that $(\mathbf{P}')^n$ converges to the zero matrix as n goes to infinity, because the states "target" and "exit" are not included in R. Let $\mathbf{z}' = \mathbf{z}|_R$ be the restriction of $\mathbf{z}$ onto R and observe that $\mathbf{t}|_R = \mathbf{0}$, as no $\mathfrak{S}$-path from s reaches target. This implies that we have $\mathbf{z}' \leq \mathbf{P}'\mathbf{z}'$ (by the assumption $\mathbf{Az} \leq \mathbf{t}$) and thus, by induction, $\mathbf{z}' \leq (\mathbf{P}')^n\mathbf{z}'$ for all $n \geq 1$. It follows that $\mathbf{z}'$ is zero in all entries. □

It follows that if $\mathcal{M}$ is in reachability form and EC-free, then any solution of the system of inequalities

$$\mathbf{Az} \leq \mathbf{t}$$

is a feasible solution of the linear program for $\mathbf{pr}^{\min}$, and vice versa.

A well-known property of $\mathbf{Az} \leq \mathbf{t}$ and $\mathbf{Az} \geq \mathbf{t}$ is that their solutions provide point-wise bounds on the vectors $\mathbf{pr}^{\min}$ and $\mathbf{pr}^{\max}$, respectively. We recall the argument here for completeness.

Lemma 2.9. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, \iota, P)$ be an MDP in reachability form. Then, for all $\mathbf{z} \in \mathbb{R}^S$:*

1. *$\mathbf{Az} \geq \mathbf{t}$ implies $\mathbf{z} \geq \mathbf{pr}^{\max}$, and*
2. *if $\mathcal{M}$ is EC-free, then $\mathbf{Az} \leq \mathbf{t}$ implies $\mathbf{z} \leq \mathbf{pr}^{\min}$.*

Proof. We first show (1.) and assume that $\mathbf{Az} \geq \mathbf{t}$ holds. Consider the sequence of vectors $\mathbf{z}_1, \mathbf{z}_2, \ldots$ (all in $\mathbb{R}^S$) generated by setting $\mathbf{z}_1 = \mathbf{z}$ and

$$\mathbf{z}_{i+1}(t) = \max_{\alpha \in \text{Act}(t)} \Big\{\, \mathbf{t}(t,\alpha) \;+\; \sum_{t' \in S} P(t,\alpha,t') \cdot \mathbf{z}_i(t') \,\Big\} \quad \text{for all } t \in S.$$

One can show by induction that $\mathbf{z}_{i+1} \leq \mathbf{z}_i$ and $\mathbf{Az}_i \geq \mathbf{t}$ holds for all $i \geq 1$.

The limit $\lim_{i\to\infty} \mathbf{z}_i$ is a solution of the equation system

$$x_t = \max_{\alpha \in \text{Act}(t)} \Big\{\, \mathbf{t}(t,\alpha) \;+\; \sum_{t' \in S} P(t,\alpha,t') \cdot x_{t'} \,\Big\} \quad \text{for all } t \in S,$$

where there is a variable x_t for each $t \in S$. It is known that $\mathbf{Pr}^{\max}(\Diamond\,\text{target})$ is the least solution of this equation system (see [deA97, Theorem 3.9] and [BK08, Theorem 10.100]). Here we use the assumption that $\mathbf{Pr}^{\max}_t(\Diamond\,\text{target}) > 0$ holds for each $t \in S$ as $\mathcal{M}$ is in reachability form. We can conclude that $\mathbf{z} \geq \mathbf{pr}^{\max}$.

The statement (2.) is shown in essentially the same way. We can apply [BK08, Theorem 10.109] which shows that the corresponding equation system for minimal reachability probabilities has a unique solution, under the assumption that for all states $s \in S$ satisfying $\mathbf{Pr}^{\min}_s(\Diamond \text{target}) = 0$ we have $\mathbf{z}(s) = 0$. But this is guaranteed by Lemma 2.8. □

An important property in the context of reachability probabilities is that optimal probabilities are always attained by some MD-scheduler [BK08, Lemmas 10.102 and 10.113].

Proposition 2.10. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form.*
Then, there exist memoryless and deterministic schedulers $\mathfrak{S}_{\min}$ and $\mathfrak{S}_{\max}$ such that

$$\left(\Pr^{\mathfrak{S}_{\min}}_{\mathcal{M},s}(\Diamond \text{target})\right)_{s\in S} = \mathbf{pr}^{\min} \quad \text{and} \quad \left(\Pr^{\mathfrak{S}_{\max}}_{\mathcal{M},s}(\Diamond \text{target})\right)_{s\in S} = \mathbf{pr}^{\max},$$

and, furthermore, $\mathfrak{S}_{\max}$ additionally satisfies $\Pr^{\mathfrak{S}_{\max}}_{\mathcal{M},s}(\Diamond\{\text{target}, \text{exit}\}) = 1$ for all $s \in S$.

Proof. The fact that optimal memoryless and deterministic schedulers exist follows from [BK08, Lemmas 10.102 and 10.113]. We now argue that when considering maximal reachability probabilities, then every optimal scheduler also reaches $\{\text{target}, \text{exit}\}$ with probability one from each state. Let $\mathfrak{S}$ be a scheduler satisfying $\left(\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond \text{target})\right)_{s\in S} = \mathbf{pr}^{\max}$ and assume, for contradiction, that $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\{\text{target}, \text{exit}\}) < 1$ for some $s \in S$. It follows that there exists an $\mathfrak{S}$-path from s to a state s' such that $\Pr^{\mathfrak{S}}_{\mathcal{M},s'}(\Diamond\{\text{target}, \text{exit}\}) = 0$. As $\mathcal{M}$ is in reachability form it satisfies $\mathbf{Pr}^{\max}_{\mathcal{M},s'}(\Diamond \text{target}) > 0$ and hence also $\mathbf{Pr}^{\max}_{\mathcal{M},s'}(\Diamond\{\text{target}, \text{exit}\}) > 0$. This contradicts the assumption that $\Pr^{\mathfrak{S}}_{\mathcal{M},s'}(\Diamond \text{target}) = \mathbf{pr}^{\max}(s')$. □

Quotient of maximal end components. We now introduce variants of the *quotient of maximal end components* [deA97, CBGK08] in the specific way that we will need them. As before, let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP and assume that states "target" and "exit" are both absorbing. The proper end components induced by these two states will be considered explicitly in the below construction.

Let $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be a set of end components of $\mathcal{M}$ such that $E_1, \ldots, E_k$ forms a partition of S. We define two versions of the quotient of $\mathcal{M}$ by $\mathcal{D}$ which only differ in whether proper end components get a direct transition to "target" or a direct transition to "exit". For $\mathsf{t} \in \{\text{target}, \text{exit}\}$, the t-*directed quotient of $\mathcal{M}$ by $\mathcal{D}$* is the MDP

$$\mathcal{M}^{\mathsf{t}}_{/\mathcal{D}} = (\{E_1, \ldots, E_k, \text{target}, \text{exit}\}, (S \times \text{Act}) \cup \{\tau\}, [s_{in}]_{\mathcal{D}}, P_{\mathcal{D}}),$$

where $[s_{in}]_{\mathcal{D}}$ is the unique E such that $(E, A) \in \mathcal{D}$ and $s_{in} \in E$, and for all $1 \le i, j \le k$:

- $P_{\mathcal{D}}\big(E_i, (s, \alpha), E_j\big) = \sum_{s' \in E_j} P(s, \alpha, s')$ if $s \in E_i$ and $\alpha \in \text{Act}(s) \setminus A_i(s)$,
- $P_{\mathcal{D}}\big(E_i, (s, \alpha), *\big) = P(s, \alpha, *)$ if $s \in E_i$ and $* \in \{\text{target}, \text{exit}\}$,
- $P_{\mathcal{D}}\big(E_i, \tau, \mathsf{t}\big) = 1$ if (E_i, A_i) is a proper end component,
- $P_{\mathcal{D}}$ maps to zero in all other cases.

If $\mathcal{D}$ is the set of maximal end components, then these are variants of the standard quotient of maximal end components, and the resulting MDP is EC-free. Internal actions (those that appear in $A(s)$, for any $s \in E$) are ignored in the quotient (see the first bullet) and a fresh action τ is

introduced which leads to either "exit" or "target" with probability one and is enabled in each state corresponding to a proper end component.

If $\mathsf{t} = \text{exit}$, then optimal probabilities of reaching target are preserved, and if $\mathsf{t} = \text{target}$, then optimal probabilities of *avoiding* exit forever are preserved, in the sense laid out by the following lemma. This is a standard observation, see for example [deA97, CBGK08].

Lemma 2.11. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ where* target *and* exit *are absorbing and let $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be the maximal end components of $\mathcal{M}$.*

1. *If $\mathcal{M}' = \mathcal{M}^{\text{exit}}_{/\mathcal{D}}$, then for all $s \in S$ we have:*

$$\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond\,\text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}',[s]_{\mathcal{D}}}(\Diamond\,\text{target}) \quad \textit{and} \quad \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond\,\text{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}',[s]_{\mathcal{D}}}(\Diamond\,\text{target}).$$

2. *If $\mathcal{M}' = \mathcal{M}^{\text{target}}_{/\mathcal{D}}$, then for all $s \in S$ we have:*

$$\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Box\neg\,\text{exit}) = \mathbf{Pr}^{\min}_{\mathcal{M}',[s]_{\mathcal{D}}}(\Box\neg\,\text{exit}) \quad \textit{and} \quad \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Box\neg\,\text{exit}) = \mathbf{Pr}^{\max}_{\mathcal{M}',[s]_{\mathcal{D}}}(\Box\neg\,\text{exit}).$$

Proof. Statement (1.) follows from the correctness of the standard MEC quotient (see [CBGK08, Theorem 2]). Adding transitions to "exit" has the same effect for the minimal probabilities as making the corresponding states in the quotient absorbing (see [CBGK08, Remark 3]) and does not change the maximal probabilities. For statement (2.) we observe that

$$\mathbf{Pr}^{\min}_{\mathcal{M},s}(\Box\neg\,\text{exit}) = 1 - \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond\,\text{exit}) \quad \text{and} \quad \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Box\neg\,\text{exit}) = 1 - \mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond\,\text{exit})$$

holds for all $s \in S$. Hence the statement follows by applying statement (1.) after interchanging the roles of "target" and "exit". □

2.2.3 Expected total reward

We will now consider the total expected reward criterion before reaching some dedicated set of states. An integer reward is associated with each state-action pair (s, α), which, intuitively, represents the value that is gained (or lost) each time action α is taken in state s. In contrast to the setting of probabilistic reachability we are not interested in what proportion of paths reaches "target" rather than "exit", but *how much reward* is collected in total before leaving the set of states in which a reward can be collected.

Hence, the distinction between "target" and "exit" is not important and we will only assume that the state "exit" exists and is reached with probability one by all schedulers. This is equivalent to requiring that "exit" induces the only proper end component. The choice of using the state "exit" here rather than "target" will be useful later to provide a unified definition of subsystems for probabilistic reachability constraints and constraints on the total expected reward.

Definition 2.12 (Reward reachability form). Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP such that

- state "exit" is absorbing and gives zero reward for all its enabled actions, and
- the only proper end component of $\mathcal{M}$ is induced by "exit".

If $\text{rew}(s, \alpha) \geq 0$ holds for all $(s, \alpha) \in S \times \text{Act}$, then $\mathcal{M}$ is in *nonnegative reward reachability form.*

Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP in reward reachability form. We define the reward of a path $\pi = s_1\alpha_1 s_2\alpha_2 \ldots$ to be $\text{rew}(\pi) = \sum_{1 \leq i} \text{rew}(s_i, \alpha_i)$. Given $T \subseteq S$, consider the random variable $\Diamond\!\!\!\!+ T : \text{Paths}(\mathcal{M}) \to \mathbb{Z}$ defined as follows:

$$\Diamond\!\!\!\!+ T(\pi) = \begin{cases} \text{rew}(\pi) & \text{if } \pi \models \Diamond T \\ \text{undefined} & \text{otherwise.} \end{cases}$$

In most cases we will have $T = \{\text{exit}\}$ and write $\Diamond\!\!\!\!+ \text{exit}$ instead of $\Diamond\!\!\!\!+ \{\text{exit}\}$. Observe that if π reaches the state "exit", then the value $\text{rew}(\pi)$ is finite as "exit" is absorbing.

If $\mathcal{M}$ is a Markov chain, we define the *expected total reward* as the expected value of $\Diamond\!\!\!\!+ \text{exit}$ in $\mathcal{M}$ (when starting in s_{in}). If $\mathcal{M}$ is an MDP we define, as in the case of probabilities:

$$\mathbb{E}^{\max}_{\mathcal{M}}(\Diamond\!\!\!\!+ \text{exit}) \;=\; \sup_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(\Diamond\!\!\!\!+ \text{exit}) \quad \text{and} \quad \mathbb{E}^{\min}_{\mathcal{M}}(\Diamond\!\!\!\!+ \text{exit}) \;=\; \inf_{\mathfrak{S}} \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(\Diamond\!\!\!\!+ \text{exit}),$$

where $\mathfrak{S}$ ranges over all schedulers for $\mathcal{M}$. Given a scheduler $\mathfrak{S}$ for $\mathcal{M}$ we will use the abbreviation $\mathbf{ex}^{\mathfrak{S}}_{\mathcal{M}} = \big(\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\!\!\!\!+ \text{exit})\big)_{s \in S}$, and in analogous fashion we define the vectors $\mathbf{ex}^{\max}_{\mathcal{M}} = \big(\mathbb{E}^{\max}_{\mathcal{M},s}(\Diamond\!\!\!\!+ \text{exit})\big)_{s \in S}$ and $\mathbf{ex}^{\min}_{\mathcal{M}} = \big(\mathbb{E}^{\min}_{\mathcal{M},s}(\Diamond\!\!\!\!+ \text{exit})\big)_{s \in S}$. If $\mathcal{M}$ is clear from the context, we will sometimes drop the subscripts. The system matrix $\mathbf{A}$ for MDPs in reward reachability form is defined as before (see Definition 2.7). It is not important for this definition that the MDP does not have a dedicated "target" state.

The following linear program characterizes $\mathbf{ex}^{\max}$: *minimize* $\sum_{s \in S} x_s$ *such that*

$$x_s \;\geq\; \text{rew}(s, \alpha) \;+\; \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} \qquad \text{for all} \quad s \in S,\ \alpha \in \text{Act}(s).$$

Similarly, $\mathbf{ex}^{\min}$ is characterized by the linear program: *maximize* $\sum_{s \in S} x_s$ *such that*

$$x_s \;\leq\; \text{rew}(s, \alpha) \;+\; \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} \qquad \text{for all} \quad s \in S,\ \alpha \in \text{Act}(s).$$

These results follow from [deA97, Theorem 3.4], see also [BT91, deA99]. If we define the vector $\mathbf{r} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ as $\mathbf{r}(s, \alpha) = \text{rew}(s, \alpha)$ for all $(s, \alpha) \in \mathcal{E}$, we may write the above inequalities as

$$\mathbf{Ax} \geq \mathbf{r} \qquad \text{and} \qquad \mathbf{Ax} \leq \mathbf{r},$$

where $\mathbf{A}$ is as defined in Definition 2.7. Vectors satisfying the above systems of inequalities yield point-wise bounds on the vectors $\mathbf{ex}^{\max}$ and $\mathbf{ex}^{\min}$. The proof is essentially the same as for the analogous statement for reachability probabilities (Lemma 2.9). Here one uses the fact that if $\mathcal{M}$ is EC-free, then the corresponding Bellman operator has a unique fixpoint (see [BT91]).

Lemma 2.13. *Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP in reward reachability form, and let $\mathbf{A}$ and $\mathbf{r}$ be defined as above. Then for all $\mathbf{z} \in \mathbb{R}^S$ we have:*

- $\mathbf{Az} \geq \mathbf{r}$ *implies* $\mathbf{z} \geq \mathbf{ex}^{\max}$
- $\mathbf{Az} \leq \mathbf{r}$ *implies* $\mathbf{z} \leq \mathbf{ex}^{\min}$.

As for reachability probabilities, it is also true that the optimal expected total reward is attained by memoryless and deterministic schedulers (see [deA97, Theorem 3.4]).

Proposition 2.14. *Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP in reward reachability form. Then, there exist memoryless and deterministic schedulers $\mathfrak{S}_{\min}$ and $\mathfrak{S}_{\max}$ such that*

$$\left(\mathbb{E}^{\mathfrak{S}_{\min}}_{\mathcal{M},s}(\Diamond\,\text{exit})\right)_{s\in S} = \mathbf{ex}^{\min} \quad \textit{and} \quad \left(\mathbb{E}^{\mathfrak{S}_{\max}}_{\mathcal{M},s}(\Diamond\,\text{exit})\right)_{s\in S} = \mathbf{ex}^{\max}.$$

2.2.4 Expected number of visits

Consider a DTMC $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ such that states "target" and "exit" are absorbing, all states in S are reachable from s_{in} in the underlying graph of $\mathcal{M}$, and the probability of reaching the set $\{\text{target}, \text{exit}\}$ from s_{in} is one (i.e., we have $\Pr_{s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$). In this section we consider in the following question: if we start in s_{in}, how often will any given state $s \in S$ be visited on average before the process reaches a state in $\{\text{target}, \text{exit}\}$? More formally, given a state $s \in S$ let us denote by $V_s : \text{Paths}(\mathcal{M}) \to \mathbb{N}$ the random variable which counts how often a path visited s. That is, we define $V_s(s_1 s_2 \ldots) = |\{i \in \mathbb{N} \mid s_i = s\}|$.

The vector $\mathbf{ev} \in \mathbb{R}^S$ is defined as the vector containing the expected values of V_s for each state, more precisely we let $\mathbf{ev}(s) = \mathbb{E}_{\mathcal{M},s_{in}}(V_s)$. Our assumptions on $\mathcal{M}$ guarantee that $\mathbf{ev}(s)$ is finite for each $s \in S$, as the probability of visiting s infinitely often is zero. We now recall how the expected number of visits can be computed. Consider the following matrix :

$$\left(\begin{array}{cc|c} 1 & 0 & 0\cdots 0 \\ 0 & 1 & 0\cdots 0 \\ \hline \mathbf{t} & \mathbf{e} & \mathbf{T} \end{array}\right)$$

where $\mathbf{t} \in \mathbb{R}^S$ is the target vector (defined by $\mathbf{t}(s) = P(s, \text{target})$ for all $s \in S$), $\mathbf{e}$ contains one-step probabilities to "exit" (defined by $\mathbf{e}(s) = P(s, \text{exit})$ for all $s \in S$) and $\mathbf{T} \in \mathbb{R}^{S\times S}$ contains the probabilities of moving from s to s' in one step for all pairs of states $s, s' \in S$.

By assumption, S includes exactly the *transient* states of $\mathcal{M}$. It follows by standard Markov chain theory (see, for example, [KS76, Theorem 3.2.1]) that the sequence $(\mathbf{T}^i)_{i\in\mathbb{N}}$ tends to the zero matrix, and, as a consequence, that $\mathbf{I} - \mathbf{T}$ has an inverse and satisfies

$$(\mathbf{I} - \mathbf{T})^{-1} \;=\; \mathbf{I} + \mathbf{T}^1 + \mathbf{T}^2 + \ldots \;=\; \sum_{i\geq 0} \mathbf{T}^i.$$

The matrix $\mathbf{F} = (\mathbf{I} - \mathbf{T})^{-1}$ is called the *fundamental matrix* of $\mathcal{M}$. For any pair of states $s, u \in S$, the entry $\mathbf{F}(s, u)$ is exactly the expected number of visits of u when starting in state s [KS76, Theorem 3.2.4]. Hence, in particular, we have $\mathbf{ev}(s) = \mathbf{F}(s_{in}, s)$ for all $s \in S$.

As a consequence, the vector $\mathbf{ev}$ can be characterized as follows. The matrix $\mathbf{F}$ is the unique matrix satisfying $\mathbf{F}(\mathbf{I} - \mathbf{T}) = \mathbf{I}$. Then the row of $\mathbf{F}$ which corresponds to state s_{in} is the unique vector satisfying the equation system $\mathbf{y}(\mathbf{I} - \mathbf{T}) = \delta_{s_{in}}$, where $\mathbf{y} = (y_s)_{s\in S}$ is a vector of variables. (Recall that $\delta_{s_{in}}$ is the dirac vector for s_{in}.) This shows the following lemma.

Lemma 2.15. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ be a Markov chain with transient states S, and let $\mathbf{T} \in \mathbb{R}^{S\times S}$ be defined by $\mathbf{T}(s, s') = P(s, s')$ for all $s, s' \in S$. Then, the vector $\mathbf{ev} \in \mathbb{R}^S$ is the unique vector satisfying $\mathbf{ev}(\mathbf{I} - \mathbf{T}) = \delta_{s_{in}}$.*

Writing out the equation system $\mathbf{y}(\mathbf{I} - \mathbf{T}) = \delta_{s_{in}}$ yields:

$$y_s \;=\; \delta_{s_{in}}(s) + \sum_{s'\in S} P(s', s) \cdot y_{s'} \quad \text{for all } s \in S.$$

These equations express that the value in state s should be the sum of the values of the *predecessors* of s multiplied by the probabilities on the corresponding edges. Additionally, the value in state s_{in} is increased by one. This reflects that s_{in} is the starting state and hence has one guaranteed visit.

The probability of reaching state "target" when starting in s_{in} can be expressed directly in terms of the values $\mathbf{ev}(s)$ as follows:

$$\Pr_{\mathcal{M},s_{in}}(\Diamond \text{target}) \;=\; \sum_{s\in S} \mathbf{ev}(s)\cdot \mathbf{t}(s).$$

This can be shown formally by first observing that the vector $\left(\Pr_{\mathcal{M},s}(\Diamond \text{target})\right)_{s\in S}$ is the unique solution of the equation system $(\mathbf{I}-\mathbf{T})\mathbf{x} = \mathbf{t}$. Hence, we have $\left(\Pr_{\mathcal{M},s}(\Diamond \text{target})\right)_{s\in S} = (\mathbf{I}-\mathbf{T})^{-1}\,\mathbf{t} = \mathbf{F}\cdot\mathbf{t}$. Now it suffices to observe that the row of $\mathbf{F}$ which corresponds to s_{in} is exactly given by the vector $\mathbf{ev}$, and hence $\Pr_{\mathcal{M},s_{in}}(\Diamond \text{target}) = \mathbf{ev}\cdot\mathbf{t}$.

Expected number of visits in MDPs. For MDPs $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ with enabled actions $\mathcal{E}$ we will consider the expected number of visits of state-action pairs $(s,\alpha) \in \mathcal{E}$. If $\mathfrak{S}$ is a scheduler satisfying $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$, then these values can be defined analogously to the case of Markov chains. That is, we let $V_{(s,\alpha)}(s_1\alpha_1 s_2\alpha_2\ldots) = |\{i \in \mathbb{N} \mid (s_i,\alpha_i) = (s,\alpha)\}|$ and define $\mathbf{ev}^{\mathfrak{S}}(s,\alpha) = \mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(V_{(s,\alpha)})$. For MDPs in reachability form which are *EC-free*, and for MDPs in *reward reachability form*, all schedulers satisfy the above assumption. This is because the only proper end components are induced by the absorbing states "target" and "exit" in those cases.

As for Markov chains, the probability of reaching target can be expressed directly in terms of values $\mathbf{ev}^{\mathfrak{S}}(s,\alpha)$. We have:

$$\Pr^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\Diamond \text{target}) = \sum_{(s,\alpha)\in\mathcal{E}} \mathbf{ev}^{\mathfrak{S}}(s,\alpha)\cdot\mathbf{t}(s,\alpha),$$

where $\mathbf{t} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ is the target vector as defined in Definition 2.7. Similarly, if $\mathcal{M}$ is equipped with a reward function $\text{rew} : \mathcal{E} \to \mathbb{Z}$, then the expected total reward from s_{in} under scheduler $\mathfrak{S}$ is given by:

$$\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\diamondsuit\{\text{target}, \text{exit}\}) = \sum_{(s,\alpha)\in\mathcal{E}} \mathbf{ev}^{\mathfrak{S}}(s,\alpha)\cdot \text{rew}(s,\alpha).$$

2.3 Probabilistic timed automata

Probabilistic timed automata are a model of computation which extend Markov decision processes by incorporating *continuous time*. This is done using a set of real-valued clock variables, which can be reset to zero and used in transition guards.

2.3.1 Definitions

We fix a finite set of *clocks* $\mathsf{Cl} = \{c_0, c_1, \ldots, c_n\}$, where by convention c_0 is a designated clock always representing the value 0. This allows encoding absolute and relative time bounds in a uniform manner. A *valuation* of Cl is a map $v\colon \mathsf{Cl} \to \mathbb{R}_{\geq 0}$ such that $v(c_0) = 0$. The set of all valuations of Cl is denoted by $\text{Val}(\mathsf{Cl})$. For a valuation v and $t \in \mathbb{R}_{\geq 0}$ we denote by $v + t$ the

valuation satisfying $(v+t)(c) = v(c) + t$ for all $c \in \mathsf{Cl} \setminus \{c_0\}$. Given $C \subseteq \mathsf{Cl}$ we let $v[C := 0]$ be the *reset* valuation defined by $v[C := 0](c) = 0$ for $c \in C$ and $v[C := 0](c) = v(c)$ for $c \notin C$.

The set of *clock constraints* $\mathrm{CC}(\mathsf{Cl})$ is formed according to the following grammar:

$$g \ ::= \ \text{true} \ \mid \ \text{false} \ \mid \ c - c' \sim x \ \mid \ g \wedge g,$$

where $c, c' \in \mathsf{Cl}$, $x \in \mathbb{Z} \cup \{\infty, -\infty\}$, and $\sim \in \{\leq, <, \geq, >\}$. A valuation v satisfies a clock constraint g, written as $v \models g$, if replacing every clock variable c in g with the value $v(c)$ leads to a true formula. We define $\mathrm{Val}(g) = \{v \in \mathrm{Val}(\mathsf{Cl}) \mid v \models g\}$ and write $g_1 \Vdash g_2$ if $\mathrm{Val}(g_1) \subseteq \mathrm{Val}(g_2)$ holds, and $g_1 \equiv g_2$ if $\mathrm{Val}(g_1) = \mathrm{Val}(g_2)$ holds. A subset $Z \subseteq \mathrm{Val}(\mathsf{Cl})$ is called a *zone* if $Z = \mathrm{Val}(g)$ for some clock constraint g.

Definition 2.16. A *probabilistic timed automaton* (PTA) is a tuple $\mathcal{T} = (\mathrm{Loc}, \mathsf{Cl}, \mathrm{Act}, \mathrm{inv}, T, l_{in})$, where

- Loc, Cl and Act are finite sets of *locations*, *clocks* and *actions*, respectively,
- $\mathrm{inv}\colon \mathrm{Loc} \to \mathrm{CC}(\mathsf{Cl})$ is the *invariant condition*,
- $T\colon \mathrm{Loc} \times \mathrm{Act} \to \mathrm{CC}(\mathsf{Cl}) \times \mathrm{Dist}(2^{\mathsf{Cl}} \times \mathrm{Loc})$ is the transition function, and
- l_{in} is the *initial location*, for which we assume that $\mathbf{0} \models \mathrm{inv}(l_{in})$. Here $\mathbf{0}$ is the valuation in which each clock is mapped to zero.

We will assume that $\mathrm{inv}(l) \not\equiv \text{false}$ holds for all $l \in \mathrm{Loc}$.

An action $\alpha \in \mathrm{Act}$ is said to be *enabled* in location $l \in \mathrm{Loc}$ if $T(l, \alpha) = (g, \mu)$ implies $g \not\equiv \text{false}$. We denote by $\mathrm{Act}(l)$ the actions enabled in l and we assume that $\mathrm{Act}(l) \neq \emptyset$ for each $l \in \mathrm{Loc}$. Given a PTA $\mathcal{T}$, we let $\mathrm{Loc}(\mathcal{T})$ be the locations of $\mathcal{T}$. A transition $T(l, \alpha) = (g, \mu)$ is also denoted by $l \xrightarrow{\alpha : g} \mu$ and the clock constraint g is called the *guard*. We say that a location $l \in \mathrm{Loc}$ is *absorbing* if for all $\alpha \in \mathrm{Act}$ and $T(l, \alpha) = (g, \mu)$ the support of μ contains no other locations than l. That is, an execution of $\mathcal{T}$ can never leave l after entering this location. The semantics of a PTA is given in terms of a *timed probabilistic system*, which is introduced next.

Timed probabilistic systems. A *timed probabilistic system* (TPS) is a tuple $\mathcal{S} = (S, \mathrm{Act}', T, s_{in})$, where S is a (possibly infinite) set of states, $\mathrm{Act}' = \mathrm{Act} \cup \mathbb{R}_{>0}$ is a set of actions (Act is assumed to be finite and disjoint with $\mathbb{R}_{>0}$), $T : S \times \mathrm{Act}' \to \mathrm{Dist}(S) \cup \{0\}$ is the probabilistic transition function, and s_{in} the *initial state*. We say that $\alpha \in \mathrm{Act}'$ is *enabled* in s if $T(s, \alpha) \neq 0$ holds, and denote by $\mathrm{Act}'(s)$ the set of enabled actions in state s.

We assume that μ has finite support whenever $T(s, \alpha) = \mu$ for some $\alpha \in \mathrm{Act}'(s)$. Instead of writing $T(s, \alpha) = \mu$, we will sometimes use the notation $s \xrightarrow{\alpha} \mu \in T$. Transitions whose action is in $\mathbb{R}_{>0}$ are called *time delays* and transitions with actions in Act are called *discrete actions*. Timed probabilistic systems can be viewed as a kind of MDP with possibly uncountable state space, and schedulers for them are defined precisely as for MDPs. A scheduler $\mathfrak{S}$ for $\mathcal{S}$ is said to be *time-divergent* if for almost every path compatible with $\mathfrak{S}$ the corresponding series of time delays is divergent. Optimal reachability probabilities $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S},s}(\Diamond G)$, for $\mathfrak{m} \in \{\min, \max\}$ and $G \subseteq S$, are defined as for MDPs, but with the quantification restricted to time-divergent schedulers.

The semantics of PTA. We now present the semantics of PTA in terms of timed probabilistic systems, which is tailored to the special context of reachability queries. A *pointed* PTA $\mathcal{T}$ is a PTA $\mathcal{T} = (\text{Loc}, \text{Cl}, \text{Act}, \text{inv}, T, l_{in})$ which includes the two distinguished absorbing locations target, exit $\in$ Loc. The semantics of a pointed PTA is the TPS $\mathcal{S}(\mathcal{T}) = (S, \text{Act}', T_{\text{sem}}, s_{in})$ with $S = \{(l, v) \in \text{Loc} \times \text{Val}(\text{Cl}) \mid v \models \text{inv}(l)\}$, $\text{Act}' = \text{Act} \cup \mathbb{R}_{>0}$, $s_{in} = (l_{in}, \mathbf{0})$, and $T_{\text{sem}} : S \times \text{Act}' \to \text{Dist}(S) \cup \{0\}$ is the function defined by the inference rules

$$\frac{t \in \mathbb{R}_{>0}, \quad \forall t' \leq t.\, v + t' \models \text{inv}(l)}{(l, v) \xrightarrow{t} \delta_{(l,\, v+t)} \in T_{\text{sem}}} \quad \text{and} \quad \frac{l \xrightarrow{\alpha\,:\,g} \mu \in T, \quad v \models g}{(l, v) \xrightarrow{\alpha} \mu_{\text{sem}} \in T_{\text{sem}}}, \quad \text{where}$$

$$\mu_{\text{sem}}(l', v') = \sum_{\substack{(C, l') \\ v' = v[C:=0]}} \mu(C, l') \quad \text{for } l' \neq \text{exit and } v' \models \text{inv}(l'), \text{ and} \tag{2.1}$$

$$\mu_{\text{sem}}(\text{exit}, v') = \sum_{\substack{(C, \text{exit}) \\ v' = v[C:=0]}} \mu(C, \text{exit}) + \sum_{\substack{(C, l'),\, l' \neq \text{exit} \\ v' = v[C:=0] \not\models \text{inv}(l')}} \mu(C, l'). \tag{2.2}$$

If $\big((l, v), \alpha\big) \in \text{Loc} \times \text{Val}(\text{Cl}) \times \text{Act}'$ does not satisfy any precondition of the rules above, we require $T_{\text{sem}}\big((l, v), \alpha\big) = 0$.

Typically, the semantics of a PTA is only defined if it is *well-formed*, which means that no transition leads to a violation of the invariant condition of the successor location. We relax this condition and, in the case that $v' = v[C := 0] \not\models \text{inv}(l')$ holds, add the probability of (C, l') to the edge $(l, v) \xrightarrow{\alpha} (\text{exit}, v')$ (this is the second sum in Equation (2.2)). The intuition is that if an edge is taken which leads to a state which does not satisfy the invariant of the successor location, then that should be counted as failure, which in our context can be represented by redirecting the transition to "exit".

We define the set of target-states in $\mathcal{S}(\mathcal{T})$ to be $\text{target}_{\mathcal{S}(\mathcal{T})} = \{(l, v) \in S \mid l = \text{target}\}$. For $\mathfrak{m} \in \{\min, \max\}$ the probability to reach target in $\mathcal{T}$ is defined as

$$\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}, l_{in}}(\Diamond\, \text{target}) \quad = \quad \mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S}(\mathcal{T}), s_{in}}(\Diamond\, \text{target}_{\mathcal{S}(\mathcal{T})}).$$

Probabilistic time-abstracting bisimulation. A *probabilistic time-abstracting bisimulation* (PTAB) on a TPS $\mathcal{S} = (S, \text{Act} \cup \mathbb{R}_{>0}, T, s_{in})$ is an equivalence relation $\sim$ on S such that if $s \sim s'$ we have:

1. for any time delay $s \xrightarrow{t} u$ there exists a time delay $s' \xrightarrow{t'} u'$ such that $u \sim u'$;
2. for any discrete action $s \xrightarrow{\alpha} \mu$, there exists a discrete action $s' \xrightarrow{\alpha} \mu'$ such that for all $E \in S/_{\sim}$ we have $\sum_{t \in E} \mu(t) = \sum_{t \in E} \mu'(t)$.

See [CHK08] for additional details. We let $[s]$ be the equivalence class of $\sim$ which includes s.

For the next definition we need the following notion: An equivalence class $[s]$ of $\sim$ is called *unbounded* if there exists a path through $[s]$ which takes only time delay transitions and is time divergent (i.e., the induced series of time delays diverges). The *quotient* of $\mathcal{S}$ by $\sim$ is the Markov decision process $\mathcal{S}/_{\sim} = (S/_{\sim},\ \text{Act} \cup \{\tau_E \mid E \in S/_{\sim}\},\ [s_{in}],\ P)$ where P is defined as follows. For

any transition $T(s,\alpha) = \mu$ of $\mathcal{S}$ such that $\mu \neq 0$, we define:

$$P([s],\alpha,[s']) \;=\; \sum_{t\in[s']} \mu(t).$$

If $T(s,\alpha) = 0$, then $P([s],\alpha,[s']) = 0$ for all $[s'] \in S/_\sim$. Furthermore, we let:

$$P([s],\tau_{[s']},[s']) \;=\; \begin{cases} 1 & \text{if } \exists t \in \mathbb{R}_{>0} \text{ s.t. } \delta_{s'} \in T(s) \text{ and } ([s] \neq [s'] \text{ or } [s] \text{ is unbounded }) \\ 0 & \text{otherwise} \end{cases}$$

The above definitions are well-defined as $\sim$ is a time-abstracting bisimulation. Excluding τ-transitions satisfying $[s] = [s']$ if $[s]$ is bounded is supported by the fact that only time-divergent schedulers are considered in timed probabilistic systems. If $[s]$ is bounded, then time-divergent schedulers cannot stay forever in $[s]$ by taking only time delays.

Let us assume that $\mathcal{S}$ is the semantics of a pointed PTA and therefore has states of the type $S = \{(l,v) \in \mathrm{Loc} \times \mathrm{Val}(\mathsf{Cl}) \mid v \models \mathrm{inv}(l)\}$ with distinguished locations target, exit $\in$ Loc. Then we say that a PTAB $\sim$ on $\mathcal{S}$ *respects* target and exit if whenever $(l,v) \sim (\mathrm{target}, v')$ holds, then so does $l = \mathrm{target}$, and likewise for exit. If $\sim$ has this property, we denote by "target" the set of states $[s]$ of $\mathcal{S}/_\sim$ such that $s = (\mathrm{target}, v)$ for some $v \in \mathrm{Val}(\mathsf{Cl})$, and analogously for "exit". More generally, the bisimulation $\sim$ is said to *distinguish locations* if whenever $(l,v) \sim (l',v')$ holds, then $l = l'$. We say that $\sim$ is PTAB on the PTA $\mathcal{T}$ if $\sim$ is a PTAB on $\mathcal{S}(\mathcal{T})$. The following is a standard result on probabilistic time-abstracting bisimulation [CHK08].

Lemma 2.17. *Let $\mathcal{S}$ be a TPS and $\sim$ a PTAB on $\mathcal{S}$ which respects* target *and* exit. *Then for all $s \in S$ and $\mathfrak{m} \in \{\min, \max\}$ we have*

$$\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S},s}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S}/_\sim,[s]}(\Diamond\,\mathrm{target}).$$

Region equivalence. A special PTAB is the *region equivalence* [AD94, BK08]. Let $\mathcal{T}$ be a PTA over clocks Cl, and K be the largest number which appears in any clock constraint in the description of $\mathcal{T}$. For the following definition, let $\mathrm{frac}(a)$ denote the fractional part of $a \in \mathbb{R}$, i.e., $\mathrm{frac}(a) = a - \lfloor a \rfloor$. Then, clock valuations $v_1, v_2 \in \mathrm{Val}(\mathsf{Cl})$ are called *region-equivalent* if one of the two following conditions hold:

- for all $c \in \mathsf{Cl} \setminus \{c_0\}$ we have $v_1(c) > K$ and $v_2(c) > K$, or
- for all $c, c' \in \mathsf{Cl}$ all of the following hold:
  - $\lfloor v_1(c) \rfloor = \lfloor v_2(c) \rfloor$,
  - $\mathrm{frac}(v_1(c)) = 0$ if and only if $\mathrm{frac}(v_2(c)) = 0$, and
  - $\mathrm{frac}(v_1(c)) \leq \mathrm{frac}(v_1(c'))$ if and only if $\mathrm{frac}(v_2(c)) \leq \mathrm{frac}(v_2(c'))$.

The region equivalence is a PTAB [CHK08], and the number of its equivalence classes is exponential in the size of the PTA [AD94].

2.3.2 Difference bounds matrices

A common data structure to represent clock constraints are *difference bounds matrices* (DBMs). A DBM is a $\mathsf{Cl} \times \mathsf{Cl}$-matrix M over $(\mathbb{Z} \cup \{\infty, -\infty\}) \times \{<, \leq\}$. The entry $M_{ij} = (a, \triangleleft)$ represents

the constraint $c_i - c_j \lhd a$. For example, consider the following clock constraint over clocks $\mathsf{Cl} = \{c_0, c_1, c_2\}$:

$$c_1 - c_0 \leq 5 \quad \wedge \quad c_2 - c_1 \leq -2 \quad \wedge \quad c_1 - c_2 < 3$$

It corresponds to the DBM:

$$\begin{pmatrix} (\infty, <) & (\infty, <) & (\infty, <) \\ (5, \leq) & (\infty, <) & (3, <) \\ (\infty, <) & (-2, \leq) & (\infty, <) \end{pmatrix}$$

As c_0 is mapped to zero in every valuation, the above constraints represents the formula

$$c_1 \leq 5 \quad \wedge \quad 2 \leq c_1 - c_2 < 3.$$

In general, a DBM M with entries $M_{ij} = (\lhd_{ij}, a_{ij})$ corresponds to the clock constraint

$$\bigwedge_{0 \leq i,j \leq n} c_i - c_j \lhd_{ij} a_{ij}.$$

We define $\mathrm{Val}(M)$ to be the set of valuations which satisfy the clock constraint one gets in this way from M.

To compare DBMs and their entries, we define $\preceq$ to be the lexicographic order on $(\mathbb{Z} \cup \{\infty, -\infty\}) \times \{<, \leq\}$ in which $<$ is strictly less than $\leq$. This order can be extended naturally to a partial order on DBMs by entry-wise comparison.

Operations on difference bounds matrices. All min and max operators in this section use the order $\preceq$ as defined above. The operations $+$ and $\sqcap$ on $(\mathbb{Z} \cup \{\infty, -\infty\}) \times \{<, \leq\}$ are defined as follows (see [Dil90]):

$$\begin{aligned} (a, \lhd_1) + (b, \lhd_2) &= (\ a + b,\ \min\{\lhd_1, \lhd_2\}\ ) \\ (a, \lhd_1) \sqcap (b, \lhd_2) &= \min\{\ (a, \lhd_1), (b, \lhd_2)\ \} \end{aligned}$$

The operations are lifted to DBMs by letting $\sqcap$ be the entry-wise application of $\sqcap$ and $+$ be a matrix multiplication which uses the scalar operations $\sqcap$ (for addition) and $+$ (for multiplication). More precisely, given two DBMs M and N over a set of clocks $\mathsf{Cl} = \{c_0, c_1, \ldots c_n\}$ we define:

$$\begin{aligned} \big(\ M \sqcap N\ \big)_{ij} &= M_{ij} \sqcap N_{ij}, \text{ and} \\ \big(\ M + N\ \big)_{ij} &= \prod_{0 \leq k \leq n} (M_{ik} + M_{kj}), \end{aligned}$$

for all $0 \leq i, j \leq n$.

Canonical difference bounds matrices. The same zone can be represented using different DBMs. In particular it can happen that some constraint can be tightened in a DBM without changing the set of accepting valuations, as the tighter bound was already implied by the other constraints. To get a canonical representation for a DBM M, [Dil90] defines

$$M^* \quad = \quad M^0 \sqcap M^1 \sqcap \ldots, \qquad \text{where} \qquad M^i = \underbrace{M + M + \ldots + M}_{i\text{-times}}.$$

We have $\mathrm{Val}(M) = \mathrm{Val}(M^*)$ for all DBMs M. Furthermore, two DBMs M and N with $\mathrm{Val}(M) = \mathrm{Val}(N) \neq \emptyset$ satisfy $M^* = N^*$ and M^* can be computed in polynomial time given M [Dil90].

To an arbitrary nonempty set of clock valuations $R \subseteq \mathrm{Val}(\mathsf{Cl})$ we also associate a canonical DBM M_R. It's entries are defined as follows for all $c_i, c_j \in \mathsf{Cl}$:

$$(M_R)_{ij} \;=\; \big(\sup\{v(c_i) - v(c_j) \mid v \in R\},\;\; \triangleleft\big),$$

where $\triangleleft = \leq$ exactly if the supremum is attained in R, and otherwise $\triangleleft = <$. The following lemma shows that M_R is the smallest zone which includes R and satisfies $M_R^* = M_R$.

Lemma 2.18. *Let $R \subseteq \mathrm{Val}(\mathsf{Cl})$ be a nonempty set of valuations. Then the following hold:*

1. $R \subseteq \mathrm{Val}(M_R)$,
2. $M_R^* = M_R$,
3. $\mathrm{Val}(M_R)$ *is the smallest zone in* $\mathrm{Val}(\mathsf{Cl})$ *which contains R, and*
4. *for any DBM M with $M = M^*$ and $\mathrm{Val}(M) \neq \emptyset$, we have $M = M_{\mathrm{Val}(M)}$.*

Proof. (1.) It is clear from the definition that all valuations in R satisfy the clock constraint induced by M_R, so we have $R \subseteq \mathrm{Val}(M_R)$.

(2.) Suppose for contradiction that $M_R \neq M_R^*$. Since $M^* \preceq M$ holds for any DBM, we must have a strict inequality $M_R^* \prec M_R$. Hence there exists a pair of indices i, j such that $(M_R^*)_{ij} \prec (M_R)_{ij}$. Let i, j be such a pair, which means that we have $(M_R^*)_{ij} = (b_1, \triangleleft_1) < (b_2, \triangleleft_2) = (M_R)_{ij}$. We first consider the case that $b_1 < b_2$. Take $\epsilon > 0$ small enough such that $b_1 + \epsilon < b_2$. By the definition of M_R we have $b_2 = \sup\{v(i) - v(j) \mid v \in R\}$, so there exists $v \in R$ such that $v(i) - v(j) > b_2 - \epsilon = b_1$. This would entail $v \notin \mathrm{Val}(M_R^*) = \mathrm{Val}(M_R)$, which contradicts $R \subseteq \mathrm{Val}(M_R)$. Now consider the case that $b_1 = b_2$, $\triangleleft_1 = <$ and $\triangleleft_2 = \leq$. There must exist $v \in R$ such that $v(i) - v(j) = b_1 = b_2$. But this point will not be contained in $\mathrm{Val}(M_R^*)$ due to the strict inequality, which results once more in a contradiction. This concludes the proof of $M_R = M_R^*$.

(3.) First consider the case that R itself is a zone, so $R = \mathrm{Val}(g)$ for some clock constraint g. Let M_g be the associated DBM. One proves along similar lines as in (2.) that $M_R \preceq M_g$ holds. This implies that $R \subseteq \mathrm{Val}(M_R) \subseteq \mathrm{Val}(M_g) = \mathrm{Val}(g) = R$, and hence $R = \mathrm{Val}(M_R)$.

For general R, let $Z \subseteq \mathrm{Val}(\mathsf{Cl})$ be any zone with $R \subseteq Z$. Then $Z = \mathrm{Val}(M_Z)$ holds for the canonical DBM M_Z of Z, as shown in the previous paragraph. From $R \subseteq Z$, we clearly have $M_R \preceq M_Z$ and thus $\mathrm{Val}(M_R) \subseteq \mathrm{Val}(M_Z) = Z$. Therefore, any zone containing R must also contain $\mathrm{Val}(M_R)$.

(4.) Let $Z = \mathrm{Val}(M)$. Since Z is a zone, by part (3) we have $\mathrm{Val}(M_Z) = Z = \mathrm{Val}(M)$. It follows then from part (2) that $M_Z = M_Z^* = M^* = M$. □

Time closure operation on DBMs. The time closure on DBMs is the unary operation $\uparrow$ defined by $(\uparrow M)_{ij} = M_{ij}$ if $j \neq 0$ and $(\uparrow M)_{i0} = (\infty, <)$ otherwise. In words, the time closure removes absolute time bounds on the clocks in Cl. It reflects the semantic time closure operation on subsets $R \subseteq \mathrm{Val}(\mathsf{Cl})$ defined by $\uparrow R = \{v + t \in \mathrm{Val}(\mathsf{Cl}) \mid v \in R \text{ and } t \geq 0\}$ [BY04].

Lemma 2.19. *For any DBM M with $M = M^*$ and $\mathrm{Val}(M) \neq \emptyset$, we have $\mathrm{Val}(\uparrow M) = \uparrow \mathrm{Val}(M)$.*

Chapter 3

Farkas certificates

As modern software and hardware systems grow more and more complex, they inevitably become harder to understand and analyze. This is a phenomenon that applies both to the ability of human users to judge the correctness of an implementation and to the applicability of automated verification methods. In the context of machine learning, usually no classical description exists at all of the algorithms which are deployed. In such cases the challenge of guaranteeing correctness of its results is even bigger. When applying complex algorithms to formally verify systems, it is even more important that the results of a concrete implementation can be trusted.

A well-known methodology to tackle this problem is that of *certifying algorithms* [MMNS11]. Certifying algorithms produce, along with the result of the computation, a certificate which proves that the result is correct. In this way, arbitrarily complex implementations can be used to solve a problem, while maintaining a high level of assurance in the results. If certifying algorithms are employed, a user of verification technology can independently ensure themselves that the outcome of the computation is valid. This greatly increases the trust that can be put into the system, and decreases the impact of bugs.

This chapter considers the problem of certifying that a probabilistic reachability constraint is satisfied by a Markov decision process. Probabilistic reachability constraints are threshold constraints on the optimal reachability probabilities in some distinguished state, and allow expressing properties such as "in the worst case, the failure probability of the system is at most ϵ", or, "there exists a scheduler which ensures reaching a good state with probability at least λ". Here by "optimal" we always mean either minimal or maximal, depending on the constraint one wants to specify.

While computing the optimal reachability probabilities (and hence also verifying probabilistic reachability constraints) can be done in polynomial time, doing so efficiently for very large systems is not trivial. A technique which scales well and is used by modern model checkers is to approximate the optimal values using *value iteration* [Put94, FKNP11]. An issue in this context is that it is hard to decide when the approximation is good enough, and it has been demonstrated that using naïve stopping criteria can lead to results which are far from the optimal value [HM14]. This issue has been addressed in multiple works [HM14, BKL+17, QK18, HK20].

Table 3.1: Overview of Farkas certificates for the different types of probabilistic reachability constraints (here $\lesssim \in \{\leq, <\}$ and $\gtrsim \in \{\geq, >\}$) in an EC-free MDP $\mathcal{M}$ in reachability form with system matrix $\mathbf{A}$ and target vector $\mathbf{t}$. Here $\mathcal{E}$ denotes the enabled state-action pairs of $\mathcal{M}$ and S its states excluding the dedicated states "target" and "exit".

constraint	certificate dimension	certificate condition
$\mathbf{Pr}^{\min}_{s_{in}}(\lozenge\,\text{target}) \gtrsim \lambda$	$\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$	$\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \gtrsim \lambda$
$\mathbf{Pr}^{\max}_{s_{in}}(\lozenge\,\text{target}) \gtrsim \lambda$	$\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$	$\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \gtrsim \lambda$
$\mathbf{Pr}^{\min}_{s_{in}}(\lozenge\,\text{target}) \lesssim \lambda$	$\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$	$\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$
$\mathbf{Pr}^{\max}_{s_{in}}(\lozenge\,\text{target}) \lesssim \lambda$	$\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$	$\mathbf{Az} \geq \mathbf{t} \wedge \mathbf{z}(s_{in}) \lesssim \lambda$

In view of these complications which come up when solving the model checking problems in practice, it becomes clear that the ability to independently validate the results of a probabilistic model checker is extremely valuable.

To define certificate conditions for all types of probabilistic reachability constraints, we build upon the well-known characterizations of optimal reachability probabilities using linear programming [Kal83, Kal94]. Consequently, the certificates that we introduce are solutions of systems of linear inequalities. A key tool which we use to cover all cases is *Farkas' lemma* and variants thereof, and hence the certificates are named *Farkas certificates*. Table 3.1 gives an overview of how Farkas certificates are defined using the system matrix $\mathbf{A}$ and the target vector $\mathbf{t}$ of an MDP in reachability form. The definition as stated in Table 3.1 hold only for end component free MDPs. The structure of the definition stays the same for arbitrary MDPs in reachability form, but it requires computing the maximal end components beforehand and uses the restriction of the system matrix to states not included in proper end components (see Definition 3.23).

Farkas certificates can be used to design *certifying* model checking procedures for probabilistic reachability constraints as follows. Given a constraint, say $\mathbf{Pr}^{\min}_{s_{in}}(\lozenge\,\text{target}) \geq \lambda$, one can check whether it holds in the system using two methods. Either, one checks whether $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ is *satisfiable*, or whether $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} < \lambda$ is *unsatisfiable*. The first method is certifying for positive answers (i.e., it provides a certificate for $\mathbf{Pr}^{\min}_{s_{in}}(\lozenge\,\text{target}) \geq \lambda$), while the second is certifying for negative answers (by providing a certificate for $\mathbf{Pr}^{\min}_{s_{in}}(\lozenge\,\text{target}) < \lambda$). Validating certificates produced in this way is simple: one only has to check whether the vector indeed satisfies the linear inequalities. This can be done in linear time and, in particular, is considerably easier than computing the optimal reachability probability, which requires solving a linear program.

Related work

Certifying algorithms. The idea of certifying algorithms as a means for reliable software was first developed within the Leda-platform [MN99], in particular in the papers [MN98, MNS+99]. Later, the name *certifying algorithm* was coined in [KMMS06] and a comprehensive survey can be found in [MMNS11]. A key result of these works is that even for problems which are solvable in polynomial time it is hugely beneficial to produce a certificate which is easy to validate independently (usually this means "in linear time"). For linear programming, [MMNS11,

Section 8.2] describes a certifying algorithm which is based on solving the primal and the dual program. A pair of solutions to both programs yields an easy-to-validate certificate. In contrast, the Farkas certificates introduced in this section are based on *either* the primal or the dual formulation of a linear program (but not both) and allow many more solutions, in general. Their derivation is based on MDP-specific properties of the considered systems of linear inequalities. See also [And01] for a discussion on certifying LP-solver results.

In the model checking context, *counterexamples* can be viewed as witnesses for negative model checking results and have been studied extensively in various domains [CV03], including for probabilistic systems [HK07a, AL09, ÁBD+14, WJÁ+14, Jan15]. The counterexamples studied for probabilistic systems are, in general, *not* certificates, in the sense that validating them is not significantly easier than checking that the property is violated in the original system. To certify a positive (non-probabilistic) model checking result the main approaches either construct a proof of the property [Nam01, PPZ01, BMS+17, GRT18] or provide rank-based certificates, for example to certify emptiness of an automaton [KV04].

Farkas-based certificates. Farkas' lemma and its numerous generalizations (see [DJ14]) are a well-known source of certificates and have been applied in many contexts. A notable application in computer science is the automatic generation of inductive invariants. This line of work was kicked off by [CSS03, SSM04], which use Farkas' lemma to encode the synthesis problem of a linear inductive invariant for linear transition systems as a satisfiability problem of a set of non-linear constraints. The approach has also been used for invariant generation of probabilistic programs [KMMM10, CNŽ17].

Linear-programming based solutions for MDPs. Linear programming is one of the main approaches to tackle optimization problems arising in MDPs, and the method has been studied extensively [Kal83, Kal94, Put94, deA97]. Many of our results use fundamental techniques that have been established in this area. The correspondence between (certain types of) Farkas certificates and schedulers is based on the standard correspondence between the dual LP for reachability probabilities and transient memoryless schedulers described in [Kal83]. Our results differ mainly in the following ways. First, we lift the assumption that all states are in the support of the initial distribution, and this implies that certain solutions no longer correspond to a scheduler. Second, we consider also solutions of a version of the dual program in which equalities are replaced by inequalities (see Lemma 3.17). Third, we analyze in detail the role that end components play in this correspondence. In general, our work puts an emphasis on certifying algorithms, which have not been studied before in this context.

Reliable probabilistic model checking. As linear programming based approaches to solve MDP model checking do not scale to very large systems at the moment [FKNP11] and do not work well for symbolically represented systems, modern probabilistic model checkers such as Prism [KNP11] and Storm [DJKV17] include algorithms based on value iteration. It was shown in [HM14] that naïve stopping criteria can lead to results which are far off from the optimum, which motivated *interval iteration* [HM14]. Interval iteration is an algorithm based on value iteration which returns an interval which is guaranteed to contain the optimal value. Further techniques on making algorithms based on value iteration both efficient and reliable were developed in [QK18, HK20], and [BKL+17] considers the same problem for the expected total reward.

Outline

We start by defining probabilistic reachability constraints and relating them to threshold properties on the probability achieved by schedulers of the MDP, which are either existentially or universally quantified. Section 3.1.1 defines Farkas certificates for EC-free MDPs, first for the universal statements and then for the existential statements. The certificates for the universal statements are derived using the linear programming characterization (Proposition 3.1). Using this fact together with a variant of Farkas' lemma, certificates for the existential statements are derived in Proposition 3.4. In Section 3.1.2, a correspondence between Farkas certificates for existential statements and the expected number of visits under certain schedulers is established. Then, Section 3.1.3 lifts the results to MDPs which are not EC-free, and gives the general definition of Farkas certificates (Definition 3.23) and the proof that they indeed certify the corresponding properties (Theorem 3.24). As computing the maximal end components is necessary to derive the defining inequalities of Farkas certificates in the general case, Section 3.1.4 proposes a certifying algorithm for this problem. Finally, Section 3.2 discusses how similar results can be obtained for constraints on the total expected reward, and Section 3.3 is concerned with how to compute and validate Farkas certificates in practice.

Relation to published work

Farkas certificates were introduced in [FJB20], which is joint work with Florian Funke and Christel Baier, and has been published at TACAS 2020. This chapter extends the results of [FJB20] by also considering MDPs with proper end components (Section 3.1.3), discussing how the computation of maximal end components can be certified (Section 3.1.4) and considering the expected total reward criterion (Section 3.2). Section 3.3 on computing and validating Farkas certificates partly builds on [JHFB20], which is joint work with Hans Harder, Florian Funke and Christel Baier, and was published at FMCAD 2020. The discussion on how to use value- or policy-iteration to compute Farkas certificates is novel.

3.1 Farkas certificates for probabilistic reachability constraints

Throughout this section we will consider an MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ in reachability form (see Definition 2.5) with a single initial state s_{in}. The aim is to establish *certificates* for all types of probabilistic reachability constraints. These are statements of the following form, where $\lesssim \in \{\leq, <\}$, $\gtrsim \in \{\geq, >\}$ and $\lambda \in [0, 1]$:

I. All schedulers $\mathfrak{S}$ for $\mathcal{M}$ satisfy $\Pr^{\mathfrak{S}}_{s_{in}}(\lozenge \text{target}) \gtrsim \lambda$ (i.e., $\mathbf{Pr}^{\min}_{s_{in}}(\lozenge \text{target}) \gtrsim \lambda$).

II. All schedulers $\mathfrak{S}$ for $\mathcal{M}$ satisfy $\Pr^{\mathfrak{S}}_{s_{in}}(\lozenge \text{target}) \lesssim \lambda$ (i.e., $\mathbf{Pr}^{\max}_{s_{in}}(\lozenge \text{target}) \lesssim \lambda$).

III. Some scheduler $\mathfrak{S}$ for $\mathcal{M}$ satisfies $\Pr^{\mathfrak{S}}_{s_{in}}(\lozenge \text{target}) \gtrsim \lambda$ (i.e., $\mathbf{Pr}^{\max}_{s_{in}}(\lozenge \text{target}) \gtrsim \lambda$).

IV. Some scheduler $\mathfrak{S}$ for $\mathcal{M}$ satisfies $\Pr^{\mathfrak{S}}_{s_{in}}(\lozenge \text{target}) \lesssim \lambda$ (i.e., $\mathbf{Pr}^{\min}_{s_{in}}(\lozenge \text{target}) \lesssim \lambda$).

The basis of our construction is the LP-based characterization of the vectors $\mathbf{pr}^{\min}$ and $\mathbf{pr}^{\max}$ containing the optimal reachability probabilities in each state and, crucially, Farkas' Lemma. We will first consider the case that $\mathcal{M}$ is EC-free (Definition 2.6), and then show in Section 3.1.3 how one can handle arbitrary MDPs.

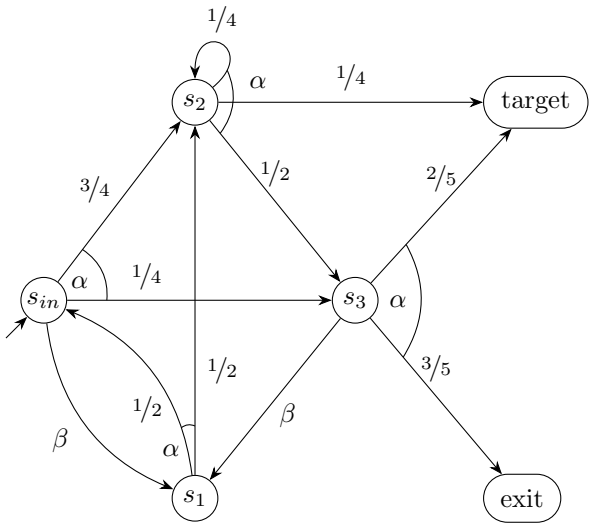


Figure 3.1: The MDP $\mathcal{M}_1$ as considered in Example 3.2.

3.1.1 End-component-free Markov decision processes

Certificates for the universal statements

To establish certificates for the statements (I.) and (II.), both of which require *all* schedulers of $\mathcal{M}$ to satisfy a certain probabilistic reachability constraint, we will use Lemma 2.9. It says that if a vector $\mathbf{z} \in \mathbb{R}^S$ satisfies $\mathbf{Az} \geq \mathbf{t}$, then $\mathbf{z}$ is a point-wise upper bound of $\mathbf{pr}^{\max}$. Furthermore, if it satisfies $\mathbf{Az} \leq \mathbf{t}$, then it is a point-wise lower bound of $\mathbf{pr}^{\min}$. The second statement requires the assumption that $\mathcal{M}$ is EC-free. This observation yields the following proposition.

Proposition 3.1. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an EC-free MDP in reachability form with system matrix $\mathbf{A}$ and target vector $\mathbf{t}$. Then for all $\gtrsim \in \{\geq, >\}$ and $\lesssim \in \{\leq, <\}$ we have:*

1. *There exists $\mathbf{z} \in \mathbb{R}^S$ satisfying $\mathbf{Az} \leq \mathbf{t}$ and $\mathbf{z}(s_{in}) \gtrsim \lambda$ if and only if $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ holds.*
2. *There exists $\mathbf{z} \in \mathbb{R}^S$ such that $\mathbf{Az} \geq \mathbf{t}$ and $\mathbf{z}(s_{in}) \lesssim \lambda$ if and only if $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target}) \lesssim \lambda$ holds.*

Proof. The linear programs characterizing the optimal reachability probabilities (see Section 2.2.2) use the systems of inequalities $\mathbf{Az} \leq \mathbf{t}$ and $\mathbf{Az} \geq \mathbf{t}$, and hence clearly the vectors $\mathbf{pr}^{\min}$ and $\mathbf{pr}^{\max}$ are solutions of the corresponding inequalities. This shows the direction from right to left. The other direction follows from Lemma 2.9. □

Proposition 3.1 provides a formulation of statements (I.) and (II.) which can be used to certify their validity, where the solution vectors $\mathbf{z}$ of the system of inequalities function as certificates. To check whether statement (I.) or (II.) holds, given a candidate certificate $\mathbf{z}$, one must merely check whether $\mathbf{z}$ is a solution of the correponding system of inequalities. While the vectors $\mathbf{pr}^{\min}$ and $\mathbf{pr}^{\max}$ are valid certificates for statements (I.) and (II.) respectively, provided that the property is satisfied in $\mathcal{M}$, many more vectors may also be.

Example 3.2. Consider the MDP $\mathcal{M}_1$ with states $S \cup \{\text{target}, \text{exit}\}$ as shown in Figure 3.1. Outgoing transitions belonging to the same action are drawn together and action labels are

indicated at the root of outgoing transitions or directly before the probabilities. For example, states s_1 and s_2 have a single enabled action α and states s_{in} and s_3 have two enabled actions α and β. Consider the vector

$$\mathbf{z}_1 = \big(s_{in} \mapsto 2/5,\quad s_1 \mapsto 2/5,\quad s_2 \mapsto 2/5,\quad s_3 \mapsto 2/5\big).$$

This vector can be used to verify that $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \geq 2/5$ holds by applying Proposition 3.1. We need to check that $\mathbf{A}\mathbf{z}_1 \leq \mathbf{t}$ holds, where $\mathbf{A}$ and $\mathbf{t}$ are the system matrix and target vector of $\mathcal{M}_1$. This amounts to checking the following condition for each enabled state-action pair (s, α):

$$\mathbf{z}_1(s) \;\leq\; \mathbf{t}(s) + \sum_{s' \in S} P(s, \alpha, s') \cdot \mathbf{z}_1(s').$$

In state s_{in} we have

$$\begin{aligned}
\alpha:&\quad \mathbf{z}_1(s_{in}) \;=\; 2/5 \;\leq\; 3/4 \cdot \mathbf{z}_1(s_2) + 1/4 \cdot \mathbf{z}_1(s_3) \;=\; 3/4 \cdot 2/5 + 1/4 \cdot 2/5 \;=\; 2/5, \quad \text{and}\\
\beta:&\quad \mathbf{z}_1(s_{in}) \;=\; 2/5 \;\leq\; 1 \cdot \mathbf{z}_1(s_1) \;=\; 2/5.
\end{aligned}$$

Similarly, this condition can be checked for all other states, which shows that $\mathbf{A}\mathbf{z}_1 \leq \mathbf{t}$ indeed holds. By Proposition 3.1, this implies $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \geq 2/5$. One can check that any vector $\mathbf{z}$ which assigns a constant value in the range $[0, 2/5]$ to all states in S satisfies the condition $\mathbf{A}\mathbf{z} \leq \mathbf{t}$. Another solution of this system of inequalities, which is a certificate for the stronger statement $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \geq 1/2$, is given by:

$$\mathbf{z}_2 = \big(s_{in} \mapsto 1/2,\quad s_1 \mapsto 1/2,\quad s_2 \mapsto 3/5,\quad s_3 \mapsto 2/5\big).$$

Let us now consider the system of inequalities $\mathbf{A}\mathbf{z} \geq \mathbf{t}$. A solution, which additionally satisfies $\mathbf{z}(s_{in}) \leq \lambda$, for some $\lambda \in [0, 1]$, certifies $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target}) \leq \lambda$. One can check that $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target}) = 1$ holds, and consequently, there exists no solution of $\mathbf{A}\mathbf{z} \geq \mathbf{t} \wedge \mathbf{z}(s_{in}) \leq \lambda$ for any $\lambda < 1$. △

Indicators on how many certificates exist (as measured by the volume of the set of valid certificates, for example) are the amount of nondeterminism (in the sense of multiple enabled actions in a single state) of the system and how far away the threshold λ is from the actual optimal value in s_{in}. If we consider a connected Markov chain $\mathcal{M}$ and set λ to be the actual probability of reaching "target" in $\mathcal{M}$, then the only vector satisfying the inequalities $\mathbf{A}\mathbf{z} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ is the vector containing the reachability probabilities $\mathbf{pr} = (\Pr_{\mathcal{M},s}(\Diamond\,\text{target}))_{s \in S}$.

Example 3.3. Consider the Markov chain $\mathcal{M}_2$ in Figure 3.2. The vector $\mathbf{pr} = (\Pr_{\mathcal{M}_2,s}(\Diamond\,\text{target}))_{s \in S}$ containing the probability to reach "target" for each state of $\mathcal{M}_2$ is:

$$\mathbf{pr} = \big(s_{in} \mapsto 8/15,\quad s_1 \mapsto 2/5,\quad s_2 \mapsto 2/3\big).$$

This vector is the only solution of $\mathbf{A}\mathbf{z} \leq \mathbf{t} \;\wedge\; \mathbf{z}(s_{in}) \geq 8/15$ (thereby providing a certificate for $\Pr_{\mathcal{M}}(\Diamond\,\text{target}) \geq 8/15$), and also the only solution of $\mathbf{A}\mathbf{z} \geq \mathbf{t} \;\wedge\; \mathbf{z}(s_{in}) \leq 8/15$ (thereby providing a certificate for $\Pr_{\mathcal{M}}(\Diamond\,\text{target}) \leq 8/15$). Intuitively, only a single solution exists in this case is because all states have to contribute exactly their probability in order to achieve the optimal value in s_{in}.

For MDPs, multiple solutions may exist even for such "tight" thresholds. To see this, consider

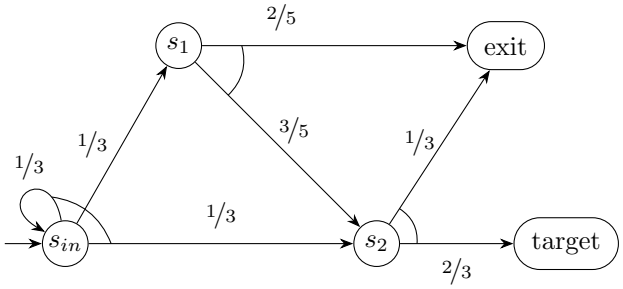


Figure 3.2: An example Markov chain $\mathcal{M}_2$, used in Example 3.3.

again the MDP $\mathcal{M}_1$ of Figure 3.1. The minimal reachability probability in states s_{in} and s_1 is $11/20$ and $23/40$ respectively. However, it is irrelevant for the value in s_{in} by how much the minimal probability in s_1 exceeds $11/20$. This is because in this case the minimizing action in s_{in} will be the one which does not visit s_1. The two following vectors are both solutions to $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq 11/20$:

$$\mathbf{z}_4 = \big(s_{in} \mapsto 11/20,\ \ s_1 \mapsto 11/20,\ \ s_2 \mapsto 3/5,\ \ s_3 \mapsto 2/5\big)$$
$$\mathbf{z}_5 = \big(s_{in} \mapsto 11/20,\ \ s_1 \mapsto 23/40,\ \ s_2 \mapsto 3/5,\ \ s_3 \mapsto 2/5\big) \qquad \triangle$$

Certificates for the existential statements

The next aim is to define certificates for statements (III.) and (IV.), both of which ask about the *existence* of a scheduler satisfying some threshold property. As for the cases (I.) and (II.) given in Proposition 3.1, we would like certificates to be solutions of some system of linear inequalities. To this end, we use the following observation: Every instance of the statements (III.) and (IV.) is equivalent to *the negation* of an instance of either statement (I.) or (II.). For example, a scheduler $\mathfrak{S}$ satisfying $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond \text{target}) \geq \lambda$ exists (III., with $\gtrsim = \geq$) if and only if it is not true that all schedulers $\mathfrak{S}$ satisfy $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond \text{target}) < \lambda$ (II., with $\lesssim = <$).

Proposition 3.1 provides formulations of statements (I.) and (II.) in terms of satisfiability of certain systems of linear inequalities. Hence the negations of these statements are equivalent to the unsatisfiability of the corresponding systems of linear inequalities. This is where we can use Farkas' Lemma. Broadly speaking, it tells us how to construct one set of linear inequalities from another one such that exactly one of them is satisfiable. In other words, the question of unsatisfiability of a set of linear inequalities can be reduced to the question of satisfiability of another set of linear inequalities, and thereby solutions of the latter are certificates for the unsatisfiability of the former.

There are two main obstacles in applying Farkas' Lemma directly. One is that many standard formulations (see Lemma 2.1) have a system of linear inequalities on one side, and a set of *equations* on the other. This is undesirable for our purposes because the set of certificates (defined by the equations) may be very restricted. To overcome this we will use a variant of Farkas' Lemma (Lemma 2.2) which is formulated using systems of inequalities on both sides but, on the other hand, includes a nonnegativity constraint on both sides. One can deal with this using the observation that if the systems of inequalities of Proposition 3.1 have a solution, then they have a nonnegative one due to their special structure. The second obstacle is the combination of strict and non-strict inequalities, which we overcome by reformulating the

systems of inequalities and applying Lemma 2.2 in both directions.

Proposition 3.4. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an EC-free MDP in reachability form with enabled state-action pairs $\mathcal{E}$, system matrix $\mathbf{A}$ and target vector $\mathbf{t}$. Then for $\gtrsim \in \{\geq, >\}$, $\lesssim \in \{\leq, <\}$ and $\lambda \in [0,1]$ we have:*

1. *There exists a row vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \geq \delta_{s_{in}}$ and $\mathbf{yt} \lesssim \lambda$ if and only if $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\, \text{target}) \lesssim \lambda$ holds.*
2. *There exists a row vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \leq \delta_{s_{in}}$ and $\mathbf{yt} \gtrsim \lambda$ if and only if $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\, \text{target}) \gtrsim \lambda$ holds.*

Proof. We first prove (1.) with $\lesssim = <$. If $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ has a solution, then it has a nonnegative one, namely the vector $\mathbf{pr}^{\min}$. Using this observation together with Proposition 3.1 yields:

$$\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\, \text{target}) < \lambda \quad \Longleftrightarrow \quad \neg\, \exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda.$$

Let us assume that the first column of $\mathbf{A}$ (resp. the first row of $\mathbf{z}$) corresponds to the state s_{in}. Then, transforming the latter statement into matrix notation and applying Farkas' Lemma (Lemma 2.2) from left to right yields:

$$\begin{aligned} &\neg\, \exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \begin{pmatrix} \mathbf{A} \\ -1\ 0 \ldots 0 \end{pmatrix} \mathbf{z} \ \leq\ \begin{pmatrix} \mathbf{t} \\ -\lambda \end{pmatrix} \\ \Longleftrightarrow \quad &\exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}\ \exists y^* \geq 0.\ \ (\mathbf{y}, y^*) \begin{pmatrix} \mathbf{A} \\ -1\ 0 \ldots 0 \end{pmatrix} \ \geq\ \mathbf{0} \ \ \wedge \ \ (\mathbf{y}, y^*) \begin{pmatrix} \mathbf{t} \\ -\lambda \end{pmatrix} \ <\ 0. \qquad (\dagger) \end{aligned}$$

The statement (†) is equivalent to the left hand side of (1.), which can be seen as follows. First, (†) can be equivalently written as $\mathbf{yA} \geq (y^*, 0, \ldots, 0) \wedge \mathbf{yt} < y^*\lambda$. If there exists $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} < \lambda$, then this vector $\mathbf{y}$ together with $y^* = 1$ is a solution of (†). For the other direction, if $(\mathbf{y}, y^*)$ is a solution of (†), then $\mathbf{y}' = 1/y^* \cdot \mathbf{y}$ is nonnegative and satisfies $\mathbf{y}'\mathbf{A} \geq \delta_{s_{in}} \wedge \mathbf{y}'\mathbf{t} < \lambda$. Observe that $y^* > 0$ holds in this case, as $\mathbf{yt} < y^*\lambda$ holds and all of $\mathbf{t}, \mathbf{y}$ and λ are nonnegative.

Now we prove (1.) with $\lesssim = \leq$. As before we have (by Proposition 3.1):

$$\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\, \text{target}) \leq \lambda \quad \Longleftrightarrow \quad \neg\, \exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) > \lambda.$$

We claim that the latter statement is equivalent to

$$\neg\, \exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}\ \exists z^* \geq 0.\ \ \begin{pmatrix} -\mathbf{A} & \mathbf{t} \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ z^* \end{pmatrix} \geq \mathbf{0} \wedge \begin{pmatrix} -\delta_{s_{in}} & \lambda \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ z^* \end{pmatrix} < 0. \qquad (\ddagger)$$

A solution of $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) > \lambda$ yields a solution of (‡) by setting $z^* = 1$. For the other direction, let $(\mathbf{z}, z^*)$ be a solution of (‡). We make a case distinction on whether $z^* = 0$ holds. If $z^* > 0$, then it again suffices to choose $\mathbf{z}' = 1/z^* \cdot \mathbf{z}$, as we then have $\mathbf{Az}' \leq \mathbf{t} \wedge \mathbf{z}'(s_{in}) > \lambda$ and $\mathbf{z}' \geq 0$. If $z^* = 0$, then $\mathbf{z}$ satisfies $\mathbf{Az} \leq \mathbf{0}$ and $\mathbf{z}(s_{in}) > z^*\lambda = 0$. Pick a $\gamma > \lambda/\mathbf{z}(s_{in})$ and set $\mathbf{z}' = \gamma\mathbf{z}$. We have $\mathbf{Az}' \leq \mathbf{0} \leq \mathbf{t}$ and $\mathbf{z}'(s_{in}) = \gamma\mathbf{z}(s_{in}) > \lambda$.

Applying Lemma 2.2 from right to left to (‡) yields the equivalent statement

$$\exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}.\ \ \mathbf{y} \begin{pmatrix} -\mathbf{A} & \mathbf{t} \end{pmatrix} \leq \begin{pmatrix} -\delta_{s_{in}} & \lambda \end{pmatrix}.$$

This can now be rewritten into $\exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}.\ \ \mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \leq \lambda$, which is the left hand side of (1.).

Case (2.) can be proved analogously, and we only summarize the chain of calculations here for completeness. We start with the case $\gtrsim = >$.

$$\begin{aligned}
&\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) > \lambda \quad \overset{\text{Prop. 3.1}}{\iff} \quad \neg\,\exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \mathbf{A}\mathbf{z} \geq \mathbf{t} \ \wedge\ \mathbf{z}(s_{in}) \leq \lambda \\
\iff\ & \neg\,\exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \begin{pmatrix} -\mathbf{A} \\ 1\,0\ldots 0 \end{pmatrix}\mathbf{z} \ \leq\ \begin{pmatrix} -\mathbf{t} \\ \lambda \end{pmatrix} \\
\overset{\text{Lem. 2.2}}{\iff}\ & \exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}\ \exists y^* \geq 0.\ \ (\mathbf{y}, y^*)\begin{pmatrix} -\mathbf{A} \\ 1\,0\ldots 0 \end{pmatrix} \geq \mathbf{0} \quad \wedge \quad (\mathbf{y}, y^*)\begin{pmatrix} -\mathbf{t} \\ \lambda \end{pmatrix} \ <\ 0 \\
\iff\ & \exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}.\ \mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} > \lambda
\end{aligned}$$

For $\gtrsim = \geq$ we calculate:

$$\begin{aligned}
&\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda \quad \overset{\text{Prop. 3.1}}{\iff} \quad \neg\,\exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}.\ \ \mathbf{A}\mathbf{z} \geq \mathbf{t} \quad \wedge \quad \mathbf{z}(s_{in}) < \lambda \\
\iff\ & \neg\,\exists \mathbf{z} \in \mathbb{R}^{S}_{\geq 0}\ \exists z^* \geq 0.\ \ \begin{pmatrix} \mathbf{A} & -\mathbf{t} \end{pmatrix}\begin{pmatrix} \mathbf{z} \\ z^* \end{pmatrix} \geq 0 \wedge \begin{pmatrix} \delta_{s_{in}} & -\lambda \end{pmatrix}\begin{pmatrix} \mathbf{z} \\ z^* \end{pmatrix} < 0 \\
\overset{\text{Lem. 2.2}}{\iff}\ & \exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}.\ \ \mathbf{y}\begin{pmatrix} \mathbf{A} & -\mathbf{t} \end{pmatrix} \leq \begin{pmatrix} \delta_{s_{in}} & -\lambda \end{pmatrix} \iff \exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}.\ \mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \geq \lambda \qquad \square
\end{aligned}$$

The above shows that solutions of the system of linear inequalities $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \gtrsim \lambda$ yield certificates for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) \gtrsim \lambda$ (statement III.), and solutions of $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \lesssim \lambda$ yield certificates for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) \lesssim \lambda$ (statement IV.). We will call the vectors satisfying Propositions 3.1 and 3.4 *Farkas certificates* for the corresponding probabilistic reachability constraints.

Example 3.5. Consider again the MDP $\mathcal{M}_1$ in Figure 3.1. The vector $\mathbf{y}_1$ defined by

$$\mathbf{y}_1 = \big((s_{in}, \alpha) \mapsto 2,\ (s_{in}, \beta) \mapsto 0,\ (s_1, \alpha) \mapsto 2,\ (s_2, \alpha) \mapsto 3,\ (s_3, \alpha) \mapsto 0,\ (s_3, \beta) \mapsto 2\big)$$

is a solution of the system of inequalities $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \geq 1/2$. To verify this one has to check that the following inequality holds for each state s:

$$\sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s, \alpha) \ \leq\ \delta_{s_{in}}(s) + \sum_{(s', \alpha) \in \mathcal{E}} P(s', \alpha, s_{in}) \cdot \mathbf{y}(s', \alpha).$$

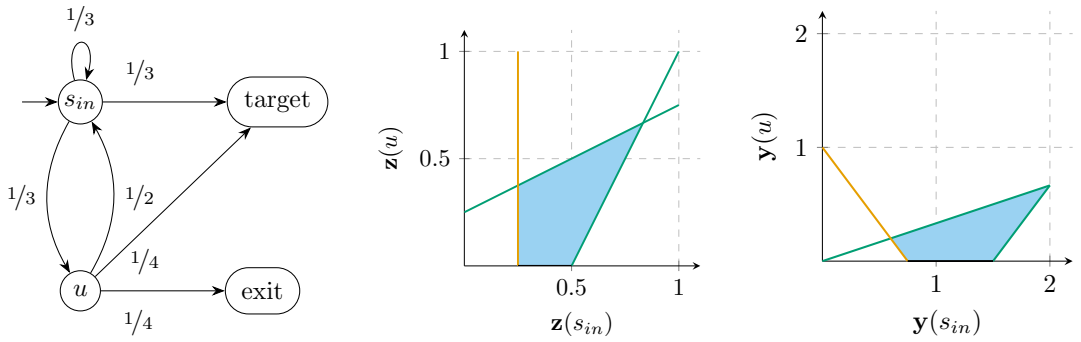
For example, for state s_{in} and vector $\mathbf{y}_1$ we have $\sum_{\alpha \in \mathrm{Act}(s_{in})} \mathbf{y}_1(s_{in}, \alpha) = \mathbf{y}_1(s_{in}, \alpha) = 2$ and

$$\delta_{s_{in}}(s_{in}) + \sum_{(s', \alpha) \in \mathcal{E}} P(s', \alpha, s) \cdot \mathbf{y}_1(s', \alpha) \ =\ 1 + 1/2 \cdot \mathbf{y}_1(s_1, \alpha) \ =\ 1 + 1 \ =\ 2.$$

For state s_2 we have $\sum_{\alpha \in \mathrm{Act}(s_2)} \mathbf{y}_1(s_2, \alpha) = 3$ and

$$\begin{aligned}
\delta_{s_{in}}(s_2) + \sum_{(s', \alpha) \in \mathcal{E}} P(s', \alpha, s_2) \cdot \mathbf{y}_1(s', \alpha) \ &=\ 0 + 3/4 \cdot \mathbf{y}_1(s_{in}, \alpha) + 1/2 \cdot \mathbf{y}_1(s_1, \alpha) + 1/4 \cdot \mathbf{y}_1(s_2, \alpha) \\
&=\ 3/2 + 1 + 3/4 \ =\ 13/4 \geq 3.
\end{aligned}$$

The corresponding inequality in state s_3 can be checked similarly. Additionally to the above constraints, $\sum_{(s,\alpha) \in \mathcal{E}} \mathbf{y}_1(s, \alpha) \cdot \mathbf{t}(s, \alpha) \ \geq\ 1/2$ must hold. We have $\sum_{(s,\alpha) \in \mathcal{E}} \mathbf{y}_1(s, \alpha) \cdot \mathbf{t}(s, \alpha) =$



(a) (b) (c)

Figure 3.3: A Markov chain (a) together with the polyhedra defining certain sets of Farkas certificates. If $\mathbf{A}, \mathbf{t}$ are system matrix and target vector of the Markov chain in (a), then (b) shows the individual inequalities given by $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ and indicates their intersection, for $\lambda = 1/4$. Similarly, (c) shows the inequalities defining $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$. The inequalities which include λ are drawn in orange. See also Example 3.7.

$\mathbf{y}_1(s_2, \alpha) \cdot 1/4 = 3/4 \geq 1/2$. As $\mathbf{y}_1$ is a solution of $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq 1/2$, it is a certificate for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 1/2$ by Proposition 3.4.

Another vector satisfying this system of inequalities is $\mathbf{y}_2$, defined by

$$\mathbf{y}_2 = \big((s_{in}, \alpha) \mapsto 0,\ (s_{in}, \beta) \mapsto 4,\ (s_1, \alpha) \mapsto 6,\ (s_2, \alpha) \mapsto 4,\ (s_3, \alpha) \mapsto 0,\ (s_3, \beta) \mapsto 2\big).$$

The stronger constraint $\mathbf{y}_2\mathbf{t} = 1$ is satisfied here, which certifies $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 1$. △

For Markov chains $\mathcal{M}$, we have $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target})$, and hence certificates for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \bowtie \lambda$ and certificates for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \bowtie \lambda$ can be interchanged. In this case the system matrix $\mathbf{A}$ is a square matrix, as there exists as many states as state-action pairs, and hence solutions $\mathbf{z}$ of $\mathbf{Az} \bowtie \mathbf{t}$ and solutions $\mathbf{y}$ of $\mathbf{yA} \bowtie \delta_{s_{in}}$ have the same dimension. However, the two types of certificates do differ also for Markov chains, as shown in the following example.

Example 3.6. Consider the Markov chain $\mathcal{M}_2$ in Figure 3.2, and let $\mathbf{A}, \mathbf{t}$ be its system matrix and target vector. One can check that the vector

$$\mathbf{z} = \big(s_{in} \mapsto 1/3,\ s_1 \mapsto 2/5,\ s_2 \mapsto 2/3\big)$$

is a solution of $\mathbf{Az} \leq \mathbf{t}$. This vector is not a solution of $\mathbf{yA} \leq \delta_{s_{in}}$, which in particular requires $\mathbf{y}(s_1) \leq 1/3 \cdot \mathbf{y}(s_{in})$. On the other hand, the vector

$$\mathbf{y} = \big(s_{in} \mapsto 1,\ s_1 \mapsto 1/3,\ s_2 \mapsto 1/2\big)$$

is a solution of $\mathbf{yA} \leq \delta_{s_{in}}$. This vector is not a solution of $\mathbf{Az} \leq \mathbf{t}$, which in particular requires $\mathbf{z}(s_1) \leq 3/5 \cdot \mathbf{z}(s_2)$. We have $\mathbf{z}(s_{in}) \geq 1/3$ and $\mathbf{yt} \geq 1/3$, and hence both vectors are valid certificates for $\Pr_{\mathcal{M}}(\Diamond \text{target}) \geq 1/3$, using Proposition 3.1 and Proposition 3.4 respectively. △

Example 3.7. Consider the Markov chain shown in Figure 3.3a and let $\mathbf{A}, \mathbf{t}$ be its system matrix

and target vector. Spelling out the inequalities $\mathbf{A}\mathbf{z} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq 1/4$ yields

$$\mathbf{z}(s_{in}) \leq 1/3 + 1/3 \cdot \mathbf{z}(s_{in}) + 1/3 \cdot \mathbf{z}(u) \qquad \mathbf{z}(u) \leq 1/4 + 1/2 \cdot \mathbf{z}(s_{in}) \qquad \mathbf{z}(s_{in}) \geq 1/4.$$

The system of linear inequalities $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \geq 1/4$ is:

$$\mathbf{y}(s_{in}) \leq 1 + 1/3 \cdot \mathbf{y}(s_{in}) + 1/2 \cdot \mathbf{y}(u) \qquad \mathbf{y}(u) \leq 1/3 \cdot \mathbf{y}(s_{in}) \qquad 1/3 \cdot \mathbf{y}(s_{in}) + 1/4 \cdot \mathbf{y}(u) \geq 1/4.$$

These systems of linear inequalities define two polyhedra in $\mathbb{R}^2_{\geq 0}$, which are depicted in Figure 3.3b and Figure 3.3c. △

Before turning to MDPs which are not EC-free, we will study some properties of the solutions of the systems of inequalities considered in the above propositions. In particular, we will show how solutions of the systems of inequalities $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}}$ and $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}}$ relate to the expected number of visits under schedulers which do not realize any proper end components (for EC-free MDPs, all schedulers satisfy this property).

We start by observing that there is a correspondence between solutions of $\mathbf{y}\mathbf{A} \leq \mathbf{0}$ and proper end components of $\mathcal{M}$. This will turn out to be useful at several places later. For EC-free MDPs, this result can be used to show boundedness of some of the sets of Farkas certificates (see Proposition 3.9). To make the correspondence precise, we say that a set of enabled state-action pairs $\mathcal{E}' \subseteq \mathcal{E}$ *induces proper end components* if there exists a partition $S_1, \ldots, S_k$ of the set $\{s \mid \exists \alpha.\ (s,\alpha) \in \mathcal{E}'\}$ and, for all $1 \leq i \leq k$, a function $A_i : S_i \to 2^{\mathrm{Act}}$ such that (S_i, A_i) is a proper end component and for all $s \in S_i$ and $\alpha \in A_i(s)$ we have $(s,\alpha) \in \mathcal{E}'$.

Lemma 3.8. *Let $\mathcal{M} = (S \cup \{\mathrm{target}, \mathrm{exit}\}, \mathrm{Act}, s_{in}, P)$ be an MDP with system matrix $\mathbf{A}$ and enabled state-action pairs $\mathcal{E}$. Then:*

1. *For all $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$: If $\mathbf{y}\mathbf{A} \leq \mathbf{0}$ holds, then $\mathrm{supp}(\mathbf{y})$ induces proper end components in $\mathcal{M}$.*
2. *If $\mathcal{E}' \subseteq \mathcal{E}$ induces proper end components, then there exists $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ such that $\mathbf{y}\mathbf{A} = \mathbf{0}$ and $\mathrm{supp}(\mathbf{y}) = \mathcal{E}'$.*

Proof. (1.) First we show that $\mathbf{y}\mathbf{A} \leq \mathbf{0}$ implies $\mathbf{y}\mathbf{A} = \mathbf{0}$ for all $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$. If $\mathbf{y}\mathbf{A} \leq \mathbf{0}$ holds, then so does $\mathbf{y}\,\mathbf{A}\,\mathbf{1} \leq 0$. By construction we have $\mathbf{A} \cdot \mathbf{1} \geq \mathbf{0}$, as $\mathbf{A} \cdot \mathbf{1}$ is the vector containing $1 - \sum_{s' \in S} P(s,\alpha,s')$ for each $(s,\alpha) \in \mathcal{E}$. Hence it follows from $\mathbf{y} \geq \mathbf{0}$ that $\mathbf{y}\,\mathbf{A}\,\mathbf{1} \geq 0$ holds. But then $\mathbf{y}\,\mathbf{A}\,\mathbf{1} = 0$ follows and hence also $\mathbf{y}\mathbf{A} = \mathbf{0}$. From $\mathbf{y}\,\mathbf{A}\,\mathbf{1} = 0$ it also follows that whenever $\mathbf{y}(s,\alpha) > 0$ holds, then we have $\sum_{s' \in S} P(s,\alpha,s') = 1$ and therefore $P(s,\alpha,\mathrm{target}) = P(s,\alpha,\mathrm{exit}) = 0$.

For all $q, q' \in S$ let $v(q,q') = \sum_{\alpha \in \mathrm{Act}(q)} P(q,\alpha,q') \cdot \mathbf{y}(q,\alpha)$. Consider the directed graph $G = (S, E)$ in which $(q,q') \in E$ holds whenever $v(q,q') > 0$ is true. For all $q \in S$ we get

$$\begin{aligned}
\sum_{q' \in S} v(q',q) &= \sum_{q' \in S} \sum_{\alpha \in \mathrm{Act}(q')} P(q',\alpha,q) \cdot \mathbf{y}(q',\alpha) \overset{(\dagger)}{=} \sum_{\alpha \in \mathrm{Act}(q)} \mathbf{y}(q,\alpha) \\
&\overset{(\ddagger)}{=} \sum_{\alpha \in \mathrm{Act}(q)} \mathbf{y}(q,\alpha) \cdot \sum_{q' \in S} P(q,\alpha,q') = \sum_{q' \in S} \sum_{\alpha \in \mathrm{Act}(q)} P(q,\alpha,q') \cdot \mathbf{y}(q,\alpha) = \sum_{q' \in S} v(q,q').
\end{aligned}$$

In the above calculation the equivalence (†) follows from $\mathbf{y}\mathbf{A} = \mathbf{0}$, and (‡) follows from the fact that $\sum_{q' \in S} P(q,\alpha,q') = 1$ holds for all (q,α) satisfying $\mathbf{y}(q,\alpha) > 0$, which was argued above. If we interpret G as a weighted graph with edge weights $v(q,q')$ for all $(q,q') \in E$, then the above expresses a flow constraint. Namely, that the total weight of incoming edges should equal the total weight of outgoing edges for all vertices.

Consider a strongly connected component C of G which has no incoming edges from outside of C. The total weights on outgoing edges from C-states equals the total weights of incoming edges to C-states. As C has no incoming edges from outside of C, we have

$$\sum_{q\in C}\sum_{q'\in S} v(q,q') \;=\; \sum_{q\in C}\sum_{q'\in S} v(q',q) \;=\; \sum_{q\in C}\sum_{q'\in C} v(q',q).$$

Hence, the weight of edges from C to a state outside of C is zero which means that there is no such edge by definition of G. As a consequence, all SCCs of G are disjoint. Let $S_1,\dots,S_k$ be the sets of states in S which induce non-trivial SCCs in G (that is, containing at least one edge). For each S_i let $A_i : S_i \to 2^{\mathrm{Act}}$ be defined by $A_i(s) = \{\alpha \in \mathrm{Act}(s) \mid \mathbf{y}(s,\alpha) > 0\}$. By construction of G, the pair (S_i, A_i) forms an end component of $\mathcal{M}$, which concludes the proof.

(2.) Suppose that $\mathcal{E}'$ induces the proper end components $(S_1,A_1),\dots,(S_k,A_k)$. Let $\mathcal{E}_i = \{(s,\alpha) \mid s \in S_i \text{ and } \alpha \in A_i(s)\}$, i.e., the state-action pairs which form the end component (S_i,A_i). We show that there exists $\mathbf{y}_i$ satisfying $\mathbf{y}_i\mathbf{A} = \mathbf{0}$ and $\mathrm{supp}(\mathbf{y}_i) = \mathcal{E}_i$ for all $1 \le i \le k$. It follows that $\mathbf{y} = \sum_{1\le i\le k} \mathbf{y}_i$ satisfies $\mathbf{y}\mathbf{A} = \mathbf{0}$ and $\mathrm{supp}(\mathbf{y}) = \mathcal{E}'$, which is what we need to show.

So let $1 \le i \le k$ and define $a_s = |A_i(s)|$ for each $s \in S_i$, which is the number of enabled actions in s in the end component (S_i,A_i). Consider the matrix $M \in \mathbb{R}^{S_i\times S_i}$ defined by $M_{s,s'} = {}^1\!/_{a_s} \cdot \sum_{\alpha\in A_i(s)} P(s,\alpha,s')$. As (S_i,A_i) is a proper end component, the matrix M is the probability matrix of an irreducible Markov chain. Hence, there exists a unique row vector $\pi \in \mathbb{R}^{S_i}$ satisfying $\pi\cdot M = \pi$, and π additionally satisfies $\pi(s) > 0$ for all $s \in S_i$ (see [KS76, Theorems 4.1.4 and 4.1.6]). Let $\mathbf{y}_i(s,\alpha) = {}^1\!/_{a_s}\cdot\pi(s)$ for all $s\in S_i$ and $\alpha \in A_i(s)$, and zero otherwise. Then for all $s \in S_i$ we have

$$\begin{aligned}\sum_{\alpha\in\mathrm{Act}(s)} \mathbf{y}_i(s,\alpha) \;&=\; \sum_{\alpha\in A_i(s)} \frac{1}{a_s}\cdot\pi(s) \;=\; \pi(s) \;\overset{(\dagger)}{=}\; \sum_{s'\in S_i}\frac{1}{a_{s'}}\cdot \sum_{\alpha\in\mathrm{Act}(s')} P(s',\alpha,s)\cdot\pi(s')\\ &=\; \sum_{s'\in S}\sum_{\alpha\in\mathrm{Act}(s')} P(s',\alpha,s)\cdot \mathbf{y}_i(s',\alpha)\end{aligned}$$

The statement (†) follows from $\pi\cdot M = \pi$. The equality between the first term and the last is exactly what is required to prove $\mathbf{y}_i\mathbf{A} = \mathbf{0}$ and we have $\mathrm{supp}(\mathbf{y}_i) = \mathcal{E}_i$. This concludes the proof. □

Using this lemma one can prove that the sets of solutions of both systems of inequalities $\mathbf{A}\mathbf{z} \le \mathbf{t}$ and $\mathbf{y}\mathbf{A} \le \delta_{s_{in}}$ are bounded, given that the MDP under consideration is EC-free.

Proposition 3.9. *Let $\mathcal{M} = (S \cup \{\mathrm{target},\mathrm{exit}\}, \mathrm{Act}, s_{in}, P)$ be an EC-free MDP with enabled state-action pairs $\mathcal{E}$, system matrix $\mathbf{A}$ and target vector $\mathbf{t}$.*

Then the sets $\{\mathbf{z}\in\mathbb{R}^S_{\ge 0} \mid \mathbf{A}\mathbf{z}\le\mathbf{t}\}$ and $\{\mathbf{y}\in\mathbb{R}^{\mathcal{E}}_{\ge 0} \mid \mathbf{y}\mathbf{A}\le\delta_{s_{in}}\}$ are both bounded.

Proof. The set $\{\mathbf{z}\in\mathbb{R}^S_{\ge 0} \mid \mathbf{A}\mathbf{z}\le\mathbf{t}\}$ is bounded because, by Lemma 2.9, $\mathbf{pr}^{\min}$ is a point-wise upper bound on all vectors $\mathbf{z}$ satisfying $\mathbf{A}\mathbf{z}\le\mathbf{t}$.

Now assume that the set $\mathcal{U} = \{\mathbf{y}\in\mathbb{R}^{\mathcal{E}}_{\ge 0} \mid \mathbf{y}\mathbf{A}\le\delta_{s_{in}}\}$ is unbounded. Then there exists $\mathbf{y}_0,\mathbf{y}_1$ such that $\mathbf{y}_1 \ne \mathbf{0}$ and for all $t\ge 0$ we have $\mathbf{y}_0 + t\,\mathbf{y}_1 \in \mathcal{U}$. It follows that $\mathbf{y}_1 \ge \mathbf{0}$, $t\cdot\mathbf{y}_1\mathbf{A} \le \delta_{s_{in}}$ for all $t\ge 0$ and $\mathrm{supp}(\mathbf{y}_1)\subseteq\mathcal{E}'$. This implies that $\mathbf{y}_1\mathbf{A}\le\mathbf{0}$ must hold. By Lemma 3.8, $\mathrm{supp}(\mathbf{y}_1)$ induces proper end components, and as $\mathbf{y}_1\ne\mathbf{0}$ holds, this set is not empty. But this contradicts our assumption that $\mathcal{M}$ is EC-free. □

Remark 3.10. The statement on boundedness of $\{\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{yA} \leq \delta_{s_{in}}\}$ in the above lemma is very similar to the statement of boundedness of a related linear program in [Kal83, Theorem 3.2.4]. While the theorem in [Kal83] is stated with the precondition that the initial distribution is strictly positive in each state, the argument remains the same. △

Remark 3.11. The sets $\{\mathbf{z} \in \mathbb{R}^{S}_{\geq 0} \mid \mathbf{Az} \geq \mathbf{t} \wedge \mathbf{z}(s_{in}) \leq \lambda\}$ and $\{\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \leq \lambda\}$ are not bounded in general. For the first set, consider an MDP $\mathcal{M}$ in reachability form and a state s which has no incoming transitions. If $\mathbf{z}$ is a solution of $\mathbf{Az} \geq \mathbf{t}$ and $\mathbf{z}(s_{in}) \leq \lambda$, then multiplying the entry $\mathbf{z}(s)$ by any positive number yields another solution. For the second set, let $\mathcal{M}$ be the simple MDP in which the initial state is the only state and has one action α leading to "target" with probability one, and one action β leading to "exit" with probability one. Then the vector $\big((s_{in}, \alpha) \mapsto 0,\ (s_{in}, \beta) \mapsto 1\big)$ is a solution of $\mathbf{yA} \geq \delta_{s_{in}}$ and $\mathbf{yt} \leq 0$. Furthermore, any positive multiple of that vector remains a solution of this system of inequalities. △

First and foremost, Farkas certificates are objects which provide simple proofs of the corresponding property. We will now discuss how they can be interpreted within the given MDP. The Farkas certificates defined in Proposition 3.1 are solutions of the systems of inequalities $\mathbf{Az} \leq \mathbf{t}$ and $\mathbf{Az} \geq \mathbf{t}$ respectively, and represent point-wise bounds on $\mathbf{pr}^{\min}$ and $\mathbf{pr}^{\max}$ by Lemma 2.9. In the following we discuss an interpretation of Farkas certificates defined as solutions of the inequalities $\mathbf{yA} \leq \delta_{s_{in}}$ and $\mathbf{yA} \geq \delta_{s_{in}}$. It was shown in Proposition 3.4 that they certify the *existence* of a scheduler satisfying certain threshold properties. Now we will discuss how such a scheduler can be computed from a given Farkas certificate, and vice versa.

3.1.2 Farkas certificates and expected number of visits

As before, let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, let $\mathbf{A}$ be its system matrix, $\mathbf{t}$ its target vector and $\mathcal{E}$ its enabled state-action pairs. The systems of linear inequalities used in Proposition 3.4 are either of the form $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$ or $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \gtrsim \lambda$. Consider the *equation system* $\mathbf{yA} = \delta_{s_{in}}$. Spelling it out yields:

$$\sum_{\alpha \in \text{Act}(s)} \mathbf{y}(s, \alpha) \quad = \quad \delta_{s_{in}}(s) + \sum_{(s', \alpha) \in \mathcal{E}} P(s', \alpha, s) \cdot \mathbf{y}(s', \alpha) \quad \text{for all } s \in S. \tag{3.1}$$

This equation can be interpreted as a "flow equation". It says that the value of a state s (represented by sum of the values of its enabled state-action pairs) should equal the weighted sum of the values of its "predecessor" state-action pairs, i.e., those which have positive probability to move to s. For Markov chains, where each state has exactly one enabled action, this is exactly the equation system characterizing the expected number of visits of a transient state (see Lemma 2.15).

Remark 3.12. The correspondence between solutions of Equation (3.1) and schedulers of $\mathcal{M}$ has been studied before, in particular by Kallenberg [Kal83, Kal94, Kal16]. It is shown in [Kal83, Theorem 3.3.3] that the solutions of Equation (3.1) are in one-to-one correspondence with the memoryless schedulers of $\mathcal{M}$ which reach $\{\text{target}, \text{exit}\}$ with probability one (called *transient stationary policies* in [Kal83]). The mapping is the one that we will also use. However, the one-to-one correspondence depends on the fact that the initial distribution is strictly positive in each state (see [Kal83, Remark 3.3.9]). As we drop this restriction, we will be concerned with states that are not reachable under a given scheduler (given the initial distribution) in the following. Furthermore, we will also consider variants of Equation (3.1) using inequalities, rather

than equalities. It should also be pointed out that the interpretation of solutions of Equation (3.1) as the expected number of visits under the corresponding scheduler is well-known (see [Kal83, Equation 3.3.12]). △

This interpretation can be generalized to MDPs, where some additional care has to be taken with respect to proper end components (a scheduler realizing a proper end component has infinite expected number of visits in the corresponding states) and states which become unreachable under a given scheduler. The following two propositions make the correspondence between solutions of $\mathbf{yA} = \delta_{s_{in}}$ and certain schedulers for $\mathcal{M}$ precise. Here, the vector $\mathbf{ev}^{\mathfrak{S}} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ contains the expected number of visits of all enabled state-action pairs in $\mathcal{M}$ under scheduler $\mathfrak{S}$ when starting in s_{in} (see Section 2.2.4 for a precise definition). Henceforth we will use the notation $\mathbf{y}(s) = \sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s, \alpha)$ for vectors of the form $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}$ and states $s \in S$.

Proposition 3.13. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \mathrm{Act}, s_{in}, P)$ be an MDP, $\mathcal{E}$ its enabled state-action pairs and $\mathbf{A}$ its system matrix. Furthermore, let $\mathfrak{S}$ be a memoryless scheduler for $\mathcal{M}$ satisfying $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$, and let $R \subseteq S$ be the states in S reachable in $\mathcal{M}$ from s_{in} under $\mathfrak{S}$.*

Then, $\mathbf{ev}^{\mathfrak{S}}$ is the unique solution of $\mathbf{yA} = \delta_{s_{in}}$ which satisfies

- $\mathbf{y}(s, \alpha) = \mathbf{y}(s) \cdot \mathfrak{S}(s, \alpha)$ *for all $s \in R$ and $\alpha \in \mathrm{Act}(s)$, and*
- $\mathbf{y}(s, \alpha) = 0$ *for all $(s, \alpha) \in \mathcal{E}$ with $s \in S \setminus R$.*

Proof. Let $\mathbf{ev}^{\mathfrak{S}}(s) = \sum_{(s,\alpha) \in \mathcal{E}} \mathbf{ev}^{\mathfrak{S}}(s, \alpha)$ denote the expected number of visits of state $s \in S$ under $\mathfrak{S}$ in $\mathcal{M}$ before reaching $\{\text{target}, \text{exit}\}$. Consider the Markov chain $\mathcal{M}_R = (R \cup \{\text{target}, \text{exit}\}, s_{in}, P_R)$, where $P_R(s, t) = \sum_{\alpha \in \mathrm{Act}(s)} \mathfrak{S}(s, \alpha) \cdot P(s, \alpha, t)$. Due to our assumptions, R is the transient part of $\mathcal{M}_R$, and, by construction, the expected number of visits of $s \in R$ in $\mathcal{M}_R$ when starting in s_{in} is given by $\mathbf{ev}^{\mathfrak{S}}(s)$. Hence, $\left(\mathbf{ev}^{\mathfrak{S}}(s)\right)_{s \in S} \in \mathbb{R}^S$ is the unique vector satisfying

$$\mathbf{ev}^{\mathfrak{S}}(s) = \begin{cases} \delta_{s_{in}}(s) + \sum_{s' \in R} \sum_{\alpha \in \mathrm{Act}(s')} P(s', \alpha, s) \cdot \mathfrak{S}(s', \alpha) \cdot \mathbf{ev}^{\mathfrak{S}}(s') & \text{if } s \in R \\ 0 & \text{if } s \in S \setminus R. \end{cases} \tag{$\dagger$}$$

Here we have used Lemma 2.15 and the fact that $\sum_{\alpha \in \mathrm{Act}(s')} P(s', \alpha, s) \cdot \mathfrak{S}(s', \alpha)$ is the probability to move from s' to s in the Markov chain $\mathcal{M}_R$. Observe that we have $\mathbf{ev}^{\mathfrak{S}}(s, \alpha) = \mathbf{ev}^{\mathfrak{S}}(s) \cdot \mathfrak{S}(s, \alpha)$, as the expected number of times that (s, α) is seen equals the expected number of times that s is visited times the probability of choosing α in s. Hence we have

$$\mathbf{ev}^{\mathfrak{S}}(s) \;=\; \delta_{s_{in}}(s) + \sum_{(s', \alpha) \in \mathcal{E}} P(s', \alpha, s) \cdot \mathbf{ev}^{\mathfrak{S}}(s', \alpha),$$

for all $s \in S$, which, in matrix form, equals the equation system $\mathbf{ev}^{\mathfrak{S}} \cdot \mathbf{A} = \delta_{s_{in}}$. Here we have used that if $s \in S \setminus R$, then $s \neq s_{in}$ and for all s' such that $\mathfrak{S}(s', \alpha) \cdot P(s', \alpha, s) > 0$ for some $\alpha \in \mathrm{Act}(s')$ we have $s' \in S \setminus R$.

Now suppose there existed a different vector $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}$ satisfying $\mathbf{y}'\mathbf{A} = \delta_{s_{in}}$, $\mathbf{y}'(s, \alpha) = \mathbf{y}'(s) \cdot \mathfrak{S}(s, \alpha)$ for all $s \in R$, and $\mathbf{y}'(s, \alpha) = 0$ otherwise. Then $\left(\mathbf{y}'(s)\right)_{s \in S}$ is a solution of ($\dagger$) which differs from $\left(\mathbf{ev}^{\mathfrak{S}}(s)\right)_{s \in S}$, but this contradicts the fact that ($\dagger$) has a unique solution. □

This shows that for each memoryless scheduler $\mathfrak{S}$ satisfying $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$, we find a corresponding solution of $\mathbf{yA} = \delta_{s_{in}}$.

Example 3.14. Consider the MDP $\mathcal{M}_1$ from Figure 3.1 and the vector

$$\mathbf{y}_2 = \big((s_{in}, \alpha) \mapsto 0,\ (s_{in}, \beta) \mapsto 4,\ (s_1, \alpha) \mapsto 6,\ (s_2, \alpha) \mapsto 4,\ (s_3, \alpha) \mapsto 0,\ (s_3, \beta) \mapsto 2\big),$$

which was shown to satisfy $\mathbf{y}_2\mathbf{A} \leq \delta_{s_{in}}$ in Example 3.5. It even satisfies $\mathbf{y}_2\mathbf{A} = \delta_{s_{in}}$ and corresponds to the memoryless deterministic scheduler $\mathfrak{S}$ which chooses β in states s_{in} and s_3, and α otherwise. One can check that the expected number of visits of all states before reaching $\{\text{target}, \text{exit}\}$ in the induced Markov chain under $\mathfrak{S}$ is given by the vector

$$\big(s_{in} \mapsto 4,\ s_1 \mapsto 6,\ s_2 \mapsto 4,\ s_3 \mapsto 2\big). \qquad \triangle$$

Let $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be a solution of $\mathbf{yA} = \delta_{s_{in}}$. We say that $\mathbf{y}$ is *self-supporting* if there exists a subset $\mathcal{E}' \subseteq \text{supp}(\mathbf{y})$ such that the vector $\mathbf{y}'$ defined by

$$\mathbf{y}'(s,\alpha) = \begin{cases} \mathbf{y}(s,\alpha) & \text{if } (s,\alpha) \in \mathcal{E}' \\ 0 & \text{otherwise} \end{cases}$$

satisfies $\mathbf{y}'\mathbf{A} = \mathbf{0}$. The next proposition shows that each non-self-supporting solution of $\mathbf{yA} = \delta_{s_{in}}$ equals the expected number of visits of a corresponding memoryless deterministic scheduler.

Proposition 3.15. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP, $\mathcal{E}$ its enabled state-action pairs and $\mathbf{A}$ its system matrix. Furthermore, let $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be a non-self-supporting solution of $\mathbf{yA} = \delta_{s_{in}}$. Let $\mathfrak{S}$ be a memoryless scheduler satisfying $\mathfrak{S}(s,\alpha) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$.*

Then, $\mathfrak{S}$ satisfies $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$ and we have $\mathbf{y} = \mathbf{ev}^{\mathfrak{S}}$.

Proof. We first show (1) that the set of states reachable from s_{in} in $\mathcal{M}$ under $\mathfrak{S}$ (henceforth called R) is exactly the set $\{s \in S \mid \mathbf{y}(s) > 0\}$. Then we show (2) that $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$, which lets us apply Proposition 3.13 to show the claim.

(1.) First, we prove that $R \subseteq \{s \in S \mid \mathbf{y}(s) > 0\}$ holds. For all $n \geq 0$, we show that states $s \in S$ reachable within n steps from s_{in} under $\mathfrak{S}$ in $\mathcal{M}$ satisfy $\mathbf{y}(s) > 0$. For the initial state s_{in}, $\mathbf{y}(s_{in}) \geq 1$ follows from $\mathbf{yA} = \delta_{s_{in}}$. If state s is reachable in $n+1$ steps, then there exists a state s' reachable in n steps such that $\mathfrak{S}(s',\gamma) \cdot P(s',\gamma,s) > 0$ holds for some $\gamma \in \text{Act}(s')$. Using the assumption that $\mathbf{yA} = \delta_{s_{in}}$ holds, we get

$$\begin{aligned} \sum_{\alpha \in \text{Act}(s)} \mathbf{y}(s,\alpha) &= \delta_{s_{in}}(s) + \sum_{(t,\beta)\in\mathcal{E}} P(t,\beta,s) \cdot \mathbf{y}(t,\beta) \quad \geq \quad P(s',\gamma,s) \cdot \mathbf{y}(s',\gamma) \\ &= P(s',\gamma,s) \cdot \mathfrak{S}(s',\gamma) \cdot \mathbf{y}(s') \quad > \quad 0. \end{aligned}$$

Here we used that $\mathbf{y}(s') > 0$ holds by induction hypothesis.

To show $\{s \in S \mid \mathbf{y}(s) > 0\} \subseteq R$, assume that there exists some t with $\mathbf{y}(t) > 0$ and $t \notin R$. Let S' be the set of states that reach t in $\mathcal{M}$ under $\mathfrak{S}$. We have $s_{in} \notin S'$ by assumption, and for all $s \in S'$:

$$\sum_{\alpha \in \text{Act}(s)} \mathbf{y}(s,\alpha) \;=\; \sum_{s' \in S} \sum_{\alpha \in \text{Act}(s')} P(s',\alpha,s) \cdot \mathbf{y}(s',\alpha) \;=\; \sum_{s' \in S'} \sum_{\alpha \in \text{Act}(s')} P(s',\alpha,s) \cdot \mathfrak{S}(s',\alpha) \cdot \mathbf{y}(s').$$

Here we used that if $s \in S'$, then all states s' satisfying $P(s',\alpha,s) \cdot \mathfrak{S}(s',\alpha) > 0$ for some $\alpha \in \text{Act}(s')$ must be in S'. Define $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ by $\mathbf{y}'(s,\alpha) = \mathbf{y}(s,\alpha)$ if $s \in S'$, and $\mathbf{y}'(s,\alpha) = 0$

otherwise. It follows that $\mathbf{y}'\mathbf{A} = \mathbf{0}$ holds which contradicts our assumption that $\mathbf{y}$ is non-self-supporting. This concludes the proof of $R = \{s \in S \mid \mathbf{y}(s) > 0\}$.

(2.) Now we show that all states in R have a path to the set $\{\text{target}, \text{exit}\}$ in $\mathcal{M}$ under $\mathfrak{S}$. Assume that this was not the case and let $S' \subseteq R$ be the set of states in R which do not reach $\{\text{target}, \text{exit}\}$. It follows that for all $(s, \alpha) \in \mathcal{E}$ such that $s \in S'$ we have $\mathbf{y}(s,\alpha) \cdot \sum_{s' \in S'} P(s, \alpha, s') = \mathbf{y}(s, \alpha)$ and therefore

$$\sum_{s \in S'} \sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s, \alpha) = \sum_{s, s' \in S'} \sum_{\alpha \in \mathrm{Act}(s)} P(s, \alpha, s') \cdot \mathbf{y}(s, \alpha). \tag{$\dagger$}$$

By summing over the rows of $\mathbf{yA} = \delta_{s_{in}}$ which correspond to states in S' we get:

$$\begin{aligned}\sum_{\substack{s \in S' \\ \alpha \in \mathrm{Act}(s)}} \mathbf{y}(s, \alpha) &= \sum_{s \in S'} \Big(\delta_{s_{in}}(s) + \sum_{\substack{s' \in S' \\ \alpha' \in \mathrm{Act}(s')}} P(s', \alpha', s) \cdot \mathbf{y}(s', \alpha') + \sum_{\substack{s' \notin S' \\ \alpha' \in \mathrm{Act}(s')}} P(s', \alpha', s) \cdot \mathbf{y}(s', \alpha') \Big) \\ &= \sum_{s \in S'} \delta_{s_{in}}(s) + \sum_{s \in S'} \sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s, \alpha) + \sum_{s \in S'} \sum_{\substack{s' \notin S' \\ \alpha' \in \mathrm{Act}(s')}} P(s', \alpha', s) \cdot \mathbf{y}(s', \alpha')\end{aligned}$$

The last equation uses the equality ($\dagger$) from right to left as a substitution rule for the second summand. It follows that

$$\sum_{s \in S'} \delta_{s_{in}}(s) + \sum_{s \in S'} \sum_{\substack{s' \notin S' \\ \alpha' \in \mathrm{Act}(s')}} P(s', \alpha', s) \cdot \mathbf{y}(s', \alpha') = 0,$$

which means that $s_{in} \notin S'$ and for all $s \in S', s' \notin S'$ and $\alpha \in \mathrm{Act}(s')$ we have

$$P(s', \alpha, s) \cdot \mathfrak{S}(s', \alpha) \cdot \mathbf{y}(s') = 0.$$

But this contradicts the fact that all states in S' are reachable from s_{in}.

We have shown that the set of states reachable in $\mathcal{M}$ under $\mathfrak{S}$ from s_{in} is $R = \{s \mid \mathbf{y}(s) > 0\}$, and all states in R have a path to $\{\text{target}, \text{exit}\}$ under $\mathfrak{S}$ in $\mathcal{M}$. The latter statement implies $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond\{\text{target}, \text{exit}\}) = 1$. Applying Proposition 3.13 yields the claim. □

To sum up, the above propositions show that

- if $\mathfrak{S}$ is a memoryless scheduler for $\mathcal{M}$ satisfying $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$, then the expected number of visits $\mathbf{ev}^{\mathfrak{S}}$ under $\mathfrak{S}$ satisfy $\mathbf{ev}^{\mathfrak{S}}\mathbf{A} = \delta_{s_{in}}$ (Proposition 3.13), and
- if $\mathbf{y}$ is a non-self-supporting solution of $\mathbf{yA} = \delta_{s_{in}}$, then there exists a corresponding (and easily computable) memoryless scheduler $\mathfrak{S}$ satisfying $\mathbf{y} = \mathbf{ev}^{\mathfrak{S}}$ (Proposition 3.15).

Both propositions hold also for MDPs which are not EC-free.

Example 3.16. Consider the MDP $\mathcal{M}_3$ in Figure 3.4 and let $\mathbf{A}$ be its system matrix. Let $\mathbf{y}_1$ be defined by

$$\big((s_{in}, \alpha) \mapsto 1,\ (s_1, \beta) \mapsto 1,\ (s_2, \alpha) \mapsto 4/3,\ (s_3, \alpha) \mapsto 2/3 \big),$$

and $\mathbf{y}_1(q, \gamma) = 0$ for all remaining enabled state-action pairs (q, γ). It satisfies $\mathbf{y}_1\mathbf{A} = \delta_{s_{in}}$, but does not correspond to the expected number of visits under any memoryless scheduler (by the correspondence used in Proposition 3.15). This is because if action β is chosen with probability

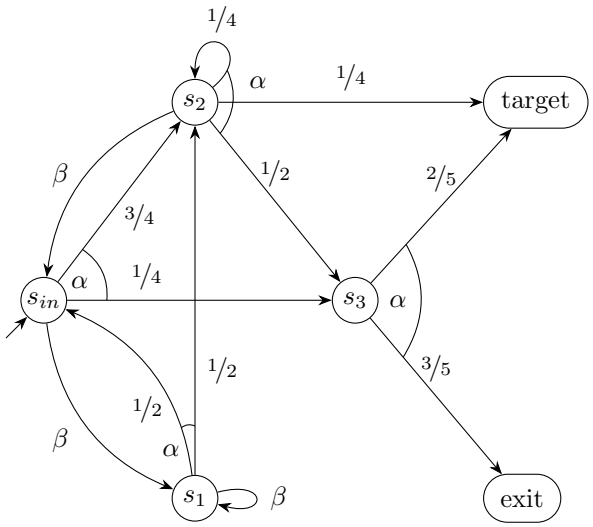


Figure 3.4: An example MDP $\mathcal{M}_3$ which differs from $\mathcal{M}_1$ (see Figure 3.1) by including additional actions β in states s_1 and s_2 and by excluding the action β in s_3. In contrast to $\mathcal{M}_1$, $\mathcal{M}_3$ is not EC-free as the sets $\{s_1\}$ and $\{s_{in}, s_1, s_2\}$ induce proper end components.

zero in s_{in}, then the expected number of visits to state s_1 must be zero. Nevertheless, $\mathbf{y}_1(s_1) > 0$ holds. The solution $\mathbf{y}_1$ is self-supporting because the vector $\mathbf{y}'$ which is defined as $\mathbf{y}'(s_1, \beta) = 1$ and zero otherwise satisfies $\mathbf{y}'\mathbf{A} = \mathbf{0}$. Intuitively, the value $\mathbf{y}_1(s_1)$ supports itself using the self loop under action β. △

The previous propositions considered linear equation systems of the form $\mathbf{y}\mathbf{A} = \delta_{s_{in}}$. Using the intuition of flow equations, using $\geq$ (respectively $\leq$) instead means that states may have a higher (respectively lower) "out-flow" than is supported by their incoming edges (see Equation (3.1) for comparison). We start with the observation that if $\mathbf{y}$ is a solution to one of these systems of *inequalities*, then there exists a vector $\mathbf{y}'$ satisfying the corresponding *equation* system. Additionally, $\mathbf{y}'$ is either a point-wise lower bound (in the case $\leq$) or a point-wise upper bound (in the case $\geq$) of $\mathbf{y}$.

Lemma 3.17. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an EC-free MDP with system matrix $\mathbf{A}$. For all $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ we have:*

- *if $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}}$, then there exists $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ such that $\mathbf{y}'\mathbf{A} = \delta_{s_{in}}$, $\mathbf{y}' \leq \mathbf{y}$ and $\mathbf{y}'(s,\alpha)/\mathbf{y}'(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$,*
- *if $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}}$, then there exists $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ such that $\mathbf{y}'\mathbf{A} = \delta_{s_{in}}$, $\mathbf{y}' \geq \mathbf{y}$, $\text{supp}(\mathbf{y}') = \text{supp}(\mathbf{y})$ and $\mathbf{y}'(s,\alpha)/\mathbf{y}'(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$.*

Proof. We only prove the second claim, the first one is proven similarly. Consider the sequence of vectors in $\mathbb{R}^{\mathcal{E}}_{\geq 0}$ defined by $\mathbf{y}_1 = \mathbf{y}$ and

$$\mathbf{y}_{i+1}(s,\alpha) = \frac{\mathbf{y}_i(s,\alpha)}{\mathbf{y}_i(s)} \cdot \big(\delta_{s_{in}}(s) + \sum_{(s',\alpha)\in\mathcal{E}} P(s',\alpha,s) \cdot \mathbf{y}_i(s',\alpha)\big), \qquad (*)$$

for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}_i(s) > 0$, and $\mathbf{y}_{i+1}(s,\alpha) = 0$ otherwise.

Our aim is to show by induction that $\mathbf{y}_{i+1} \geq \mathbf{y}_i$, $\operatorname{supp}(\mathbf{y}_{i+1}) = \operatorname{supp}(\mathbf{y}_i)$, $\mathbf{y}_i\mathbf{A} \leq \delta_{s_{in}}$ and $\mathbf{y}_i(s,\alpha)/\mathbf{y}_i(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $i \geq 1$ and $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$. The base case follows by assumption. We first show $\mathbf{y}_{i+1} \geq \mathbf{y}_i$ and $\operatorname{supp}(\mathbf{y}_{i+1}) = \operatorname{supp}(\mathbf{y}_i)$. For $s \in S$ such that $\mathbf{y}_i(s) = 0$ we have $\mathbf{y}_{i+1}(s,\alpha) = 0 \geq \mathbf{y}_i(s,\alpha)$ for all $\alpha \in \operatorname{Act}(s)$ by definition. This also shows $\operatorname{supp}(\mathbf{y}_{i+1}) \subseteq \operatorname{supp}(\mathbf{y}_i)$. For all other $s \in S$ we can use the hypothesis $\mathbf{y}_i\mathbf{A} \leq \delta_{s_{in}}$, which implies

$$\mathbf{y}_i(s) \;\leq\; \delta_{s_{in}}(s) + \sum_{(s',\alpha)\in\mathcal{E}} P(s',\alpha,s)\cdot \mathbf{y}_i(s',\alpha),$$

for all $s \in S$ and hence, by $(*)$, $\mathbf{y}_{i+1}(s,\alpha) \geq \mathbf{y}_i(s,\alpha)$. This also shows $\operatorname{supp}(\mathbf{y}_{i+1}) \supseteq \operatorname{supp}(\mathbf{y}_i)$, and therefore $\operatorname{supp}(\mathbf{y}_{i+1}) = \operatorname{supp}(\mathbf{y}_i)$.

To show that $\mathbf{y}_{i+1}\mathbf{A} \leq \delta_{s_{in}}$ holds we calculate:

$$\begin{aligned}
\sum_{\beta\in\operatorname{Act}(s)} \mathbf{y}_{i+1}(s,\beta) \;&=\; \sum_{\beta\in\operatorname{Act}(s)} \frac{\mathbf{y}_i(s,\beta)}{\mathbf{y}_i(s)} \cdot \Big(\delta_{s_{in}}(s) + \sum_{(s',\alpha)\in\mathcal{E}} P(s',\alpha,s)\cdot \mathbf{y}_i(s',\alpha)\Big) \\
&=\; \delta_{s_{in}}(s) + \sum_{(s',\alpha)\in\mathcal{E}} P(s',\alpha,s)\cdot \mathbf{y}_i(s',\alpha) \\
&\leq\; \delta_{s_{in}}(s) + \sum_{(s',\alpha)\in\mathcal{E}} P(s',\alpha,s)\cdot \mathbf{y}_{i+1}(s',\alpha),
\end{aligned}$$

where the last inequality follows from $\mathbf{y}_{i+1}(s',\alpha) \geq \mathbf{y}_i(s',\alpha)$, which holds for all $(s',\alpha) \in \mathcal{E}$. Finally, we show that $\mathbf{y}_{i+1}(s,\alpha)/\mathbf{y}_{i+1}(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ holds for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$. Expanding the definition yields

$$\frac{\mathbf{y}_{i+1}(s,\alpha)}{\mathbf{y}_{i+1}(s)} = \frac{\frac{\mathbf{y}_i(s,\alpha)}{\mathbf{y}_i(s)} \cdot \big(\delta_{s_{in}}(s) + \ldots\big)}{\sum_{\beta\in\operatorname{Act}(s)} \frac{\mathbf{y}_i(s,\beta)}{\mathbf{y}_i(s)} \cdot \big(\delta_{s_{in}}(s) + \ldots\big)} = \frac{\mathbf{y}_i(s,\alpha)}{\mathbf{y}_i(s)},$$

where $(\delta_{s_{in}}(s) + \ldots)$ represents the sum in brackets from $(*)$.

As $\mathcal{M}$ is EC-free, the set $\{\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}'\mathbf{A} \leq \delta_{s_{in}}\}$ is bounded by Proposition 3.9. Hence, the limit of the constructed sequence $\mathbf{y}_\infty = \lim_{i\to\infty} \mathbf{y}_i$ exists and satisfies $\mathbf{y}_\infty\mathbf{A} = \delta_{s_{in}}$, $\mathbf{y}_\infty \geq \mathbf{y}$ and $\mathbf{y}_\infty(s,\alpha)/\mathbf{y}_\infty(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $(s,\alpha) \in \mathcal{E}$ such that $\mathbf{y}(s) > 0$. □

The assumption of EC-freeness was used in the above proof only to guarantee that the set $\{\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}'\mathbf{A} \leq \delta_{s_{in}}\}$ is bounded. For the first statement this is not needed, as the constructed sequence $\mathbf{y}_1, \mathbf{y}_2, \ldots$ is point-wise non-increasing and all vectors remain nonnegative. Hence, boundedness of the sequence holds without the assumption of EC-freeness.

Together with Proposition 3.15, the above lemma shows that for EC-free MDPs, solutions of the systems of *inequalities* $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}}$ and $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}}$, provide point-wise lower, respectively upper, bounds on the expected number of visits of some memoryless scheduler.

Proposition 3.18. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \operatorname{Act}, s_{in}, P)$ be an EC-free MDP with system matrix $\mathbf{A}$ and enabled state-action pairs $\mathcal{E}$. Take $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ and let $\mathfrak{S}$ be a memoryless scheduler satisfying $\mathfrak{S}(s,\alpha) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ whenever $\mathbf{y}(s) > 0$. Then:*

- *if $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}}$, then $\mathbf{y} \geq \mathbf{ev}^{\mathfrak{S}}$, and*
- *if $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}}$, then $\mathbf{y} \leq \mathbf{ev}^{\mathfrak{S}}$.*

Proof. If $\mathbf{y}\mathbf{A} \geq \delta_{s_{in}}$ holds, then by Lemma 3.17 we find $\mathbf{y}'$ such that $\mathbf{y}'\mathbf{A} = \delta_{s_{in}}$, $\mathbf{y}' \leq \mathbf{y}$ and $\mathbf{y}'(s,\alpha)/\mathbf{y}'(s) = \mathbf{y}(s,\alpha)/\mathbf{y}(s)$ for all $s \in S$ such that $\mathbf{y}(s) > 0$. The vector $\mathbf{y}'$ is non-self-

supporting, as otherwise we could construct a vector $\mathbf{y}'' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{y}'' \neq \mathbf{0}$ and $\mathbf{y}''\mathbf{A} = \mathbf{0}$. But this would contradict EC-freeness of $\mathcal{M}$ by Lemma 3.8.

Now we can apply Proposition 3.15 to conclude that $\mathbf{y}' = \mathbf{ev}^{\mathfrak{S}}$ holds. The other case can be shown analogously. □

Recall that the probability of reaching the state "target" under a scheduler $\mathfrak{S}$ satisfying $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond\{\text{target}, \text{exit}\}) = 1$ can be expressed in terms of its expected number of visits $\mathbf{ev}^{\mathfrak{S}}$ (see Section 2.2.4). Concretely, we have $\Pr^{\mathfrak{S}}_{s_{in}}(\Diamond\,\text{target}) = \sum_{(s,\alpha)\in\mathcal{E}} \mathbf{ev}^{\mathfrak{S}}(s,\alpha)\cdot\mathbf{t}(s,\alpha) = \mathbf{ev}^{\mathfrak{S}}\cdot\mathbf{t}$, where $\mathbf{t}$ is the target vector of $\mathcal{M}$. Now if we are given a solution of $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \gtrsim \lambda$, then by Proposition 3.18 we find a scheduler $\mathfrak{S}$ such that $\mathbf{y} \leq \mathbf{ev}^{\mathfrak{S}}$. It follows that $\mathbf{ev}^{\mathfrak{S}}\mathbf{t} \gtrsim \lambda$, and hence indeed $\mathfrak{S}$ is a scheduler which witnesses that $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ holds. The analogous observation holds for solutions of $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$.

This yields another proof that solutions to these systems of inequalities are valid certificates of the corresponding probabilistic reachability constraint (see Proposition 3.4) for EC-free MDP. The proof of Proposition 3.4 using Farkas' Lemma is certainly more direct, and extends directly to MDPs which are not EC-free for maximal reachability probabilities, which is discussed in the following section.

3.1.3 MDPs with proper end components

In Section 3.1.1 certificates for all kinds of probabilistic reachability constraints were introduced for EC-free MDPs. This assumption implies that a vector $\mathbf{z}$ satisfying $\mathbf{Az} \leq \mathbf{t}$ is indeed a pointwise lower bound on $\mathbf{pr}^{\min}$ (Lemma 2.9) which was used in Proposition 3.1 to characterize certificates for $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$. Using Farkas' Lemma, this characterization then led to a definition of certificates for $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \lesssim \lambda$ in Proposition 3.4

On the other hand, the proofs of statements (2.) of Propositions 3.1 and 3.4, which concern the maximal probability of reaching target, do not depend on this assumption. This is because the part of Lemma 2.9 concerning $\mathbf{pr}^{\max}$ does not rely on EC-freeness. Hence, these statements hold also for MDPs which are not EC-free. This is not true for the statements of Propositions 3.1 and 3.4 concerning minimal reachability probabilities, which fail for MDPs that are not EC-free.

Example 3.19. Consider the MDP $\mathcal{M}_3$ in Figure 3.4 and let $\mathbf{A}$ be its system matrix and $\mathbf{t}$ its target vector. The vector

$$\mathbf{z}_1 = \left(s_{in} \mapsto 2/5,\quad s_1 \mapsto 2/5,\quad s_2 \mapsto 2/5,\quad s_3 \mapsto 2/5\right)$$

defined in Example 3.2 satisfies $\mathbf{Az}_1 \leq \mathbf{t}$ and $\mathbf{z}(s_{in}) \geq 2/5$. However, $\mathbf{Pr}^{\min}_{\mathcal{M}_3}(\Diamond\,\text{target}) = 0$ holds, as s_{in} is included in a proper end component (which does not contain target), and hence there exists a scheduler which avoids target forever from s_{in}. This example shows that solutions of $\mathbf{Az} \leq \mathbf{t}$ cannot be used to certify $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda$ in the same way as for EC-free MDPs.

On the other hand, consider the system of inequalities $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \leq \lambda$, which was used to certify $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) \leq \lambda$ for EC-free MDPs in the previous section. One can check that the minimum value of $\mathbf{yt}$ when ranging over all solutions $\mathbf{y}$ of $\mathbf{yA} \geq \delta_{s_{in}}$ is $11/20$. In particular, there is no solution $\mathbf{y}$ of this system of inequalities satisfying $\mathbf{yt} = 0$. Hence, there is no certificate for the statement $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) \leq 0$ of the form introduced for EC-free MDPs. △

The above example shows that for MDPs which are not EC-free, solutions of the system of inequalities $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \gtrsim \lambda$ may be "spurious" in the sense that the system is satisfiable

but the property $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \gtrsim \lambda$ does not hold. Dually, the system of inequalities $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$ may be unsatisfiable although $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \lesssim \lambda$ is satisfied. The other directions hold, however. That is, if $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \gtrsim \lambda$ holds, then $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ has a solution, and if $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$ is satisfiable, then $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \lesssim \lambda$ holds.

Proposition 3.20. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, $\mathbf{A}$ be its system matrix and $\mathbf{t}$ be its target vector. Then for all $\gtrsim \in \{\geq, >\}$, $\lesssim \in \{\leq, <\}$ and $\lambda \in [0, 1]$:*

- *If $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \gtrsim \lambda$ holds, then there exists $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ satisfying $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \gtrsim \lambda$.*
- *If $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfies $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$, then $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \lesssim \lambda$ holds.*

Proof. By the linear-programming based characterization of $\mathbf{pr}^{\min}$ it follows that $\mathbf{pr}^{\min}$ satisfies $\mathbf{A} \cdot \mathbf{pr}^{\min} \leq \mathbf{t}$. If $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \gtrsim \lambda$ holds, then additionally we have $\mathbf{pr}^{\min}(s_{in}) \gtrsim \lambda$.

For the second claim, let $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be a solution of $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yt} \lesssim \lambda$. By Lemma 3.17 we find a vector $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ such that $\mathbf{y}'\mathbf{A} = \delta_{s_{in}}$ and $\mathbf{y}' \leq \mathbf{y}$. Here we use that EC-freeness is not actually required for the first statement of Lemma 3.17, as the set $\{\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}' \leq \mathbf{y}\}$ is bounded for all $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$.

If $\mathbf{y}'$ is non-self-supporting, then there exists a scheduler $\mathfrak{S}$ for $\mathcal{M}$ such that $\mathbf{ev}^{\mathfrak{S}} = \mathbf{y}'$ by Proposition 3.15. It follows that $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\lozenge \text{target}) = \mathbf{ev}^{\mathfrak{S}} \cdot \mathbf{t} \leq \mathbf{yt} \lesssim \lambda$.

If $\mathbf{y}'$ is self-supporting, then, by definition, we find $\mathbf{y}_1, \mathbf{y}_2$ such that $\mathbf{y}_1\mathbf{A} = \mathbf{0}$, $\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{y}'$, $\text{supp}(\mathbf{y}_1) \cap \text{supp}(\mathbf{y}_2) = \varnothing$ and $\mathbf{y}_2$ is non-self-supporting. By Lemma 3.8, $\text{supp}(\mathbf{y}_1)$ induces proper end components, and hence if $(s, \alpha) \in \text{supp}(\mathbf{y}_1)$, then $\mathbf{t}(s, \alpha) = 0$. But this implies $\mathbf{yt} = \mathbf{y}_2\mathbf{t}$. The vector $\mathbf{y}_2$ is non-self-supporting and we have $\mathbf{y}_2\mathbf{A} = (\mathbf{y}_1 + \mathbf{y}_2)\mathbf{A} = \mathbf{y}'\mathbf{A} = \delta_{s_{in}}$. Then, by the same reasoning as above, we may conclude that $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \lesssim \lambda$. □

Our next aim is to define certificate conditions which are sound and complete for minimal reachability probabilities in MDPs with proper end components, thereby dealing with the issues presented in Example 3.19. We propose to first determine the set of states which can only reach the state "target" in $\mathcal{M}$ by passing through some proper end component, and exclude them from further considerations. This is supported by the fact that the minimal probability to reach target from such a state is zero. We call states that do not satisfy this property *min-relevant*.

Definition 3.21 (min-relevant states). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form. We denote by S_R the states of $\mathcal{M}$ from which there exists a path to target in the underlying graph of $\mathcal{M}$ that does not pass through a proper end component. Formally:

$$S_R = \{s \in S \mid \text{there exists } s_0 s_1 \dots s_n \, \text{target} \in \text{Paths}_{\text{fin}}(\mathcal{M}, s) \text{ such that } s_i \text{ is not included in a proper end component for all } i \in \{0, \dots, n\}.\}$$

States in S_R are called *min-relevant*. Furthermore, let $\mathcal{E}^* = \{(s, \alpha) \mid s \in S_R\}$, $\mathbf{A}^* = \mathbf{A}|_{\mathcal{E}^* \times S_R}$ and $\mathbf{t}^* = \mathbf{t}|_{S_R}$ be the restrictions of the corresponding matrices/vectors defined in Definition 2.7 to states in S_R.

Observe that each state $s \in S$ has some path to the state "target" if $\mathcal{M}$ is in reachability form, by the assumption that $\mathbf{Pr}^{\max}_{s}(\lozenge \text{target}) > 0$ holds for all $s \in S$ (see Definition 2.5).

Remark 3.22. If the initial state s_{in} is not included in the min-relevant states S_R, then this proves that there exists a scheduler which avoids target with probability one from s_{in}. Hence, in this case we have $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) = 0$. This holds, for example, for the MDP $\mathcal{M}_3$ defined

in Figure 3.4, where the only min-relevant state is s_3 (i.e., we have $S_R = \{s_3\}$). When considering Farkas certificates for minimal reachability probabilities, we will assume that $s_{in} \in S_R$ holds. △

Another way to define $\mathbf{A}^*$ and $\mathbf{t}^*$ is to consider the MDP $\mathcal{M}^*$ one gets by identifying all states in $S \setminus S_R$ with "exit", and then taking the matrices as defined by Definition 2.7 for $\mathcal{M}^*$. The MDP $\mathcal{M}^*$ will be defined precisely in the proof of Theorem 3.24, and will appear again in Chapter 4 in the context of witnessing subsystems (see Definition 4.9). As $S \setminus S_R$ includes all proper end components (except for those induced by $\{\text{target}, \text{exit}\}$), $\mathcal{M}^*$ is EC-free and, furthermore, preserves minimal reachability probabilities for all states $s \in S_R$. We will use this fact later in the proof that Farkas certificates, as defined next for all cases of probabilistic reachability constraints, indeed certify that the corresponding properties are satisfied.

Definition 3.23 (Farkas certificates). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, let $\mathbf{A}, \mathbf{t}$ be the system matrix and target vector of $\mathcal{M}$ (Definition 2.7) and $\mathcal{E}$ be the set of enabled state-action pairs. Furthermore, let S_R be the min-relevant states of $\mathcal{M}$, and $\mathcal{E}^*, \mathbf{A}^*, \mathbf{t}^*$ be the corresponding restrictions to states in S_R (Definition 3.21).

The sets of *Farkas certificates* of $\mathcal{M}$ are defined as follows:

- $\mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda) = \{\mathbf{z} \in \mathbb{R}^{S_R}_{\geq 0} \mid \mathbf{A}^*\mathbf{z} \leq \mathbf{t}^* \wedge \mathbf{z}(s_{in}) \gtrsim \lambda\}$
- $\mathcal{F}^{\max}_{\mathcal{M},\lesssim}(\lambda) = \{\mathbf{z} \in \mathbb{R}^{S}_{\geq 0} \mid \mathbf{A}\mathbf{z} \geq \mathbf{t} \wedge \mathbf{z}(s_{in}) \lesssim \lambda\}$
- $\mathcal{F}^{\min}_{\mathcal{M},\lesssim}(\lambda) = \{\mathbf{y} \in \mathbb{R}^{\mathcal{E}^*}_{\geq 0} \mid \mathbf{y}\mathbf{A}^* \geq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t}^* \lesssim \lambda\}$
- $\mathcal{F}^{\max}_{\mathcal{M},\gtrsim}(\lambda) = \{\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \gtrsim \lambda\}$

If $\mathcal{M}$ is clear from the context we will omit it from the subscripts.

If $\mathcal{M}$ is EC-free, then $S_R = S$ and therefore $\mathbf{A}^* = \mathbf{A}$ and $\mathbf{t}^* = \mathbf{t}$ hold. Hence, in this case the Farkas certificates as defined above correspond exactly to the nonnegative solutions of the systems of inequalities appearing in Propositions 3.1 and 3.4. Observe that the proof of Proposition 3.1 shows that one always finds a nonnegative certificate for the corresponding property. For the other direction of Proposition 3.1 the stronger statement holds: Any solution of the inequalities (even those with negative entries) can be used as a certificate for the property. But we will, in the following, restrict ourselves to nonnegative Farkas certificates.

As a consequence of Propositions 3.1 and 3.4, the following theorem, which states that Farkas certificates indeed certify the corresponding conditions, follows directly for EC-free MDPs. For MDPs with proper end components we use the following: First, the statements of Propositions 3.1 and 3.4 which concern $\mathbf{Pr}^{\max}$ actually do not require EC-freeness. And, second, the MDP $\mathcal{M}^*$, which identifies all states in $S \setminus S_R$ with the state "exit", is EC-free and preserves the minimal reachability probabilities in all states that are not in S_R. Furthermore, the sets of Farkas certificates for $\mathcal{M}$ and $\mathcal{M}^*$ concerning minimal reachability probabilities coincide.

Theorem 3.24. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form (but not necessarily EC-free). Then, for all $\mathfrak{m} \in \{\min, \max\}$, $\bowtie \in \{\leq, <, \geq, >\}$ and $\lambda \in [0, 1]$ we have*

$$\mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\bowtie}(\lambda) \text{ is not empty if and only if } \mathbf{Pr}^{\mathfrak{m}}_{s_{in}}(\Diamond\, \text{target}) \bowtie \lambda \text{ holds.}$$

Proof. If $\mathcal{M}$ is EC-free, then $S_R = S$ and all statements follow from Propositions 3.1 and 3.4. Here we use that the proof of the direction from right to left of Proposition 3.1 shows that indeed nonnegative certificates always exist.

If $\mathcal{M}$ is not EC-free, then we first observe that the statements (2.) of Propositions 3.1 and 3.4, which concern $\mathbf{Pr}^{\max}$, do not depend on EC-freeness of the MDP. The only point where this assumption is used in the proofs is to apply Lemma 2.9. But Lemma 2.9 requires EC-freeness only for the statement corresponding to $\mathbf{Pr}^{\min}$.

It remains to show that the cases with $\mathfrak{m} = \min$ hold if $\mathcal{M}$ is not EC-free. Consider the MDP $\mathcal{M}^* = (S_R \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P^*)$ where for all $s \in S_R$ and $\alpha \in \text{Act}(s)$ we have $P^*(s, \alpha, t) = P(s, \alpha, t)$ if $t \in S_R$, $P^*(s, \alpha, \text{target}) = P(s, \alpha, \text{target})$ and $P^*(s, \alpha, \text{exit}) = P(s, \alpha, \text{exit}) + \sum_{t \in (S \setminus S_R)} P(s, \alpha, t)$. It follows by the definition of S_R that $\mathcal{M}^*$ is EC-free and in reachability form. Hence, by reduction to the case of EC-free MDPs which we have already proved, we get: $\mathcal{F}^{\min}_{\mathcal{M}^*, \bowtie}(\lambda)$ is not empty iff $\mathbf{Pr}^{\min}_{\mathcal{M}^*, s_{in}}(\Diamond \text{target}) \bowtie \lambda$ holds. Furthermore, $\mathcal{F}^{\min}_{\mathcal{M}, \bowtie}(\lambda) = \mathcal{F}^{\min}_{\mathcal{M}^*, \bowtie}(\lambda)$ and $\mathbf{Pr}^{\min}_{\mathcal{M}^*, s_{in}}(\Diamond \text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}, s_{in}}(\Diamond \text{target})$ hold by construction, which concludes the proof. □

In order to apply the above theorem for minimal probabilities in MDP that are not EC-free, one has to first compute the set S_R of min-relevant states. This can be done using standard methods to compute maximal end components [deA97, BK08]. Farkas certificates produced using these methods can only be trusted if the computation of the maximal end components is sound, which by itself is not a trivial procedure. Ideally, a Farkas certificate for min-properties should always be paired with a certificate that the set S_R has been computed correctly. Certifying the correctness of a computation returning the set of maximal end components is the topic of the next section.

Remark 3.25. The only properties of S_R that the above proof depends on is that $S \setminus S_R$ includes all proper end components and is included in the set of states s satisfying $\mathbf{Pr}^{\min}_s(\Diamond \text{target}) = 0$. Our definition of S_R additionally satisfies that identifying all states in $S \setminus S_R$ with "exit" yields an MDP in reachability form (i.e., all states have a path to "target" in the resulting MDP) and for EC-free MDPs we have $S_R = S$. Another choice which also yields correct certificates is to set $S_R = \{s \in S \mid \mathbf{Pr}^{\min}_s(\Diamond \text{target}) > 0\}$. △

3.1.4 Certifying the decomposition into maximal end components

The aim of this section is to define a certificate that can be returned along with a set of sub-MDPs which proves, in a simple-to-check way, that the returned sub-MDPs are indeed the maximal end components of the given MDP. Several algorithms (some of them quite elaborate) for computing the set of maximal end components exist [deA97, HC11, CŁ13].

We will build on the following characterization, which says that a set of sub-MDPs $\mathcal{D}$ equals the set of maximal end components of $\mathcal{M}$ if all sub-MDPs are end components and the induced quotient is EC-free. Here we use the target-directed quotient $\mathcal{M}^{\text{target}}_{\mathcal{D}}$ as defined in Section 2.2.2 (at this point it doesn't matter, we could also have used the exit-directed quotient).

Proposition 3.26. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and let $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be a set of sub-MDPs of $\mathcal{M}$. Then, $\mathcal{D}$ is the set of maximal end components of $\mathcal{M}$ if and only if all of the following hold:*

(a) $E_1, \ldots, E_k$ is a partition of S,

(b) (E, A) is an end component, for all $(E, A) \in \mathcal{D}$, and

(c) $\mathcal{M}^{\text{target}}_{/\mathcal{D}}$ is EC-free.

Proof. The direction from left to right is clear. So suppose, by contraposition, that $\mathcal{D}$ is not the set of maximal end components of $\mathcal{M}$. We will additionally assume that (a) and (b) hold, and show that in this case (c) must be violated, i.e., $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$ is not EC-free.

First, let us assume that some state set E_i is strictly included in a bigger end component. Then there must exist a subset $\{E_{c_1}, \dots, E_{c_l}\} \subseteq \{E_1, \dots, E_k\}$ such that $l > 1$ and all states in $\bigcup_{1 \leq i \leq l} E_{c_i}$ belong to the same maximal end component. But then the set of states $\{(E_{c_1}, A_{c_1}), \dots, (E_{c_l}, A_{c_l})\}$ of $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$ induces a proper end component in $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$, and hence $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$ is not EC-free.

If $\{E_1, \dots, E_k\}$ is the state partition which induces the maximal end components, then the only possibility that $\mathcal{D}$ is not the MEC decomposition of $\mathcal{M}$ is that for some $1 \leq i \leq k$ and $s \in E_i$ the set $A_i(s)$ is missing an action which is actually internal in the corresponding end component, by the definition of $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$. Such and action α would, however, induce a self loop with probability one in $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$. More precisely, we would have $P_{\mathcal{M}_{/\mathcal{D}}}\big((E_i, A_i), \alpha, (E_i, A_i)\big) = 1$. But then $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$ is not EC-free. □

Let us assume that the sequence $E_1, \dots, E_k$ is provided as a function $S \to \mathbb{N}$ which assigns to each state the index of its end component (concretely this can be given as a vector in $\mathbb{N}^S$). Then, $E_1, \dots, E_k$ forms a partition of S (condition (a)) if each state is assigned exactly one index. This is easy to check. In the following, we will focus on how to certify conditions (b) and (c) of Proposition 3.26. By definition, the sub-MDP (E, A) is an end component exactly if for all $q \in E$ and $\alpha \in A(q)$, the possible successors of state-action pair (q, α) are included in E, and the induced graph of the sub-MDP (E, A) is strongly connected. The first condition can be easily checked, and it remains to show how to certify strong connectedness.

Certifying strong connectedness. Here we use the fact that a directed graph is strongly connected if and only if there exists a state which reaches all other states and which is reachable from all other states.

Lemma 3.27. *Let $G = (V_G, E_G)$ be a directed graph. G is strongly connected if and only if there exist two functions* $\mathtt{bwd}, \mathtt{fwd} : V_G \to \mathbb{N}$ *such that:*

- *there exists a unique vertex $v_s \in V_G$ such that* $\mathtt{bwd}(v_s) = \mathtt{fwd}(v_s) = 0$,
- *for each $v \in V_G \setminus \{v_s\}$ there exists $v' \in V_G$ such that $(v, v') \in E$ and* $\mathtt{fwd}(v') < \mathtt{fwd}(v)$, *and*
- *for each $v \in V_G \setminus \{v_s\}$ there exists $v' \in V_G$ such that $(v', v) \in E$ and* $\mathtt{bwd}(v') < \mathtt{bwd}(v)$.

Proof. Let us first assume that G is strongly connected. Pick an arbitrary vertex $v_s \in V_G$ and define $\mathtt{fwd}, \mathtt{bwd}$ by setting $\mathtt{fwd}(v_s) = \mathtt{bwd}(v_s) = 0$ and for all other vertices $v \in V_G \setminus \{v_s\}$ set $\mathtt{fwd}(v)$ to be the length of a shortest path from v to v_s in G, and $\mathtt{bwd}(v)$ to be the length of a shortest path from v to v_s in the graph G' one gets by reversing all edges in G. As both G and G' are strongly connected, such paths always exist, and both $\mathtt{fwd}$ and $\mathtt{bwd}$ satisfy the required properties by construction. If functions $\mathtt{fwd}$ and $\mathtt{bwd}$ exist satisfying the above properties, then this guarantees that each vertex has a path to v_s and is reachable from v_s, where v_s is the unique vertex satisfying $\mathtt{fwd}(v_s) = \mathtt{bwd}(v_s) = 0$. It follows that G is strongly connected. □

Certifying EC-freeness. To certify that the quotient $\mathcal{M}_{/\mathcal{D}}^{\text{target}}$ is EC-free, we will use the following lemma.

Lemma 3.28. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and $\mathbf{A}, \mathbf{t}$ be the system matrix and target vector of $\mathcal{M}$. Furthermore, let $\mathbf{b} \in \mathbb{R}^{\mathcal{E}}_{>0}$ be any vector satisfying $\mathbf{b}(s,\alpha) > 0$ for all enabled state-action pairs $(s,\alpha) \in \mathcal{E}$.*

Then, $\mathcal{M}$ is EC-free if and only if there exists a vector $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ such that $\mathbf{Az} \geq \mathbf{b}$ holds.

Proof. For the direction from left to right, suppose that $\mathcal{M}$ is EC-free. Consider the MDP $\mathcal{M}'$ one gets by collapsing states "exit" and "target" and adding the reward function $\text{rew} : \mathcal{E} \to \mathbb{N}$ defined as $\text{rew}(s,\alpha) = \mathbf{b}(s,\alpha)$ for all $(s,\alpha) \in \mathcal{E}$. Then $\mathcal{M}'$ is in reward reachability form and hence the maximal expected total reward is finite for each state. It is characterized by the linear program: *minimize* $\sum_{s \in S} x_s$ *such that* $\mathbf{Ax} \geq \mathbf{b}$ (see Section 2.2.3). Hence, in particular, there exists a nonnegative vector satisfying these linear inequalities.

For the other direction, suppose that $\mathcal{M}$ is not EC-free and there exists a vector $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ satisfying $\mathbf{Az} \geq \mathbf{b}$. Then there exists a sub-MDP (E, A) of $\mathcal{M}$ which is a proper end component. We may assume, w.l.o.g., that for each $s \in E$ there exists a single $\alpha_s \in A(s)$. As for all $s \in E$ the actions in $A(s)$ are internal to the end component we have $\sum_{s' \in E} P(s,\alpha_s,s') = 1$ for all $s \in E$. Hence, from $\mathbf{Az} \geq \mathbf{b}$ we get for all $s \in E$

$$\mathbf{z}(s) \quad \geq \quad \mathbf{b}(s,\alpha_s) + \sum_{s' \in E} P(s,\alpha_s,s') \cdot \mathbf{z}(s') \quad \geq \quad \mathbf{b}(s,\alpha_s) + \min_{s' \in E}\{\mathbf{z}(s')\}.$$

But for $s \in E$ such that $\mathbf{z}(s) = \min_{s' \in E}\{\mathbf{z}(s')\}$ this implies $\mathbf{z}(s) \geq \mathbf{b}(s,\alpha_s) + \mathbf{z}(s) > \mathbf{z}(s)$, which yields is a contradiction. □

Remark 3.29 (Certifying non-EC-freeness). It is worth pointing out that with Farkas' Lemma, the above statement can be used to derive a certificate for the property that an MDP is *not* EC-free. Applying Farkas' Lemma (Lemma 2.2) yields that there exists $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ such that $\mathbf{Az} \geq \mathbf{1}$ if and only if there is no $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ such that $\mathbf{yA} \leq \mathbf{0}$ and $\mathbf{y} \cdot \mathbf{1} > 0$ (by multiplying the inequalities with -1). Now the above lemma (Lemma 3.28) implies that $\mathcal{M}$ is not EC-free if and only if such a vector $\mathbf{y}$ exists. Compare this with Lemma 3.8, which shows that indeed the support of vectors satisfying $\mathbf{yA} \leq \mathbf{0}$ contains only states which are included in proper end components. △

A comparison with the contraction property. In [Kal83] an MDP is defined to be *contracting* if there exists $\gamma \in [0,1)$ and strictly positive $\mathbf{x} \in \mathbb{R}^{S}_{>0}$ satisfying

$$\gamma \cdot \mathbf{x}(s) \geq \sum_{s' \in S} P(s,\alpha,s') \cdot \mathbf{x}(s')$$

for all enabled state-action pairs (s,α). In their model the transition probability function is assumed to be substochastic, which means that the probabilities of a state-action pair should sum up to *at most* one. For a comparison with our setting, we will assume that all remaining probability is added to a transition to "exit", and that an absorbing state "target" can be reached from every state apart from "exit". It is shown that an MDP is contracting iff all schedulers reach $\{\text{target}, \text{exit}\}$ with probability one (see [Kal83, Theorem 3.2.4]). As before, the states target and exit are excluded from the set S. The latter condition is equivalent to being EC-free. Hence, the definition of the contraction property yields another certificate condition for EC-freeness.

The two are very closely connected, as we now briefly discuss. Let $\mathbf{x}, \gamma$ be as defined above

and satisfying the contraction property. Then

$$\mathbf{x}(s) > \sum_{s' \in S} P(s, \alpha, s') \cdot \mathbf{x}(s')$$

holds for all $(s, \alpha) \in \mathcal{E}$, and hence we can find $\mathbf{b} \in \mathbb{R}^{\mathcal{E}}_{>0}$ such that $\mathbf{Ax} \geq \mathbf{b}$ holds. Thereby $\mathbf{x}$ is also a certificate by the condition of Lemma 3.28. Now let $\mathbf{b} \in \mathbb{R}^{\mathcal{E}}_{>0}$ be strictly positive, and let $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ satisfy $\mathbf{Az} \geq \mathbf{b}$. This means that

$$\mathbf{z}(s) \geq \mathbf{b}(s, \alpha) + \sum_{s' \in S} P(s, \alpha, s') \cdot \mathbf{z}(s')$$

holds for all $(s, \alpha) \in \mathcal{E}$. It follows that $\mathbf{z}$ is strictly positive in all entries. Now define

$$\gamma = \max_{(s,\alpha) \in \mathcal{E}} \left\{ \frac{\sum_{s' \in S} P(s, \alpha, s') \cdot \mathbf{z}(s')}{\mathbf{z}(s)} \right\}.$$

One can check that $0 < \gamma < 1$ holds and $\mathbf{z}, \gamma$ satisfy the contraction property.

Two algorithms to check EC-freeness are proposed in [Kal83, page 48]. The first (*Algorithm IV*) checks that all states have a positive minimal probability of leaving S within $|S|$ steps. The second algorithm (*Algorithm V*) checks whether the linear program characterizing the expected number of visits is unbounded. None of these algorithms is certifying, as they do not directly yield a solution of the inequality which defines contraction.

Certifying the MEC decomposition. We now use the above lemmas to provide a certificate condition for the fact that the decomposition into maximal end components was computed correctly. Combining Proposition 3.26 with Lemmas 3.27 and 3.28 yields the following theorem.

Theorem 3.30. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, and $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be a set of sub-MDPs of $\mathcal{M}$. Then, $\mathcal{D}$ equals the set of maximal end components of $\mathcal{M}$ if and only if:*

- *for each $(E_i, A_i) \in \mathcal{D}$ there exist functions $\mathtt{fwd}_i, \mathtt{bwd}_i$ which satisfy the conditions of Lemma 3.27 with respect to the underlying graph of (E_i, A_i), and*
- *there exists $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ such that $\mathbf{A}_{\mathcal{D}}\mathbf{z} \geq \mathbf{1}$ holds, where $\mathbf{A}_{\mathcal{D}}$ is the system matrix of the induced quotient $\mathcal{M}^{\text{target}}_{/\mathcal{D}}$.*

Example 3.31. Consider again the MDP $\mathcal{M}_3$ in Figure 3.4. If we ignore target and exit, $\mathcal{M}_3$ contains two maximal end components, namely $(E_1, A_1) = \big(\{s_{in}, s_1, s_2\}, (s_{in} \mapsto \{\beta\}, s_1 \mapsto \{\alpha, \beta\}, s_2 \mapsto \{\beta\})\big)$ and $(E_2, A_2) = \big(\{s_3\}, (s_3 \mapsto \varnothing)\big)$. Only the first one is proper. To verify that these two sub-MDPs indeed are precisely the maximal end components of $\mathcal{M}_3$, Theorem 3.30 proposes to provide certificates that the sub-MDPs are maximal end components, and that the resulting quotient is EC-free.

To show that (E_1, A_1) is a maximal end component, we have to check that the induced graph of (E_1, A_1) is strongly connected. Let $\mathtt{fwd}, \mathtt{bwd}$ be defined as follows:

$$\mathtt{bwd}(s_{in}) = \mathtt{fwd}(s_{in}) = 0 \quad \text{and} \quad \mathtt{bwd}(s_1) = \mathtt{bwd}(s_2) = \mathtt{fwd}(s_1) = \mathtt{fwd}(s_2) = 1.$$

These functions satisfy the requirements of Lemma 3.27. For (E_2, A_2) it suffices to set $\texttt{fwd}(s_3) = \texttt{bwd}(s_3) = 0$.

Let $\mathcal{D} = \{(E_1, A_2), (E_2, A_2)\}$. To certify that $\mathcal{M}_{3/\mathcal{D}}^{\text{target}}$ is EC-free, observe that the equation system $\mathbf{A}_{\mathcal{D}}\mathbf{z} \geq \mathbf{1}$ as defined in Theorem 3.30 is given by

$$(s_{in}, \alpha):\ {}^1\!/_4 \cdot \mathbf{z}(E_1) \geq 1 + {}^1\!/_4 \cdot \mathbf{z}(E_2) \quad (s_2, \alpha):\ {}^3\!/_4 \cdot \mathbf{z}(E_1) \geq 1 + {}^1\!/_2 \cdot \mathbf{z}(E_2) \quad (s_3, \alpha):\ \mathbf{z}(E_2) \geq 1.$$

Here the individual constraints correspond to the state-action pairs which are not internal in any of the sub-MDPs included in $\mathcal{D}$. Any solution of this system of inequalities proves that $\mathcal{M}_{/\mathcal{D}}$ is EC-free by Lemma 3.28. For example, $\mathbf{z}(E_2) = 1$ and $\mathbf{z}(E_1) = 5$ is such a solution. Together, $\mathbf{z}$ and the certificates for the individual maximal end components certify that $\mathcal{D}$ constitutes the set of maximal end components of $\mathcal{M}_3$. △

3.2 Farkas certificates for expected rewards

The previous section introduced Farkas certificates for probabilistic reachability constraints which were derived from the linear programming characterization of optimal reachability probabilities and Farkas' Lemma. In the following we will consider how these methods can be used to define certificates for constraints on the optimal expected total reward that is achievable in an MDP.

Consider an MDP $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ in *reward reachability form* (as defined in Section 2.2.3), which implies that all schedulers $\mathfrak{S}$ for $\mathcal{M}$ satisfy $\Pr_s^{\mathfrak{S}}(\Diamond\,\text{exit}) = 1$ for all states $s \in S$. In particular, $\mathcal{M}$ is also EC-free. We will consider the following constraints on the optimal expected reward achievable in $\mathcal{M}$:

I. All schedulers $\mathfrak{S}$ for $\mathcal{M}$ satisfy $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$ (i.e., $\mathbb{E}^{\min}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$).

II. All schedulers $\mathfrak{S}$ for $\mathcal{M}$ satisfy $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$ (i.e., $\mathbb{E}^{\max}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$).

III. Some scheduler $\mathfrak{S}$ for $\mathcal{M}$ satisfies $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$ (i.e., $\mathbb{E}^{\max}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$).

IV. Some scheduler $\mathfrak{S}$ for $\mathcal{M}$ satisfies $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$ (i.e., $\mathbb{E}^{\min}_{\mathcal{M},s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$).

These are defined completely analogously to the probabilistic reachability constraints. To derive Farkas certificates for these constraints on the achievable expected reward, we make use of the fact that the optimal reachability probabilities and expected rewards are characterized by very similar linear programs.

Let $\mathbf{A}$ be the system matrix of $\mathcal{M}$ (Definition 2.7) and $\mathbf{r} \in \mathbb{R}^{\mathcal{E}}$ be the vector containing the reward of each state-action pair (henceforth called the *reward vector*), i.e., $\mathbf{r}(s, \alpha) = \text{rew}(s, \alpha)$ for all enabled state-action pairs $(s, \alpha) \in \mathcal{E}$. By Lemma 2.13, vectors $\mathbf{z} \in \mathbb{R}^S$ satisfying the systems of inequalities $\mathbf{Az} \geq \mathbf{r}$ or $\mathbf{Az} \leq \mathbf{r}$ yield point-wise bounds on the vectors $\mathbf{ex}^{\max}$ and $\mathbf{ex}^{\min}$. These vectors contain the optimal values for the expected total reward in each state (see Section 2.2.3). As a direct consequence we get the following proposition.

Proposition 3.32. *Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be in reward reachability form, $\mathbf{A}, \mathbf{r}$ be defined as above and $\gtrsim \in \{\geq, >\}, \lesssim \in \{\leq, <\}$. Then for all $\lambda \geq 0$:*

1. *There exists $\mathbf{z} \in \mathbb{R}^S$ satisfying $\mathbf{Az} \leq \mathbf{r}$ and $\mathbf{z}(s_{in}) \gtrsim \lambda$ if and only if $\mathbb{E}^{\min}_{s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$ holds.*
2. *There exists $\mathbf{z} \in \mathbb{R}^S$ satisfying $\mathbf{Az} \geq \mathbf{r}$ and $\mathbf{z}(s_{in}) \lesssim \lambda$ if and only if $\mathbb{E}^{\max}_{s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$ holds.*

To derive certificates for statements III. and IV., which require the existence of a scheduler meeting the bound, we again use Farkas' Lemma. The proof is very similar to the one of the corresponding proposition for reachability probabilities (Proposition 3.4).

One difference is that we cannot assume the vectors $\mathbf{ex}^{\max}$ and $\mathbf{ex}^{\min}$ to be nonnegative, in contrast to the vectors containing optimal reachability probabilities. This dissallows using the version of Farkas' Lemma (Lemma 2.2) used in Proposition 3.4. However, we can use the main formulation of Farkas' Lemma (Lemma 2.1) and the observation that the systems of inequalities $\mathbf{yA} = \delta_{s_{in}} \wedge \mathbf{yr} \lesssim \lambda$ and $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{yr} \lesssim \lambda$ are equisatisfiable if $\mathcal{M}$ is EC-free, which is a consequence of Lemma 3.17. The analogous statement holds for $\mathbf{yA} = \delta_{s_{in}} \wedge \mathbf{yr} \gtrsim \lambda$ and $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yr} \gtrsim \lambda$.

Proposition 3.33. *Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP in reward reachability form, $\mathcal{E}$ be the enabled state-action pairs of $\mathcal{M}$, $\mathbf{A}$ its system matrix and $\mathbf{r}$ its reward vector. Then, for $\gtrsim \in \{\geq, >\}$, $\lesssim \in \{\leq, <\}$ and $\lambda \in [0, 1]$:*

1. *There exists a row vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \geq \delta_{s_{in}}$ and $\mathbf{y}\,\mathbf{r} \lesssim \lambda$ if and only if $\mathbb{E}^{\min}_{s_{in}}(\diamondplus\,\text{exit}) \lesssim \lambda$ holds.*
2. *There exists a row vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \leq \delta_{s_{in}}$ and $\mathbf{y}\,\mathbf{r} \gtrsim \lambda$ if and only if $\mathbb{E}^{\max}_{s_{in}}(\diamondplus\,\text{exit}) \gtrsim \lambda$ holds.*

Proof. The proof is essentially analogous to the proof of Proposition 3.4. Hence, we will only consider the statement (1.) with $\lesssim = <$. Using Proposition 3.32 we get:

$$\mathbb{E}^{\min}_{s_{in}}(\diamondplus\,\text{exit}) < \lambda \quad \iff \quad \neg\, \exists \mathbf{z} \in \mathbb{R}^{S}.\ \ \mathbf{Az} \leq \mathbf{r} \wedge \mathbf{z}(s_{in}) \geq \lambda.$$

Applying Farkas' Lemma (Lemma 2.1) yields that the latter is equivalent to

$$\exists \mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}\ \exists y^* \geq 0.\ \ (\mathbf{y}, y^*) \begin{pmatrix} \mathbf{A} \\ -1\ 0 \ldots 0 \end{pmatrix} = \mathbf{0} \wedge (\mathbf{y}, y^*) \begin{pmatrix} \mathbf{r} \\ -\lambda \end{pmatrix} < 0.$$

We assume here that the first row of $\mathbf{z}$ corresponds to s_{in}. Now it follows by the same argument as in the proof of Proposition 3.4 that this statement is equivalent to $\mathbf{yA} = \delta_{s_{in}} \wedge \mathbf{y}\,\mathbf{r} < \lambda$. As $\mathcal{M}$ is EC-free, it follows from Lemma 3.17 that there exists $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} = \delta_{s_{in}} \wedge \mathbf{y}\,\mathbf{r} < \lambda$ if and only if there exists $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{yA} \geq \delta_{s_{in}} \wedge \mathbf{y}\,\mathbf{r} < \lambda$, which concludes the proof of this case. □

The certificate one gets from the two propositions above can be interpreted in an analogous way as the Farkas certificates for reachability probabilities. In particular, the $\mathbf{y}$-vectors in the last proposition induce memoryless randomized schedulers whose expected number of visits for each state-action pair is bounded by the corresponding entries of the $\mathbf{y}$-vector (see Proposition 3.18). As we have seen, the expected total reward achieved by a scheduler is equivalent to the expected number of visits of each state times the reward in this state (see also Section 2.2.4). But this is exactly the expression $\mathbf{y}\,\mathbf{r}$ in the above proposition.

Remark 3.34 (Expected discounted reward). In contrast to the expected total reward, the *expected discounted reward* adds a discount factor which exponentially benefits rewards achieved early. For an MDP $\mathcal{M} = (S, \text{Act}, P, s_{in}, \text{rew})$ and a *discount factor* $\gamma \in (0, 1)$ define the random variable

$\mathrm{drew}_\gamma : \mathrm{Paths}(\mathcal{M}) \to \mathbb{R}$ by

$$\mathrm{drew}_\gamma(s_1\alpha_1 s_2\alpha_2 \ldots) \;=\; \sum_{1\leq i} \gamma^i \cdot \mathrm{rew}(s_i, \alpha_i).$$

We define $\mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(\mathrm{drew}_\gamma)$ to be the expected discounted reward under $\mathfrak{S}$, and as the above series converges this value is always finite. Minimal and maximal expected discounted rewards can be defined as for the expected total reward.

It is shown in [Kal83, Remark 3.4.4] that considering the expected total reward criterion in EC-free MDPs is essentially the same thing as considering the expected discounted reward in arbitrary MDPs. In particular, if we are interested in the discounted problem sketched above, we can construct an MDP $\mathcal{M}' = (S \cup \{\mathrm{exit}\}, \mathrm{Act}, P', s_{in}, \mathrm{rew})$ in reward reachability form by setting $P'(s,\alpha,s') = \gamma \cdot P(s,\alpha,s')$ for all $s, s' \in S$ and $\alpha \in \mathrm{Act}$, and additionally $P'(s,\alpha,\mathrm{exit}) = 1-\gamma \sum_{s'\in S} P'(s,\alpha,s')$. The resulting MDP $\mathcal{M}'$ is clearly EC-free. Furthermore, the expected total reward in $\mathcal{M}'$ equals the expected discounted reward in $\mathcal{M}$ under all schedulers, and hence, in particular, Farkas certificates for constraints on the former model can be used as Farkas certificates for constraints on the latter. △

3.3 Computing and validating Farkas certificates

This section is concerned with the computation of Farkas certificates as defined in Definition 3.23 for probabilistic reachability constraints and for constraints on the expected reward in Propositions 3.32 and 3.33. We will first show how to do this using linear programs, and then discuss how to obtain Farkas certificates using value iteration or policy iteration.

3.3.1 Computing Farkas certificates using linear programs

All types of Farkas certificates can be computed by finding a solution to a system of linear inequalities. It is well-known that this problem is computationally very close to the problem of solving a linear program [Sch99, Theorem 10.4]. We will distinguish whether the threshold condition is a strict inequality, where the certificate condition can be written in the form

$$\mathbf{M}\mathbf{x} \geq \mathbf{b} \;\wedge\; \mathbf{c}\,\mathbf{x} < \theta, \tag{3.2}$$

or a non-strict inequality, where we simply have a polyhedron described by linear inequalities such as

$$\mathbf{M}\mathbf{x} \geq \mathbf{b}. \tag{3.3}$$

Here $\mathbf{M} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ are generic matrices, vectors and numbers meant to describe any of the Farkas certificate conditions, and $\mathbf{x}$ is a vector of variables of dimension n. To find a solution of Equation (3.2) one can in a first step solve the linear program: *minimize* $\mathbf{c}\,\mathbf{x}$ *under the condition* $\mathbf{M}\mathbf{x} \geq \mathbf{b}$, and then check whether the solution vector satisfies $\mathbf{c}\,\mathbf{x} < \theta$. In this case the solution vector satisfies the condition, and otherwise no solution of Equation (3.2) exists.

Checking satisfiability of a set of non-strict linear inequalities such as Equation (3.3) may be seen as a simple instance of a linear programming problem. Algorithms aimed at solving it are often referred to as *phase 1* methods, because standard algorithms for linear programming such as the simplex method apply them as a first step (see [CLRS09, Section 29.5]). Still, as shown

in [Sch99], the problem of solving a linear program reduces in linear time to the problem of solving a set of linear inequalities, and hence the latter cannot be considered much simpler theoretically. A straight forward method to solve a system of non-strict linear inequalities is to solve a linear program over this system with an arbitrary objective function.

Robust certificates

In most cases, especially when efficiency is a concern, computer programs are based on floating point arithmetic and do not use exact rational arithmetic. This is true in particular for most optimization software, including many linear programming solvers. Optimal solutions of a linear program are always attained on the boundary of the feasible region. Hence, approaches to compute Farkas certificates using linear programming as described above will generally yield Farkas certificates on the boundary of the corresponding polyhedra. Such certificates are susceptible to rounding errors, as slight deviations of the vector may no longer be valid certificates. Similarly, if the tool validating the certificate uses floating point arithmetic it may return a false negative, even though the certificate is valid. The challenges and caveats of using floating point arithmetic in LP-solvers are discussed in [ACDE07].

This raises the question of how to efficiently compute Farkas certificate which are *not on the boundary* of the corresponding polyhedron. Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$ and consider the system of linear inequalities given by $\mathbf{Mx} \geq \mathbf{b}$. To compute an interior solution of this system one can use the following trick. Let s be an additional variable and $\mathbf{s} = (s, \ldots, s)$ be the vector of dimension m containing s in each entry. Consider the linear program

$$\textit{maximize } s \textit{ under the condition} \quad \mathbf{Mx} \geq \mathbf{b} + \mathbf{s}.$$

If the optimal value s^* of this LP satisfies $s^* > 0$, then the $\mathbf{x}$-variables of the corresponding solution yield a strictly feasible solution of $\mathbf{Mx} \geq \mathbf{b}$. If, on the other hand $s^* = 0$ holds, then $\mathbf{Mx} \geq \mathbf{b}$ is satisfiable but there is no solution which is strictly feasible in all constraints. Finally, if $s^* < 0$ holds, then $\mathbf{Mx} \geq \mathbf{b}$ is infeasible. This is a standard trick in mathematical optimization (see [BV14, Section 11.4]) and can be used to compute strictly feasible Farkas certificates.

3.3.2 Computing Farkas certificates using value- or policy iteration

Two classes of algorithms which can be applied to a wide range of problems in the context of probabilistic model checking are *value iteration* and *policy iteration*. In particular, both can be used to compute the optimal reachability probabilities in Markov decision processes [Put94, FKNP11]. Value iteration offers the advantage that it can usually be implemented in a straight-forward manner for systems encoded symbolically using *multi-terminal binary decision diagrams* [deAKN+00], which is not true for linear programming based approaches.

Value iteration. Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an EC-free Markov decision process in reachability form with system matrix $\mathbf{A}$ and target vector $\mathbf{t}$. To compute (or rather approximate) the minimal probabilities of reaching "target" in $\mathcal{M}$ using value iteration, we start with the vector $\mathbf{z}_1 = \mathbf{0}$ and then iteratively compute:

$$\mathbf{z}_{i+1}(s) = \min_{\alpha \in \text{Act}(s)} \{\, \mathbf{t}(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot \mathbf{z}_i(s') \,\}.$$

One can show that for all $i \geq 1$ we have $\mathbf{A}\mathbf{z}_i \leq \mathbf{t}$ and that the sequence converges to $\mathbf{pr}^{\min}$ [deA97, BK08] (here we use that the MDP is in reachability form thus has positive probability to reach "target" from any state, and we have assumed EC-freeness). It follows that all the vectors $\mathbf{z}_i$ are Farkas certificates for the property $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond \text{target}) \geq \mathbf{z}_i(s_{in})$ (see Definition 3.23). To compute a Farkas certificate for $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond \text{target}) \triangleright \lambda$, for a given $\lambda \in [0,1]$ and $\triangleright \in \{\geq, >\}$, one can repeat the value iteration until a vector $\mathbf{z}_i$ is computed such that $\mathbf{z}_i(s_{in}) \triangleright \lambda$ holds. Unless λ equals the minimal reachability probability in s_{in}, and $\triangleright = \geq$, this process will eventually terminate. The fact that intermediate solutions obtained during value iteration are lower bounds on the optimal value is well-known, and it follows directly from Lemma 2.9.

For maximal reachability probabilities, one can start with $\mathbf{z}_1 = \mathbf{1}$ and use the update rule

$$\mathbf{z}_{i+1}(s) = \max_{\alpha \in \mathrm{Act}(s)} \{\, \mathbf{t}(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot \mathbf{z}_i(s') \,\}.$$

Analogously to the previous case, we have $\mathbf{A}\mathbf{z}_i \geq \mathbf{t}$ for all $i \geq 1$ and the sequence converges to $\mathbf{pr}^{\max}$. Hence, the intermediate vectors are all Farkas certificates for $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond \text{target}) \leq \mathbf{z}(s_{in})$. This again depends on EC-freeness, as the sequence may not converge to $\mathbf{pr}^{\max}$ otherwise.

The standard way of applying value iteration differs from the above description in that for both minimal and maximal probabilities one would use $\mathbf{z}_1 = \mathbf{0}$ as initial vector. While this is also correct, the intermediate vectors are no longer Farkas certificates in the case of maximal probabilities. Using different starting vectors for value iteration has been considered in the literature. In particular, *interval iteration* uses two value iterations, from above and from below, to provide sound stopping criteria [HM14]. In their terminology, our above description matches value iteration from below for minimal probabilities, and from above for maximal probabilities.

We now comment on the case that $\mathcal{M}$ is not EC-free. The minimal reachability probability of all states included in proper end components is zero, and it is necessary to compute these states a priori. They can be identified with "exit", which yields an EC-free MDP. Then the above arguments can be applied. Now let us consider maximal reachability probabilities. While value iteration from below (i.e., using $\mathbf{z}_1 = \mathbf{0}$ as initial vector) does converge to $\mathbf{pr}^{\max}$, this is not true for value iteration from above (using $\mathbf{z}_1 = \mathbf{1}$) [HM14]. Hence, one has to first compute the quotient of maximal end components (see Section 2.2.2, and also the *max-reduced* MDP in [HM14]). Then the above arguments again apply, as the resulting MDP is EC-free. Having computed a Farkas certificate $\mathbf{z}$ for $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond \text{target}) \leq \lambda$ in the MEC-quotient, one can derive a Farkas certificate in terms of the original system by setting $\mathbf{z}'(s) = \mathbf{z}([s])$, where $[s]$ denotes the maximal end component of s.

The main obstacle of applying this approach to Farkas certificates for expected total rewards is the computation of initial vectors $\mathbf{z}_1$ satisfying $\mathbf{A}\mathbf{z}_1 \leq \mathbf{r}$ (for $\mathbb{E}^{\min}_{s_{in}}(\diamondplus \text{exit}) \geq \lambda$), respectively $\mathbf{A}\mathbf{z}_1 \geq \mathbf{r}$ (for $\mathbb{E}^{\max}_{s_{in}}(\diamondplus \text{exit}) \leq \lambda$). For nonnegative reward functions and certificates for $\mathbb{E}^{\min}_{s_{in}}(\diamondplus \text{exit}) \geq \lambda$, one can choose $\mathbf{z}_1 = \mathbf{0}$. But in general, it is not clear how to find such initial vectors. While computing upper bounds $\mathbf{z}$ on $\mathbf{ex}^{\max}$ as starting points for value iteration has been studied [BKL+17], these are not guaranteed to satisfy $\mathbf{A}\mathbf{z} \geq \mathbf{r}$. Such upper bounds are called *inductive* in [HK20].

We have shown how one can use value iteration to compute Farkas certificates for reachability constraints of the form $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond \text{target}) \geq \lambda$ and $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond \text{target}) \leq \lambda$. These are the properties which assert that *all* schedulers satisfy a certain threshold condition. Farkas certificates for the properties which assert that *some* scheduler satisfies a bound can be computed more naturally in the context of *policy iteration*.

Policy iteration In policy iteration a sequence of memoryless deterministic schedulers (sometimes also called policies) is computed whose induced reachability probabilities increase (if computing the maximal probability) or decrease (if computing the minimal probability). The resulting algorithms are usually exponential (as is the set of MD-schedulers) but belong to the main approaches of solving MDPs and have been proved useful in various context [Put94, FKNP11, KM17].

The following describes a simple policy iteration scheme for computing $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target})$. First, pick arbitrary MD-scheduler $\mathfrak{S}_1$. Construct a sequence of MD-schedulers $\mathfrak{S}_1, \mathfrak{S}_2, \ldots$ as follows, starting with $i = 1$:

- Let

$$\mathbf{v}_i(s) = \Pr^{\mathfrak{S}_i}_s(\Diamond\,\text{target}) \qquad \text{and} \qquad \mathbf{v}_i(s,\alpha) = \mathbf{t}(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot \mathbf{v}_i(s'),$$

  for all $s \in S$ and $\alpha \in \mathrm{Act}(s)$.

- Let $\alpha_s = \arg\max_{\alpha \in \mathrm{Act}(s)} \{\, \mathbf{v}_i(s,\alpha) \,\}$ and set

$$\mathfrak{S}_{i+1}(s) = \begin{cases} \alpha_s & \text{if } \mathbf{v}_i(s,\alpha_s) > \mathbf{v}_i(s,\mathfrak{S}_i(s)) \\ \mathfrak{S}_i(s) & \text{otherwise.} \end{cases}$$

This process is repeated until $\mathfrak{S}_i = \mathfrak{S}_{i+1}$ holds, which implies that $\mathfrak{S}_i$ is a scheduler satisfying $\Pr^{\mathfrak{S}_i}_s(\Diamond\,\text{target}) = \mathbf{Pr}^{\max}_s(\Diamond\,\text{target})$ for all $s \in S$. If $\mathfrak{S}_i$ is the resulting scheduler, then the vector $\mathbf{ev}^{\mathfrak{S}_i}$ containing the expected number of visits in $\mathcal{M}$ under $\mathfrak{S}_i$ from s_{in} forms a Farkas certificate for $\mathbf{Pr}^{\max}_{s_{in}}(\Diamond\,\text{target}) \geq \lambda$, for any λ which is indeed a lower bound on the maximal probability. This follows directly from Proposition 3.13 and the fact that $\mathbf{ev}^{\mathfrak{S}_i} \cdot \mathbf{t} = \Pr^{\mathfrak{S}_i}_{s_{in}}(\Diamond\,\text{target})$ holds. We may assume that $\mathfrak{S}_i$ satisfies $\Pr^{\mathfrak{S}_i}_s(\Diamond\,\text{target}) > 0$ from every state $s \in S$, as $\mathcal{M}$ is in reachability form and hence every state $s \in S$ has a path to target. Computing $\mathbf{ev}^{\mathfrak{S}_i}$ amounts to solving a system of linear equalities, as discussed in Section 2.2.4.

An analogous algorithm can be used for minimal reachability probabilities. If the MDP is not EC-free, one has to collapse all states which do not reach $\{\text{target}, \text{exit}\}$ in $\mathcal{M}^{\mathfrak{S}_i}$ and identify them with "exit". Then, the expected number of visits under $\mathfrak{S}_i$ in the resulting MDP yield a Farkas certificate for $\mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target}) \leq \lambda$, for any $\lambda \geq \mathbf{Pr}^{\min}_{s_{in}}(\Diamond\,\text{target})$.

Remark 3.35. For EC-free MDP, the policy iteration scheme is polynomial in the number of states and actions of the MDP, if one uses the right pivoting rule [Ye11, Corollary 5.1]. The upper bound on the number of required iterations given in this paper is, however, exponential in the values of the transition probabilities (when encoded in binary). △

3.3.3 Validating Farkas certificates

To validate a Farkas certificate, by which we mean checking that a given vector $\mathbf{x}$ indeed is a Farkas certificate for some property, it suffices to check whether $\mathbf{Mx} \geq \mathbf{b}$ (or $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{cx} < \theta$) holds for the system of linear inequalities which defines the corresponding set of Farkas certificates. Both of these checks can be done in linear time by computing $\mathbf{Mx}$ (and additionally $\mathbf{cx}$, in case of strict inequalities), and checking whether each value of the result satisfies the corresponding threshold condition. Hence, validating Farkas certificate is significantly simpler than finding a Farkas certificate, which requires solving a linear program in general.

It was mentioned above that the linear programming based approaches to compute Farkas certificates cannot be easily realized if the system is encoded symbolically using multi-terminal binary decision diagrams, which is a common representation for probabilistic systems. However, this representation still allows *validating* Farkas certificates efficiently, because matrix multiplication is an operation which can be computed efficiently in this setting [FMY97].

Chapter 4

New techniques for witnessing subsystems

Given a system $\mathcal{M}$ and a property ϕ which is satisfied by $\mathcal{M}$, it is natural to ask: Which part of $\mathcal{M}$ contributes most towards the satisfaction of ϕ? Being able to answer such questions can be extremely useful to increase the understanding of the system by a user, to aid debugging of programs or to assign responsibility to software components or particular code fragments if something goes wrong. It can also help the automated analysis of systems, by restricting the analysis to relevant parts.

In order to approach this question one has to formalize what constitutes a "part" of the system and when such a part is relevant or important with respect to a given specification. For probabilistic reachability constraints in Markov decision processes, the notion of *subsystem* [JÁK+11] achieves this purpose. If $\mathcal{M}$ is an MDP in reachability form, then a subsystem of $\mathcal{M}$ is any MDP obtained from $\mathcal{M}$ by redirecting a subset of its transitions to the state "exit". As "exit" cannot reach the target state by definition, redirecting a transition t to "exit" effectively means making a worst case assumption for t. The underlying question is: What happens to the global probability of reaching "target" in $\mathcal{M}$ if we assume that t contributes zero probability?

A subsystem $\mathcal{M}'$ of $\mathcal{M}$ is called a *witness* for the probabilistic reachability constraint $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$, if $\mathcal{M}'$ itself satisfies this constraint (with $\mathfrak{m} \in \{\min, \max\}$). As the optimal probability of reaching "target" in a subsystem cannot be higher than in $\mathcal{M}$ by construction, the name *witness* is justified. However, checking whether a subsystem is a witness is as hard as checking whether $\mathcal{M}$ satisfies the property in general, which distinguishes witnesses from Farkas certificates. In the foundational work on witnessing subsystems [JÁK+11, WJÁ+12, WJÁ+14] they were called *critical subsystems*, as the main intention was to provide *counterexamples* for properties of the form $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) < \lambda$. This is only a matter of naming, however, and we use the term *witnessing subsystem* as it is more in line with the idea of certification and lets us avoid artificially introducing a negation in many cases.

A witnessing subsystem is more useful the *smaller* it is. Small witnessing subsystems correspond to small parts of the system which *by themselves* contain enough probability to achieve the required threshold. Hence the essential algorithmic question is to compute minimal

witnessing subsystems with respect to some notion of size.

This chapter describes a tight connection between Farkas certificates and witnessing subsystems. It is based on the observation that entries in Farkas certificates with value zero correspond to states (or state-action pairs) which can be removed to form a witnessing subsystem. More precisely, from a Farkas certificate $\mathbf{c}$ we can derive a witnessing subsystem whose states correspond to the non-zero entries of $\mathbf{c}$, and vice versa.

The presented framework extends naturally to witnessing subsystems for lower bounds on both the *maximal* and *minimal* reachability probabilities in MDPs. Previously, only witnessing subsystems for lower bounds on the maximal probability were considered [WJÁ+12]. A witnessing subsystem for lower bounds on the minimal reachability probability gives the guarantee that *all schedulers* of $\mathcal{M}$ satisfy the given lower bound. Furthermore, the framework extends directly to witnessing subsystems for lower bounds on the expected total accumulated reward before reaching "exit", under a nonnegative reward function.

The connection to Farkas certificates gives rise to new algorithms and heuristics for computing minimal witnessing subsystems. They are shown to be competitive with existing approaches through an experimental evaluation. A further contribution of the chapter is to show that (the decision version of) computing minimal witnessing subsystems is NP-hard already for *acyclic Markov chains*. This settles the question of the precise complexity of this problem for Markov chains, which was raised in [WJÁ+12].

Related work

Large parts of the literature on witnesses for probabilistic systems have been described in Chapter 1. We focus here on the works which are most closely related the results of this chapter.

The notion of subsystems as witnesses for probabilistic reachability constraints in Markov chains was first defined in [JÁK+11], which includes heuristics aimed at computing witnessing subsystems with few states. They are based on iteratively extending subsystems which do not yet satisfy the lower bound on the probability by adding paths. Each iteration includes a model checking step, to check whether the current subsystem is a witness. The tool Comics [JÁV+12] implements these heuristics. They are fundamentally different from the quotient-sum heuristic which we describe, as the latter does not rely on iteratively expanding a subsystem.

This approach was extended to lower bounds on the maximal reachability probability in MDPs, to ω-regular properties and high-level counterexamples [WJÁ+12, WJV+13, WJÁ+14, Jan15]. Exact algorithms based on mixed-integer linear programs (MILPs) to compute (state-) minimal witnessing subsystem were presented in [WJÁ+12]. The objective functions of these MILPs aim at finding vectors with many zero-entries, similarly to our approach. One of the two MILP-formulations which we give for DTMCs coincides with the one in [WJÁ+12], whereas the MILPs for MDPs differ. The idea of minimizing subsystems with respect to the number of labels they include was proposed in [WJV+13]. Our MILPs computing label-minimal witnesses use similar ideas as described there.

A related notion of witnessing subsystem for MDPs has been studied in [CV10]. They considered a safety-fragment of PCTL, and only witnesses for lower bounds on the maximal probability of satisfying a path formula. Computing minimal witnessing subsystem for reachability in MDPs (rather, the associated decision problem) is determined to be NP-complete [CV10], but the precise complexity for Markov chains has so far been open [WJÁ+12, WJÁ+14].

Witnessing subsystems for constraints on the expected total reward for Markov chains were first considered in [QJD+15]. This work introduces two kinds of witnessing subsystems, and

one of them fits very naturally into our framework. The heuristics introduced in [QJD+15] are based on the known techniques for computing small witnessing subsystems for probabilistic reachability constraints [AL06, JÁK+11]. In contrast, we consider also witnessing subsystems for constraints on the expected total reward in MDPs, and show that the connection to Farkas certificates, and the resulting algorithms, extend naturally to this case.

The MILP we introduce to compute minimal witnesses for lower bounds on the maximal reachability probability in MDPs depends on an upper bound on the expected number of visits of state-action pairs under all deterministic and memoryless schedulers which reach {target, exit} with probability one. Computing such bounds is not trivial, and [BKL+17] considers this problem for EC-free MDPs. In particular, they leverage the structure of the underlying graph to compute better bounds.

Outline

The chapter starts by defining subsystems (Definition 4.1) and showing that their minimal and maximal probability cannot increase with respect to the original system (Proposition 4.4). *Witnessing* subsystems are defined in Definition 4.5. Section 4.1.1 considers several notions of size of subsystems and Section 4.1.2 shows that computing minimal witnessing subsystems is NP-complete already for acyclic Markov chains (Theorem 4.16). Then, Section 4.2 discusses the connection between Farkas certificates and witnessing subsystems for the same properties (Theorem 4.23). This connection leads to mixed-integer linear programs for computing minimal witnessing subsystems, which are described in Section 4.2.1. Section 4.2.3 introduces the *quotient-sum heuristic* which computes Farkas certificates with small support by solving a sequence of linear programs. Section 4.3 describes the correspondence of Farkas certificates and witnessing subsystems for the expected total reward criterion, and Section 4.4 discusses how witnessing subsystems for *invariants* can be computed.

Relation to published work

The main results of this chapter have been published in [FJB20], which is joint work with Florian Funke and Christel Baier. In comparison to [FJB20], all results are extended to MDPs with proper end components, which pose additional challenges as the maximal number of expected visits in a state may now be unbounded. Furthermore, we consider label-based and weighted minimization problems, witnessing subsystems for constraints on the expected total reward, and witnessing subsystems for invariants, all of which were not discussed in [FJB20]. The discussion of the quotient-sum heuristic has also been extended significantly, and is partly based on [JHFB20]. This paper is joint work with Hans Harder, Florian Funke and Christel Baier and introduces the tool Switss. Finally, the chapter includes experimental studies which in parts were also presented in [JHFB20].

4.1 Witnessing subsystems

We start with the definition of a subsystem. Given an MDP $\mathcal{M}$ with dedicated states "target" and "exit", a subsystem of $\mathcal{M}$ is any MDP one may get by taking $\mathcal{M}$ and redirecting some of its transitions to the state "exit". Formally we define it as follows.

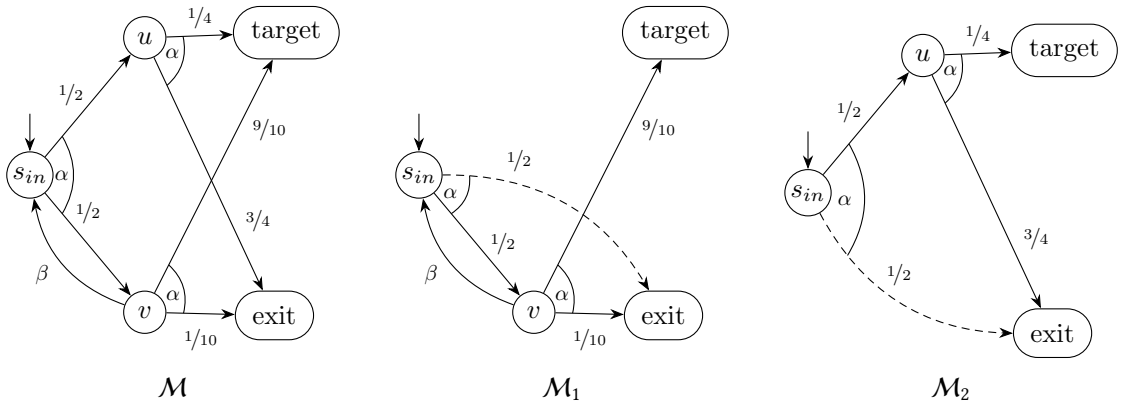


Figure 4.1: An MDP $\mathcal{M}$ together with two subsystems $\mathcal{M}_1$ and $\mathcal{M}_2$. The subsystem $\mathcal{M}_1$ is induced by states $\{s_{in}, v\}$ and $\mathcal{M}_2$ is induced by states $\{s_{in}, u\}$.

Definition 4.1 (Subsystem). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP. An MDP $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ is a *subsystem* of $\mathcal{M}$ if $s_{in} \in S' \subseteq S$, target and exit are absorbing in $\mathcal{M}'$ and for all $s, s' \in S' \cup \{\text{target}\}$ and $\alpha \in \text{Act}$:

1. $P'(s, \alpha, s') \in \{\, 0, P(s, \alpha, s') \,\}$, and
2. α is enabled in s in $\mathcal{M}'$ if and only if α is enabled in s in $\mathcal{M}$.

The fact that s' ranges only over $S' \cup \{\text{target}\}$ in the above definition makes sure that whenever $P'(s, \alpha, s') = 0$ and $P(s, \alpha, s') > 0$ hold (i.e., some transition of the original MDP is removed), then the missing probability is added to the transition towards exit in $\mathcal{M}'$. (The outgoing probabilities of an enabled state-action pair must add up to one.) This matches the intuitive description that a subsystem is produced by redirecting transitions to the state "exit".

Definition 4.2 (Induced subsystems). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP and $S' \subseteq S$. The subsystem *induced by* S' is the MDP $\mathcal{M}_{S'} = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ where for all $s, s' \in S' \cup \{\text{target}\}$ and $\alpha \in \text{Act}$ we have:

$$P'(s, \alpha, s') = P(s, \alpha, s') \quad \text{and} \quad P'(s, \alpha, \text{exit}) = P(s, \alpha, \text{exit}) + \sum_{s' \in S \setminus S'} P(s, \alpha, s').$$

Observe that not all subsystems are induced by a set of states S', as a subsystem may exclude individual transitions without excluding any states.

Assume that $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ is an MDP in reachability form and consider a set of states $S' \subseteq S$ and $\mathcal{E}' = \{(s, \alpha) \in \mathcal{E} \mid s \in S'\}$. Then, the system matrix $\mathbf{A}' \in \mathbb{R}^{\mathcal{E}' \times S'}$ (as defined in Definition 2.7) for the subsystem $\mathcal{M}_{S'}$ induced by $S' \subseteq S$ is the restriction of the system matrix $\mathbf{A}$ of $\mathcal{M}$ to S' (i.e., $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$). Similarly, the target vector $\mathbf{t}'$ of $\mathcal{M}_{S'}$ is the restriction of the target vector $\mathbf{t}$ of $\mathcal{M}$ to S' (i.e., $\mathbf{t}' = \mathbf{t}|_{S'}$). This follows directly from the definition of the transition matrix of the induced subsystem and will be used later in the chapter to relate Farkas certificates of induced subsystems to Farkas certificates of the original system.

Example 4.3. Consider the MDP $\mathcal{M}$ defined in Figure 4.1 together with the two subsystems $\mathcal{M}_1$ and $\mathcal{M}_2$. While $\mathcal{M}_1$ has nondeterministic choice (in state v), $\mathcal{M}_2$ is purely probabilistic. The

optimal probabilities in these three MDPs are given as follows:

$$\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) = 1/4 \qquad \mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) = 23/40$$
$$\mathbf{Pr}^{\min}_{\mathcal{M}_1}(\Diamond\,\text{target}) = 0 \qquad \mathbf{Pr}^{\max}_{\mathcal{M}_1}(\Diamond\,\text{target}) = 9/20$$
$$\mathbf{Pr}^{\min}_{\mathcal{M}_2}(\Diamond\,\text{target}) = 1/8 \qquad \mathbf{Pr}^{\max}_{\mathcal{M}_2}(\Diamond\,\text{target}) = 1/8 \qquad \triangle$$

The crucial property of subsystems is that their maximal and minimal reachability probabilities do not increase when compared with the original system. For minimal probabilities point (2.) of Definition 4.1 is important, as it makes sure that all actions remain available for schedulers to choose. It is important that the considered MDP is in reachability form for the following proposition, as it relies on the fact that paths which have reached "exit" can no longer reach "target" thereafter.

Proposition 4.4. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and $\mathcal{M}'$ be a subsystem of $\mathcal{M}$ with states $S' \cup \{\text{target}, \text{exit}\}$. Then for all states $s \in S'$ we have*

$$\mathbf{Pr}^{\min}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond\,\text{target}) \quad \textit{and} \quad \mathbf{Pr}^{\max}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond\,\text{target}).$$

Proof. For both statements we use the fact that the finite paths of $\mathcal{M}'$ which have not yet visited exit form a subset of the corresponding paths in $\mathcal{M}$. More precisely:

$$\{\pi \in \text{Paths}_{\text{fin}}(\mathcal{M}') \mid \text{last}(\pi) \neq \text{exit}\} \subseteq \{\pi \in \text{Paths}_{\text{fin}}(\mathcal{M}) \mid \text{last}(\pi) \neq \text{exit}\}.$$

To prove $\mathbf{Pr}^{\min}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \mathbf{Pr}^{\min}_{\mathcal{M},s}(\Diamond\,\text{target})$ we show that for any scheduler $\mathfrak{S}$ for $\mathcal{M}$ we find a scheduler $\mathfrak{S}'$ for $\mathcal{M}'$ such that $\Pr^{\mathfrak{S}'}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\,\text{target})$. We define $\mathfrak{S}'$ to choose the same action as chosen by $\mathfrak{S}$ for all paths which have not yet visited exit. This is possible, because every action which is enabled in a state in $\mathcal{M}$ is also enabled in the same state in $\mathcal{M}'$ (point (2.) of Definition 4.1). Then any finite $\mathfrak{S}'$-path in $\mathcal{M}'$ from s that visits target is also an $\mathfrak{S}$-path in $\mathcal{M}$ carrying the same probability, which shows that $\Pr^{\mathfrak{S}'}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\,\text{target})$ holds.

To prove $\mathbf{Pr}^{\max}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Diamond\,\text{target})$ we show that for any scheduler $\mathfrak{S}'$ for $\mathcal{M}'$ we find a scheduler $\mathfrak{S}$ for $\mathcal{M}$ such that $\Pr^{\mathfrak{S}'}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\,\text{target})$. We define $\mathfrak{S}$ to choose the same action as $\mathfrak{S}'$ for all paths which are also $\mathfrak{S}'$-paths in $\mathcal{M}'$, and arbitrary otherwise. Then $\Pr^{\mathfrak{S}'}_{\mathcal{M}',s}(\Diamond\,\text{target}) \leq \Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond\,\text{target})$ follows as in the other case. □

Due to the above property we may call a subsystem a *witness* for a lower-bounded probabilistic reachability constraint if it already satisfies the bound, as the existence of such a witness implies that the original MDP satisfies the constraint.

Definition 4.5 (Witnessing subsystems). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, $\mathcal{M}'$ be a subsystem of $\mathcal{M}$, $\lambda \in [0, 1]$ and $\gtrsim \in \{\geq, >\}$. We say that

- $\mathcal{M}'$ is a *witness* for $\mathbf{Pr}^{\min}_{\mathcal{M},s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ if $\mathbf{Pr}^{\min}_{\mathcal{M}',s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ holds, and
- $\mathcal{M}'$ is a *witness* for $\mathbf{Pr}^{\max}_{\mathcal{M},s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ if $\mathbf{Pr}^{\max}_{\mathcal{M}',s_{in}}(\Diamond\,\text{target}) \gtrsim \lambda$ holds.

Remark 4.6 (Witnesses for upper-bounded threshold properties). Witnessing subsystems are only defined above for lower-bounded probabilistic reachability constraints. For upper-bounds, one can use a dual definition in which transitions are redirected to target. This guarantees that the optimal probability of reaching "target" in subsystem can never decrease. As the

problem of computing witnessing subsystems is different in this case, we consider it separately in Section 4.4. However, for EC-free MDPs (and therefore also for Markov chains) the problem of computing witnesses for upper-bounded thresholds can be reduced to the lower-bounded case by changing the meaning of "target" and "exit" and using the equations

$$\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) = 1 - \mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{exit}) \quad \text{and} \quad \mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) = 1 - \mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{exit}),$$

which hold for EC-free MDPs $\mathcal{M}$ in reachability form. △

Remark 4.7 (Purely probabilistic and nondeterministic systems). If $\mathcal{M}$ is a Markov chain, then $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target})$ holds and hence witnesses for the two types of properties defined in Definition 4.5 coincide. Now consider the case that $\mathcal{M}$ is purely nondeterministic (i.e., it is a transition system). This means that for all states s and enabled actions $\alpha \in \text{Act}(s)$ there exists a unique successor state s' satisfying $P(s, \alpha, s') = 1$. Then, all subsystems $\mathcal{M}'$ of $\mathcal{M}$ satisfy $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}'}(\Diamond \text{target}) \in \{0, 1\}$ (with $\mathfrak{m} \in \{\min, \max\}$). We have $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) = 1$ iff "exit" is not reachable from s_{in} in $\mathcal{M}$, and for all subsystems $\mathcal{M}'$ of $\mathcal{M}$, apart from $\mathcal{M}$ itself, we have $\mathbf{Pr}^{\min}_{\mathcal{M}'}(\Diamond \text{target}) = 0$. In other words, the only potential witness for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq 1$ is $\mathcal{M}$ itself. On the other hand, a subsystem $\mathcal{M}'$ of $\mathcal{M}$ is a witness for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 1$ iff "target" is reachable from s_{in} in $\mathcal{M}'$. Hence, all paths from s_{in} to "target" in $\mathcal{M}$ induce witnesses for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 1$. △

Example 4.8. Consider again the MDPs $\mathcal{M}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ as shown in Figure 4.1. The subsystem $\mathcal{M}_1$ is a witness for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 9/20$, but does not witness any non trivial lower bound on the minimal reachability probability (as we have $\mathbf{Pr}^{\min}_{\mathcal{M}_1}(\Diamond \text{target}) = 0$). On the other hand, $\mathcal{M}_2$ is a witness for both $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq 1/8$ and $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq 1/8$. △

As the minimal probability to reach "target" is zero in all states which are part of some proper end component, these states can always be removed when considering witnessing subsystem for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$. This motivates the following definition.

Definition 4.9 (Largest min-relevant subsystem). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and S_R its min-relevant states (see Definition 3.21). We call $\mathcal{M}^* = \mathcal{M}_{S_R}$, i.e., the subsystem of $\mathcal{M}$ induced by S_R, the *largest min-relevant subsystem* of $\mathcal{M}$.

One can check that $\mathcal{M}^*$ is identical to the MDP with the same name defined in the proof of Theorem 3.24. There, it was used to argue that the existence of a Farkas certificate for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$ indeed proves that the constraint holds in $\mathcal{M}$. As argued there, $\mathbf{A}^*$ and $\mathbf{t}^*$, which were defined in Definition 3.21 as the restrictions of $\mathbf{A}, \mathbf{t}$ to states in S_R, are identical to the system matrix and target vector of $\mathcal{M}^*$. The following proposition shows that when looking for witnesses for lower bounds on the minimal reachability probability, it is enough to consider subsystems of $\mathcal{M}^*$.

Proposition 4.10. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$, $\mathcal{M}^*$ be its largest min-relevant subsystem and $\lambda \in [0, 1]$. Then, for all subsystems $\mathcal{M}'$ of $\mathcal{M}$ satisfying $\mathbf{Pr}^{\min}_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$, there exists a subsystem $\mathcal{M}''$ of both $\mathcal{M}^*$ and $\mathcal{M}'$ which satisfies $\mathbf{Pr}^{\min}_{\mathcal{M}''}(\Diamond \text{target}) \geq \lambda$.*

Proof. Let $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ be a subsystem of $\mathcal{M}$ such that $\mathbf{Pr}^{\min}_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$ holds and let $\mathcal{M}''$ be the subsystem one gets by taking $\mathcal{M}'$ and redirecting all incoming transitions to states in $S \setminus S_R$ to "exit". As we have $\mathbf{Pr}^{\min}_{\mathcal{M}',s}(\Diamond \text{target}) = 0$ for all $s \in S \setminus S_R$ it follows that $\mathbf{Pr}^{\min}_{\mathcal{M}''}(\Diamond \text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$. The MDP $\mathcal{M}''$ is a subsystem of both $\mathcal{M}^*$ and $\mathcal{M}'$, which proves the claim. □

4.1.1 The witness problem

Any MDP $\mathcal{M}$ in reachability form is a subsystem of itself, and hence it is also a witnessing subsystem for any lower-bounded probabilistic reachability constraint which is satisfied by $\mathcal{M}$. But the entire system $\mathcal{M}$ is clearly not very informative as a witness. Rather, useful witnessing subsystems should highlight a restricted part of the system which by itself carries enough probability to satisfy the given constraint. Hence the important computational problem in this context is to compute small (or minimal) witnessing subsystems.

Definition 4.11 (Witness problem). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, $\lambda \in [0,1] \cap \mathbb{Q}$ and $k \in \mathbb{N}$, both encoded in binary.

- The *min-witness problem* asks whether a subsystem $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ of $\mathcal{M}$ exists such that $\mathbf{Pr}^{\min}_{\mathcal{M}', s_{in}}(\Diamond\, \text{target}) \geq \lambda$ and $|S'| \leq k$.
- The *max-witness problem* asks whether a subsystem $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ of $\mathcal{M}$ exists such that $\mathbf{Pr}^{\max}_{\mathcal{M}', s_{in}}(\Diamond\, \text{target}) \geq \lambda$ and $|S'| \leq k$.

A polynomial time algorithm for the witness problem directly yields a polynomial time algorithm to compute a minimal witnessing subsystem. If $\mathcal{M}$ is a Markov chain the two problems coincide as $\mathbf{Pr}^{\min}_{\mathcal{M}, s_{in}}(\Diamond\, \text{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}, s_{in}}(\Diamond\, \text{target})$ holds.

Remark 4.12 (Handling multiple target states). In many cases it is useful to consider a set T of target-states, rather than a single state "target". One way to get an equivalent MDP in reachability form is to collapse the states included in T into a single state. However, as a result one would only consider subsystems in which all states of T are included, which may not always be desired. To reason about subsystems in which T may be partially included one can transform the MDP into reachability form by adding a fresh state "target" and adapting all states in T such that they have only one transition to "target", carrying probability one. △

Notions of size for subsystems

The witness problem as defined above implicitly considers a specific notion of *size* for subsystems, namely the number of states that the subsystem includes. However, depending on the situation, other notions might be useful. We first consider two generalizations of the witness problem where one is allowed to specify either a weight function, or a labeling function, which both induce a specific notion of size for subsystems. Labels can be used to group states which, for example, belong to the same component of a compositional system, or to the same statement in a high-level description of the model. The weighted version of the problem associates with each state a natural number which represents the cost of including that state in a subsystem.

A *weight function* for an MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ is a function $wgt : S \to \mathbb{N}$. The weight of a subsystem $\mathcal{M}'$ of $\mathcal{M}$ with states $S' \cup \{\text{target}, \text{exit}\}$ is defined to be $wgt(\mathcal{M}') = \sum_{s \in S'} wgt(s)$. A *labeling function* for $\mathcal{M}$ into a finite set of labels L is of the form $\Lambda : S \to 2^L$, and, for a given subsystem $\mathcal{M}'$ of $\mathcal{M}$ with states $S' \cup \{\text{target}, \text{exit}\}$, we define the set of labels that $\mathcal{M}'$ hits as $\Lambda(\mathcal{M}') = \{l \in L \mid \text{there exists } s \in S' \text{ such that } l \in \Lambda(s)\}$.

Definition 4.13 (label-based and weighted versions). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, $\lambda \in [0,1] \cap \mathbb{Q}$ and $k \in \mathbb{N}$. Furthermore, let wgt be a weight function for $\mathcal{M}$ and Λ be a labeling function for $\mathcal{M}$ into the set of labels L.

- The *label-based max-witness problem* asks whether a subsystem $\mathcal{M}'$ of $\mathcal{M}$ with states $S' \cup \{\text{target}, \text{exit}\}$ exists such that $\mathbf{Pr}^{\max}_{\mathcal{M}', s_{in}}(\Diamond\, \text{target}) \geq \lambda$ and $|\Lambda(\mathcal{M}')| \leq k$ hold.
- The *weighted max-witness problem* asks whether a subsystem $\mathcal{M}'$ of $\mathcal{M}$ with states $S' \cup \{\text{target}, \text{exit}\}$ exists such that $\mathbf{Pr}^{\max}_{\mathcal{M}', s_{in}}(\Diamond\, \text{target}) \geq \lambda$ and $wgt(\mathcal{M}') \leq k$ hold.

The *min*-versions of the two problems are defined analogously by replacing $\mathbf{Pr}^{\max}$ by $\mathbf{Pr}^{\min}$.

The above definitions all take a *state-based* view. Other natural ways of measuring the size of subsystems include counting the transitions included in the subsystem rather than the states, or counting both transitions and states. The transitions of an MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ are defined as $T_{\mathcal{M}} = \{(s, \alpha, u) \in S \times \text{Act} \times S \mid P(s, \alpha, u) > 0\}$. We do not consider transitions in which states "target" and "exit" participate. If one wants to count transitions of "target", one can first apply a reduction as described in Remark 4.12.

Both the weighted and labeled witness problems can be adapted canonically to take the transition-based (or combined) view, simply by considering weight functions (respectively the labeling functions) with domain $T_{\mathcal{M}}$ or $T_{\mathcal{M}} \cup S$. Such weight- or labeling functions are called *transition-based* (respectively *combined*). We now show that the resulting problems can be reduced in polynomial time to the corresponding state-based versions. The combined view generalizes both state-based and transition-based views, as states, respectively transitions, can be zero-weighted or have no labels. Hence it is enough to provide a reduction from the combined setting to the state-based setting. Using it, any algorithm for the state-based problem can be transferred to solve the corresponding problem under the combined or transition-based view.

Proposition 4.14. *The label-based and weighted witness problems under the combined view can be reduced in polynomial time to the corresponding state-based witness problems.*

Proof. Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP and wgt, Λ be combined weight and labeling functions for $\mathcal{M}$. The proof proceeds by constructing an MDP $\mathcal{M}_s$ together with state-based weight and labeling functions wgt_s, Λ_s for $\mathcal{M}_s$ such that the subsystems of $\mathcal{M}$ and the state-induced subsystems of $\mathcal{M}_s$ are in a one-to-one correspondence which preserves probabilities, weights and labels. We will use the notation $S_{\text{all}} = S \cup \{\text{target}, \text{exit}\}$, $T_{\text{all}} = \{(s, \alpha, t) \in S_{\text{all}} \times \text{Act} \times S_{\text{all}} \mid P(s, \alpha, t) > 0\}$ and $T = T_{\mathcal{M}}$ as defined above. Observe that T_{all} includes transitions with end-points in $\{\text{target}, \text{exit}\}$, which are not included in T, and recall that $\mathcal{E} \subseteq S \times \text{Act}$ are the enabled state-action pairs of $\mathcal{M}$.

Consider the MDP $\mathcal{M}_s = (S \cup T_{\text{all}} \cup \{\text{target}, \text{exit}\}, \text{Act}', s_{in}, P')$ whose probabilistic transition function P' is defined such that the states "target" and "exit" are absorbing, and additionally including the following transitions:

$$s \xrightarrow{\alpha} (s, \alpha, u) \text{ with probability } P(s, \alpha, u) \qquad \text{for all } (s, \alpha) \in \mathcal{E},$$
$$(s, \alpha, u) \xrightarrow{\alpha} u \text{ with probability } 1 \qquad \text{for all } (s, \alpha, u) \in T_{\text{all}}.$$

Furthermore, we define $wgt_s(x) = wgt_s(x)$ and $\Lambda_s(x) = \Lambda(x)$ for all $x \in T \cup S$, and $wgt_s(x) = 0$ and $\Lambda_s(x) = \varnothing$ otherwise. Note that in $\mathcal{M}_s$, wgt_s and Λ_s are indeed state-based weight, respectively labeling, functions. There is a bijection between paths in $\mathcal{M}$ and $\mathcal{M}_s$ given by

$$s_0\alpha_0 s_1\alpha_1 s_2 \ldots \text{ corresponds to } s_0\alpha_0(s_0, \alpha_0, s_1)\alpha_0 s_1\alpha_1(s_1, \alpha_1, s_2)\alpha_1 s_2 \ldots$$

and this bijection preserves probabilities. This immediately implies that a subsystem $\mathcal{M}'$ of $\mathcal{M}$ including exactly states $S' \subseteq S_{\text{all}}$ and transitions $T' \subseteq T_{\text{all}}$ is a witness if and only if the

subsystem $\mathcal{M}'_s$ of $\mathcal{M}_s$ induced by states $S' \cup T'$ is a witness. Also it follows directly that $wgt_s(\mathcal{M}'_s) = wgt(\mathcal{M}')$ and $\Lambda_s(\mathcal{M}'_s) = \Lambda(\mathcal{M}')$, which concludes the proof. □

4.1.2 Complexity of the witness problem

All variants of the witness problem that we have introduced are included in NP. This has been observed for the versions of the problems which have been studied previously in the literature, namely the max-versions of the standard [WJÁ+12] and label-based [KÁJW15] witness problems. The proof for the other cases is exactly the same, however: one can guess a subset of the states of the given MDP and check (1) whether the size of the induced subsystem (which is either the combined weight of the participating states, or the number of labels that appear) is at most k and (2) whether the subsystem is a witness, i.e., whether it satisfies the corresponding probabilistic reachability constraint. Both checks can be done in polynomial time.

Proposition 4.15 ([WJÁ+12, KÁJW15]). *The weighted and label-based max- and min-witness problems are in NP.*

In sections 4.2.1 and 4.2.3 we will discuss practical approaches to solve the optimization versions of these problems, i.e., for computing minimal witnesses with respect to the different size measures. Now we turn to the question of whether the problems are NP-hard. It is known that the max-witness problem is NP-hard (see [WJÁ+14, Theorem 7], whose proof is based on a result in [CV10]). However, the precise complexity for the restricted case of Markov chains was open so far, as noted in [WJÁ+14, Jan15]. We now show that the problem remains NP-hard even for acyclic Markov chains. In particular, this shows that the min-witness problem is NP-hard for MDPs. The proof goes by reduction from the clique problem which asks, given an undirected graph G and a natural number C, whether G contains a complete (i.e., fully connected) subgraph with C vertices [Kar72].

Theorem 4.16. *The witness problem for acyclic Markov chains is NP-hard.*

Proof. Let an instance of the clique problem be given by the undirected graph $G = (V, E)$ and the natural number C. We will assume that $C \geq 3$ holds and that no vertex has a self loop. Consider the Markov chain $\mathcal{M} = (V \cup E \cup \{s_{in}, \text{target}, \text{exit}\}, s_{in}, P)$ containing a state for each vertex and edge of G, and additional states $\{s_{in}, \text{target}, \text{exit}\}$, where s_{in} is the initial state. The probability transition function P is defined as follows, where $n = |V|$:

- $P(s_{in}, v) = 1/n$ for all $v \in V$,
- $P(v, \{v, w\}) = 1/n$ for all $v \in V$ and edges $\{v, w\} \in E$ in which v participates,
- $P(\{v, w\}, \text{target}) = 1$ for all $\{v, w\} \in E$,
- the remaining probability in each state is added to a transition to exit.

So each state of $\mathcal{M}$ corresponding to a vertex of G has as many outgoing transitions as it has neighbors in G, and each state corresponding to an edge has exactly two incoming transitions. A sketch of the construction can be seen in Figure 4.2.

Let $\lambda = \frac{C(C-1)}{n^2}$ and $k = C + \frac{C(C-1)}{2} + 1$. We claim that there exists a subsystem $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, s_{in}, P')$ of $\mathcal{M}$ such that $\Pr_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$ and $|S'| \leq k$ if and only if G has a clique of size C.

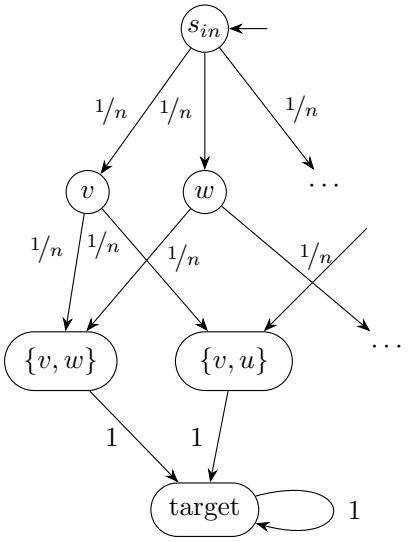


Figure 4.2: A sketch of the reduction from the clique problem to the witness problem for acyclic Markov chains. The state "exit" together with transitions to it are omitted.

"$\Longleftarrow$": Let $V' \subseteq V$ be a set of vertices which form a clique in G such that $|V'| = C$. Let $E' = \{\{u, v\} \mid u, v \in V'\} \subseteq E$ be the edges between vertices in V' and consider the subsystem $\mathcal{M}'$ of $\mathcal{M}$ induced by $V' \cup E' \cup \{s_{in}\}$. We have $|V'| = C$ and $|E'| = \frac{C(C-1)}{2}$, and hence the number of states of $\mathcal{M}'$ is exactly k. For each state $\{u, v\} \in E'$ there exist two paths in $\mathcal{M}'$ from s_{in} to target visiting $\{u, v\}$. As each such path has probability $1/n^2$ we can conclude that $\Pr_{\mathcal{M}'}(\Diamond \text{target}) = \frac{C(C-1)}{n^2} = \lambda$.

"$\Longrightarrow$": Let $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, s_{in}, P')$ be a subsystem of $\mathcal{M}$ such that $\Pr_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$ and $|S'| \leq k$. Consider the partitioning $S' = V' \cup E' \cup \{s_{in}\}$ of S' such that $V' \subseteq V$ and $E' \subseteq E$, and define $a = |V'|$ and $b = |E'|$. We have: $a + b \leq k - 1 = C + \frac{C(C-1)}{2}$. Let us denote by T the number of transitions between sets V' and E' that exist in the subsystem $\mathcal{M}'$. As $\Pr_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda = \frac{C(C-1)}{n^2}$ there must be at least $C(C-1)$ such transitions, i.e., $T \geq C(C-1)$. Since each state in E' has at most two incoming transitions from states in V' we have $2b \geq T \geq C(C-1)$. By invoking $a + b \leq C + \frac{C(C-1)}{2}$ it now follows that $a \leq C$ must hold.

Now let us partition E' into those states that have exactly one, respectively two, incoming transition from V', called E'_1 and E'_2. The set E'_2 is bounded from above by $\frac{a(a-1)}{2}$, which is the possible number of edges with both endpoints in V'. We have:

$$\begin{aligned} T &= 2E'_2 + E'_1 \leq 2 \cdot \frac{a(a-1)}{2} + b - \frac{a(a-1)}{2} \\ &= \frac{a(a-1)}{2} + b \overset{\text{(I.)}}{\leq} \frac{a(a-1)}{2} + C + \frac{C(C-1)}{2} - a \\ = \frac{a(a-3)}{2} + \frac{C(C+1)}{2} &\overset{\text{(II.)}}{\leq} \frac{C(C-3)}{2} + \frac{C(C+1)}{2} = C(C-1) \leq T \end{aligned}$$

Inequality (I.) uses $a + b \leq C + \frac{C(C-1)}{2}$, and for (II.) we use $a \leq C$ and our assumption that $C \geq 3$ holds. In summary, we have equality for all the above expressions. By rewriting the equality between the left and right hand side of the inequality (II.) we get $a(a-3) = C(C-3)$, which, given that $C \geq 3$ and $a \geq 0$ hold, implies $a = C$. It follows that $E'_2 = b = \frac{C(C-1)}{2}$ and hence all states in E' have two incoming transitions from V'. This implies that V' forms a clique in G of size C, which concludes the proof. □

Remark 4.17 (Complexity for fixed λ). The above proof can be adapted to the situation where $\lambda \in (0,1)$ is fixed and not part of the input. To this end choose $n \geq |V|$ such that

$$\lambda \geq \frac{C(C-1)}{n^2} \qquad \text{and} \qquad \lambda - \frac{C(C-1)}{n^2} + \frac{|V|}{n} \leq 1.$$

Then, add a transition to the constructed Markov chain which goes directly from s_{in} to "target" carrying probability $\lambda - \frac{C(C-1)}{n^2}$. The other parts of the construction and proof remain the same, after substituting λ in the proof by $\lambda' = \frac{C(C-1)}{n^2}$. Here $\lambda - \frac{C(C-1)}{n^2} + \frac{|V|}{n} \leq 1$ ensures that the construction yields a legal Markov chain (any superfluous probability is redirected to exit). Such an n can be found as we have assumed $\lambda < 1$. △

Remark 4.18 (Strong NP-hardness). From the proof of Theorem 4.16 it also follows that the witness problem is NP-complete in the strong sense, that is, even if all numbers are encoded in unary. This is because $\lambda = \frac{C(C-1)}{n^2}$ and all transition probabilities (which are either $1/n$ or 1) are polynomial in the size of the clique instance, even if encoded in unary. △

4.1.3 The core-problem for Markov chains

The *core* of a Markov decision process is a concept introduced in [KM20] which is similar to the notion of a subsystem. A major difference, however, is that cores are defined independently of any property and require only that the maximal probability of ever leaving the core is small. On the other hand, witnessing subsystems are defined relative to some fixed reachability query.

Definition 4.19 ([KM20]). Let $\mathcal{M} = (S, \mathrm{Act}, s_{in}, P)$ be an MDP and $\epsilon \in \mathbb{Q}_{>0}$. A set $S_\epsilon \subseteq S$ is an *ϵ-core* of $\mathcal{M}$ if $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond(S \setminus S_\epsilon)) < \epsilon$.

We will call the problem of deciding whether an ϵ-core with at most k states exists the *core-problem*, where k and ϵ are part of the input. This problem is NP-complete in general [KM20, Theorem 3.6], but NP-hardness was open for the restricted class of Markov chains [KM20, Remark 3.7]. We now show that the witness problem for Markov chains can be reduced to the core-problem for Markov chains. This, together with Theorem 4.16, implies that the core-problem is NP-hard for Markov chains.

Proposition 4.20. *The core-problem for Markov chains is NP-hard.*

Proof. Consider an instance of the witness problem for Markov chains, given by a Markov chain $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ in reachability form, a natural number k and a $\lambda \in (0,1] \cap \mathbb{Q}$. The problem asks whether a witnessing subsystem for $\Pr_{\mathcal{M}}(\Diamond \text{ target}) \geq \lambda$ exists with at most k states, and we can assume that $k \leq |S|$. We describe a polynomial reduction from this problem to the core-problem for Markov chains.

Let $\mathcal{M}' = (S \cup \{\text{target}, \text{exit}\} \cup L, s_{in}, P')$ be a Markov chain which is constructed by adding a self loop involving $|S| + 1$ fresh states to the state "exit" of $\mathcal{M}$. Formally, let $L = \{l_0, \ldots, l_{|S|}\}$ and define:

- $P'(s,u) = P(s,u)$ if $s, u \in S$,
- $P(\text{exit}, l_0) = 1$, $P(l_i, l_{i+1}) = 1$ for all $0 \leq i < |S|$, and $P(l_{|S|}, \text{exit}) = 1$.

We claim that $\mathcal{M}$ has a λ-core of size at most $k+1$ if and only if $\mathcal{M}$ has a witnessing subsystem for $\Pr_{\mathcal{M}}(\Diamond \text{ target}) \geq \lambda$ of size at most k. The one state difference stems from the fact that target does not count towards the size of a witnessing subsystem, by convention.

If $\mathcal{N} = (S' \cup \{\text{target}, \text{exit}\}, s_{in}, P_{\mathcal{N}})$ is a subsystem of $\mathcal{M}$ and satisfies $\Pr_{\mathcal{N}}(\Diamond \text{target}) \geq \lambda$, then clearly $S' \cup \{\text{target}\}$ is a λ-core of $\mathcal{M}'$ and has exactly $|S'|+1$ states. For the other direction, we first observe that we may assume that any given λ-core of $\mathcal{M}'$ includes either all states in $L \cup \{\text{exit}\}$, or none of them. This is because if the set is partially included in the core, then each of the included states has probability one of leaving the core, and can hence be removed. So let S_λ be a λ-core of $\mathcal{M}'$ with this property and such that $|S_\lambda| \leq k+1$. As $\mathcal{M}$ is in reachability form, the only bottom strongly connected components of $\mathcal{M}'$ are induced by $\{\text{target}\}$ and $\{\text{exit}\} \cup L$. Hence we have $\{\text{target}\} \subseteq S_\lambda$ or $\{\text{exit}\} \cup L \subseteq S_\lambda$. The latter can be excluded as $|S_\lambda| \leq k+1 \leq |S|+1$ and $|L| = |S|+1$. Hence, we have $\text{target} \in S_\lambda$ and $\text{exit} \notin S_\lambda$ which implies, by the core-property, that the subsystem induced by S_λ in $\mathcal{M}$ is a witness for $\Pr_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$. □

Remark 4.21. As the witness problem remains NP-hard in acyclic Markov chains for fixed λ (see Remark 4.17), the core-problem remains NP-hard for fixed ϵ using the same proof. △

4.2 Farkas certificates and witnessing subsystems

Farkas certificates were introduced in Chapter 3 as a means to certify the result of model checking algorithms for probabilistic reachability constraints. As such they are tokens which can be used to derive a simple proof showing that the given property holds in an MDP. However, at first sight it is not clear whether they contain any information that can be used to intuitively explain *why* the constraint is satisfied.

This section establishes a connection between Farkas certificates and witnessing subsystems for lower-bounded probabilistic reachability constraints. The correspondence relates zero-valued entries in a Farkas certificate to states (or state-action pairs) which can be removed to form a witnessing subsystem. This allows us to derive a witnessing subsystem from a Farkas certificate, and vice versa. Furthermore, we use this connection to develop new algorithms to compute minimal or small witnessing subsystems. All these algorithms are based on finding *small* (i.e., with few non-zero entries) Farkas certificates for the corresponding property, and rely on (mixed-integer) linear programming techniques.

We first show a technical lemma which relates solutions of the linear inequalities defining Farkas certificates to solutions of "reduced" systems of linear inequalities, in which some rows and columns in the matrix representation are omitted. Such reduced systems are related to the corresponding inequalities of *subsystems* of $\mathcal{M}$, as was pointed out directly following Definition 4.2. For a given vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$, where $\mathcal{E}$ is the set of enabled state-action pairs of an MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$, we define the *state support* of $\mathbf{y}$ as the set of states such that at least one action has non-zero value in $\mathbf{y}$. More formally, we have $\text{state-supp}(\mathbf{y}) = \{s \in S \mid \sum_{\alpha \in \text{Act}(s)} \mathbf{y}(s,\alpha) > 0\}$.

Lemma 4.22. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP, $\mathcal{E}$ its enabled state-action pairs and $\mathbf{A}$ its system matrix. Fix $\mathbf{b} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ and $\lambda \geq 0$.*

1. *Let $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$, $S' \subseteq S$ be a set of states such that $\text{supp}(\mathbf{z}) \subseteq S'$ and $\mathcal{E}' = \{(s,\alpha) \mid s \in S', \alpha \in \text{Act}(s)\}$. We assume that $s_{in} \in S'$. Furthermore, let $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$, $\mathbf{b}' = \mathbf{b}|_{\mathcal{E}'}$ and $\mathbf{z}' = \mathbf{z}|_{S'}$ be the corresponding restrictions. Then:*

$$\mathbf{A}\mathbf{z} \leq \mathbf{b} \wedge \mathbf{z}(s_{in}) \geq \lambda \text{ holds} \quad \text{if and only if} \quad \mathbf{A}'\mathbf{z}' \leq \mathbf{b}' \wedge \mathbf{z}'(s_{in}) \geq \lambda \text{ holds.}$$

2. *Let* $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$, $S' \subseteq S$ *be such that* state-supp$(\mathbf{y}) \subseteq S'$ *and* $\mathcal{E}' = \{(s,\alpha) \mid s \in S', \alpha \in \mathrm{Act}(s)\}$. *Furthermore, let* $\mathbf{y}' = \mathbf{y}|_{\mathcal{E}'}$, $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$ *and* $\mathbf{b}' = \mathbf{b}|_{\mathcal{E}'}$. *Then:*

$$\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\,\mathbf{b} \geq \lambda \text{ holds} \quad \text{if and only if} \quad \mathbf{y}'\mathbf{A}' \leq \delta_{s_{in}} \wedge \mathbf{y}'\,\mathbf{b}' \geq \lambda \text{ holds.}$$

Proof. (1.) As $s_{in} \in S'$ and by definition of $\mathbf{z}'$ we directly get $\mathbf{z}(s_{in}) \geq \lambda$ if and only if $\mathbf{z}'(s_{in}) \geq \lambda$. The set S' contains all states $s \in S$ such that $\mathbf{z}(s) > 0$, and hence we have for all $(s,\alpha) \in \mathcal{E}$:

$$\underbrace{\mathbf{b}(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot \mathbf{z}(s')}_{R(s,\alpha)} \;=\; \underbrace{\mathbf{b}(s,\alpha) + \sum_{s' \in S'} P(s,\alpha,s') \cdot \mathbf{z}(s')}_{R'(s,\alpha)}.$$

By construction, $\mathbf{A}\mathbf{z} \leq \mathbf{b}$ is equivalent to $\mathbf{z}(s) \leq R(s,\alpha)$ for each $(s,\alpha) \in \mathcal{E}$, and $\mathbf{A}'\mathbf{z}' \leq \mathbf{b}'$ is equivalent to $\mathbf{z}'(s) \leq R'(s,\alpha)$ for each $(s,\alpha) \in \mathcal{E}'$. Hence, the direction from left to right follows immediately from the above equality. For the other direction, we additionally observe that $\mathbf{z}(s) = 0$ for all $s \in S \setminus S'$ and hence $\mathbf{z}(s) \leq R(s,\alpha)$ trivially holds for all $(s,\alpha) \in \mathcal{E}$ with $s \in S \setminus S'$.

(2.) As each non-zero entry of $\mathbf{y}$ is also an entry of $\mathbf{y}'$, we directly get $\mathbf{y}\,\mathbf{b} = \mathbf{y}'\,\mathbf{b}'$. By similar reasoning as above it follows that for all $s \in S$ we have:

$$\underbrace{\delta_{s_{in}}(s) + \sum_{(s',\alpha') \in \mathcal{E}} P(s',\alpha',s) \cdot \mathbf{y}(s',\alpha')}_{R(s)} \;=\; \underbrace{\delta_{s_{in}}(s) + \sum_{(s',\alpha') \in \mathcal{E}'} P(s',\alpha',s) \cdot \mathbf{y}(s',\alpha')}_{R'(s)}.$$

By construction $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}}$ amounts to requiring $\sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s,\alpha) \leq R(s)$ for all $s \in S$, and $\mathbf{y}'\mathbf{A}' \leq \delta_{s_{in}}$ is equivalent to $\sum_{\alpha \in \mathrm{Act}(s)} \mathbf{y}(s,\alpha) \leq R'(s)$ for all $s \in S'$. Now the claim follows in the same way as above. □

Now we are in a position to state the main theorem of the section, which says that the existence of a Farkas certificate whose support is included in some subset of states S' implies that the induced subsystem $\mathcal{M}_{S'}$ is a witness for the corresponding property, and vice versa. This insight will be used throughout this chapter to devise novel algorithms to compute small witnessing subsystems by searching for Farkas certificates with small support.

Theorem 4.23. *Let* $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ *be an MDP in reachability form,* $\lambda \in [0,1]$ *and* $S' \subseteq S$. *Then:*

1. *There exists* $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ *such that* $\mathrm{supp}(\mathbf{z}) \subseteq S'$ *if and only if* $\mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond\,\text{target}) \geq \lambda$ *holds.*
2. *There exists* $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ *such that* state-supp$(\mathbf{y}) \subseteq S'$ *if and only if* $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Diamond\,\text{target}) \geq \lambda$ *holds.*

Proof. Let S_R be the min-relevant states of $\mathcal{M}$ and $\mathcal{E}' = \{(s,\alpha) \in \mathcal{E} \mid s \in S'\}$.

(1.) For this part of the proof, let us fix $S'' = S' \cap S_R$ and $\mathcal{E}'' = \{(s,\alpha) \in \mathcal{E} \mid s \in S''\}$.

"$\Longrightarrow$". From $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ we get by definition that $\mathbf{A}^*\mathbf{z} \leq \mathbf{t}^* \wedge \mathbf{z}(s_{in}) \geq \lambda$ holds (see Definition 3.23). We have already observed that $\mathbf{A}^*$ is the system matrix of the largest min-relevant subsystem $\mathcal{M}^*$ of $\mathcal{M}$ (see Definition 4.9), and $\mathbf{t}^*$ is its target vector. If we take $\mathbf{A}' = \mathbf{A}^*|_{\mathcal{E}'' \times S''}$, $\mathbf{t}' = \mathbf{t}^*|_{S''}$ and $\mathbf{z}' = \mathbf{z}|_{S''}$ we may conclude that $\mathbf{A}'\mathbf{z}' \leq \mathbf{t}'$ holds by point (1.) of Lemma 4.22. We also know that $\mathbf{A}'$ is the system matrix of the induced subsystem $\mathcal{M}_{S''}$, and $\mathbf{t}'$ is its target vector. From $S'' \subseteq S_R$ it follows that $\mathcal{M}_{S''}$ is EC-free. Hence, from $\mathbf{A}'\mathbf{z}' \leq \mathbf{t}'$ and $\mathbf{z}'(s_{in}) = \mathbf{z}(s_{in}) \geq \lambda$

we may conclude that $\mathbf{z}'$ is a Farkas certificate for the corresponding property in $\mathcal{M}_{S''}$ (i.e., $\mathbf{z}' \in \mathcal{F}^{\min}_{\mathcal{M}_{S''},\geq}(\lambda)$) and hence $\mathbf{Pr}^{\min}_{\mathcal{M}_{S''}}(\lozenge \text{target}) \geq \lambda$ holds by Theorem 3.24. As $S'' \subseteq S'$ holds, $\mathcal{M}_{S''}$ is a subsystem of $\mathcal{M}_{S'}$ and hence we also have $\mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\lozenge \text{target}) \geq \lambda$ by Proposition 4.4.

"$\Longleftarrow$". As states in $S' \setminus S''$ have minimal probability zero of reaching target in all subsystems of $\mathcal{M}$, we have $\mathbf{Pr}^{\min}_{\mathcal{M}_{S''}}(\lozenge \text{target}) \geq \lambda$. Hence we find a Farkas certificate $\mathbf{z}' \in \mathcal{F}^{\min}_{\mathcal{M}_{S''},\geq}(\lambda)$, which satisfies $\mathbf{A}'\mathbf{z}' \leq \mathbf{t}'$ and $\mathbf{z}'(s_{in}) \geq \lambda$, where $\mathbf{A}', \mathbf{t}'$ are the system matrix and target vector of $\mathcal{M}_{S''}$. At the same time, $\mathbf{A}', \mathbf{t}'$ are the restrictions of $\mathbf{A}^*, \mathbf{t}^*$ to domains $\mathcal{E}'' \times S''$ and S'' respectively. Then it follows by Lemma 4.22 that the vector $\mathbf{z} \in \mathbb{R}^{S_R}_{\geq 0}$ one gets by setting the missing entries in $\mathbf{z}'$ to zero satisfies $\mathbf{A}^*\mathbf{z} \leq \mathbf{t}^*$ and $\mathbf{z}(s_{in}) = \mathbf{z}'(s_{in}) \geq \lambda$. But this shows that $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ holds.

(2.) Let $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$ and $\mathbf{t}' = \mathbf{t}|_{S'}$. As we have seen, $\mathbf{A}'$ and $\mathbf{t}'$ are the system matrix and target vector of MDP $\mathcal{M}_{S'}$.

"$\Longrightarrow$". Let $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ be such that state-supp$(\mathbf{y}) \subseteq S'$ holds and $\mathbf{y}' = \mathbf{y}|_{\mathcal{E}'}$ be the restriction of $\mathbf{y}$ to $\mathcal{E}'$. By Lemma 4.22 we have $\mathbf{y}'\mathbf{A}' \leq \delta_{s_{in}}$ and $\mathbf{y}'\,\mathbf{t}' \geq \lambda$. Hence, $\mathbf{y}'$ is a Farkas certificate for $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\lozenge \text{target}) \geq \lambda$ and therefore $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\lozenge \text{target}) \geq \lambda$ holds by Theorem 3.24.

"$\Longleftarrow$". If $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\lozenge \text{target}) \geq \lambda$ holds, then we find a Farkas certificate $\mathbf{y}'$ for this property by Theorem 3.24, which satisfies $\mathbf{y}'\mathbf{A}' \leq \delta_{s_{in}} \wedge \mathbf{y}'\mathbf{t}' \geq \lambda$. Define $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ by $\mathbf{y}(s,\alpha) = \mathbf{y}'(s,\alpha)$ for $(s,\alpha) \in \mathcal{E}'$, and $\mathbf{y}(s,\alpha) = 0$ otherwise. Now Lemma 4.22 yields that $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{t} \geq \lambda$ must hold, and therefore we have $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$. This concludes the proof. □

Example 4.24. Consider again the MDP $\mathcal{M}$ defined in Figure 4.1 along with subsystems $\mathcal{M}_1$ and $\mathcal{M}_2$. Since $\mathbf{Pr}^{\max}_{\mathcal{M}_1}(\lozenge \text{target}) = 9/20$ holds, there must exist a Farkas certificate $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ for $\lambda = 9/20$ satisfying state-supp$(\mathbf{y}) \subseteq \{s_{in}, v\}$ by the above theorem. Such a certificate is given, for example, by

$$\mathbf{y} \;=\; \big((s_{in},\alpha) \mapsto 1,\ (u,\alpha) \mapsto 0,\ (v,\alpha) \mapsto 1/2,\ (v,\beta) \mapsto 0\big).$$

Similarly, since $\mathbf{Pr}^{\min}_{\mathcal{M}_2}(\lozenge \text{target}) = 1/8$ holds, there must exist a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ for $\lambda = 1/8$ satisfying supp$(\mathbf{z}) \subseteq \{s_{in}, u\}$. An example for such a certificate is

$$\mathbf{z} \;=\; \big(s_{in} \mapsto 1/8,\ u \mapsto 1/4,\ v \mapsto 0\big). \qquad \triangle$$

A direct consequence of Theorem 4.23 is that *minimal* witnessing subsystems and Farkas certificates with a *maximal* amount of zero-entries are related. Given some set $\mathcal{P}$, we say that a vector $\mathbf{v} \in \mathcal{P}$ has *minimal support* if $|\text{supp}(\mathbf{v})| = \min\{\ |\text{supp}(\mathbf{v}')|\ :\ \mathbf{v}' \in \mathcal{P}\ \}$ holds and *minimal state-support* if $|\text{state-supp}(\mathbf{v})| = \min\{\ |\text{state-supp}(\mathbf{v}')|\ :\ \mathbf{v}' \in \mathcal{P}\ \}$ holds. The latter is only defined for vectors whose domain is a subset of the state-action pairs of some MDP.

Corollary 4.25. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, and $\lambda \in [0,1]$. Then for all $S' \subseteq S$:*

1. *There exists $\mathbf{v} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with minimal support satisfying $S' = \text{supp}(\mathbf{v})$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem of $\mathcal{M}$ for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$.*
2. *There exists $\mathbf{v} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ with minimal state-support satisfying $S' = \text{state-supp}(\mathbf{v})$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem of $\mathcal{M}$ for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$.*

Proof. We only prove (1.) as (2.) is proven analogously.

(1.) "$\Longrightarrow$". Suppose, for contradiction, that there exists a witnessing subsystem $\mathcal{M}' = (S'' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ such that $|S''| < |S'|$. We may assume, without loss of generality,

that $\mathcal{M}' = \mathcal{M}_{S''}$. Then, by Theorem 4.23, there exists a vector $\mathbf{v}' \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\mathrm{supp}(\mathbf{v}') \subseteq S''$, but this contradicts the support-minimality of $\mathbf{v}$.

"$\Longleftarrow$". By Theorem 4.23, there exists a vector $\mathbf{v} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\mathrm{supp}(\mathbf{v}) \subseteq S'$. If $\mathbf{v}$ is not support-minimal or $\mathrm{supp}(\mathbf{v}) \subset S'$ holds, then we find $\mathbf{v}' \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $|\mathrm{supp}(\mathbf{v}')| < |S'|$. But then, again by Theorem 4.23, $\mathcal{M}_{\mathrm{supp}(\mathbf{v}')}$ is a witnessing subsystem, which contradicts minimality of $\mathcal{M}_{S'}$. □

Finally, we show that it suffices to inspect the vertices of the set of Farkas certificates (viewed as a polyhedron) to find a minimal witness.

Proposition 4.26. *Let $\mathcal{M}$ be an MDP in reachability form, $\mathfrak{m} \in \{\min, \max\}$, $\gtrsim \in \{\geq, >\}$ and $\lambda \in [0,1]$. Then, for each point $\mathbf{p} \in \mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\gtrsim}(\lambda)$ there exists a vertex $\mathbf{v} \in \mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\gtrsim}(\lambda)$ such that* $\mathrm{supp}(\mathbf{v}) \subseteq \mathrm{supp}(\mathbf{p})$.

Proof. Let $\mathcal{M} = (S \cup \{\mathrm{target}, \mathrm{exit}\}, \mathrm{Act}, s_{in}, P)$ be an MDP in reachability form and $\mathbf{p} \in \mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda)$. Consider the set $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^S_{\geq 0} \mid \mathbf{x}(s) = 0 \text{ for all } s \notin \mathrm{supp}(\mathbf{p})\} \subseteq \mathbb{R}^S_{\geq 0}$. The inequalities $\mathbf{x}(s) \geq 0$ (for $s \in S$) are part of the defining system of linear inequalities of $\mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda)$ and we have $\mathbf{p} \in \mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda) \cap \mathcal{H}$. This implies that $\mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda) \cap \mathcal{H}$ is a face of $\mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda)$. Furthermore, as $\mathcal{F}^{\min}_{\mathcal{M},\gtrsim}(\lambda) \cap \mathcal{H}$ is contained in the nonnegative orthant, it must include a vertex $\mathbf{v}$. As $\mathbf{v} \in \mathcal{H}$ holds, we have $\mathrm{supp}(\mathbf{v}) \subseteq \mathrm{supp}(\mathbf{p})$, which concludes the proof. The case $\mathfrak{m} = \max$ is proven analogously. □

The last proposition shows that one way of computing minimal witnessing subsystems is to enumerate all vertices of the corresponding set of Farkas certificates, and return the witnessing subsystem which corresponds to a vertex with a maximal amount of zeros. Vertex enumeration algorithms have been studied extensively [AF92, AF96] in the literature and implemented in state-of-the-art tools such as the Parma Polyhedra Library [BHZ08]. The implementations of vertex enumeration that we are aware of focus on exact numerical computations and do not scale well in the dimension (which in our case corresponds to the number of states of the system). We will now explore another approach to compute minimal witnesses based on (mixed-integer) linear programming. Mixed-integer linear programs have already been used in previous works to compute minimal witnesses [WJÁ+12, WJÁ+14], and the following section also contains a comparison with those approaches.

4.2.1 Mixed-integer programming formulations

This section considers how the relation between Farkas certificates and witnessing subsystems can be used to derive novel mixed-integer linear programming (MILP) formulations for the problem of computing minimal witnessing subsystems. Here we use that minimal witnesses correspond to Farkas certificates with a maximal number of zero entries, by Corollary 4.25.

It is known that finding a solution to a system of linear inequalities having at least k zero entries is NP-complete [GJ90]. Indeed, it is not even approximable in polynomial time within any constant factor unless $P = NP$ [AK98]. We generalize the problem by adding a labeling function which labels each dimension with a (possibly empty) set of labels. The problem asks for a solution such that the number of labels induced by its non-zero entries is minimal. This generalization will allow us to handle the label-based witness problem naturally.

Lemma 4.27. *Let $M \in \mathbb{Q}^{m\times n}$, $\mathbf{b} \in \mathbb{Q}^m$, $\mathbf{u} \in \mathbb{Q}^n$, $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Mx} \leq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}\}$ and assume that for all $\mathbf{p} \in \mathcal{P}$ we have $\mathbf{p} \leq \mathbf{u}$. Furthermore, let $\Lambda : \{1, \ldots, n\} \to 2^L$ be a labeling function into a finite set of labels L and define $\Lambda(\mathbf{x}) = \{l \in L \mid$ there exists i such that $\mathbf{x}(i) > 0$ and $l \in \Lambda(i)\}$, for $\mathbf{x} \in \mathbb{R}^n$.*

Consider the following MILP: minimize $\sum_{l\in L} \sigma(l)$ such that $\sigma \in \{0,1\}^L$ and

$$\mathbf{Mx} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}, \quad \text{and} \quad \mathbf{x}(i) \leq \sigma(l) \cdot \mathbf{u}(i) \quad \text{for all} \quad 1 \leq i \leq n \text{ and } l \in \Lambda(i).$$

Then for all $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$: there exists $\sigma \in \{0,1\}^L$ such that $(\sigma, \mathbf{x})$ is an optimal solution of this MILP if and only if $\mathbf{x}$ is a point in $\mathcal{P}$ such that $|\Lambda(\mathbf{x})|$ is minimal among all points in $\mathcal{P}$.

Proof. "$\Longrightarrow$". Suppose that $(\sigma, \mathbf{x}) \in \{0,1\}^L \times \mathbb{Q}^n$ is an optimal solution of the MILP, and $\mathbf{x}'$ is a point in $\mathcal{P}$ with $|\Lambda(\mathbf{x}')| < |\Lambda(\mathbf{x})|$. Let $\sigma' \in \{0,1\}^L$ be the vector such that $\sigma'(l) = 1$ iff $l \in \Lambda(\mathbf{x}')$. Clearly $(\sigma', \mathbf{x}')$ is a solution of the MILP with a better objective value, which contradicts the assumption.

"$\Longleftarrow$". Suppose that $\mathbf{x}$ is a point in $\mathcal{P}$ such that $|\Lambda(\mathbf{x})|$ is minimal, and let $\sigma \in \{0,1\}^n$ be the vector such that $\sigma(l) = 1$ iff $l \in \Lambda(\mathbf{x})$, for all $1 \leq i \leq n$ and $l \in L$. Assume, for contradiction, that $(\sigma, \mathbf{x})$ is not an optimal solution of the MILP. Then we find a solution $(\sigma', \mathbf{x}')$ with better objective value. This implies $|\Lambda(\mathbf{x}')| < |\Lambda(\mathbf{x})|$ and thereby contradicts our assumption. □

By taking $L = \{1, \ldots, n\}$ and $\Lambda(i) = \{i\}$ for all $i \in \{1, \ldots, n\}$ in the above lemma we get exactly the problem of maximizing the number of zero entries.

The min-witness program

The above lemma can be used to derive MILPs whose solutions correspond to Farkas certificates with a maximal number of zero entries and thereby, using Corollary 4.25, to minimal witnessing subsystems. To define the MILPs more concisely, we will use expressions of the form $\mathbf{x} \in \mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\bowtie}(\lambda)$ as placeholders for the defining linear inequalities of the corresponding set of Farkas certificates (Definition 3.23).

Definition 4.28 (min-witness MILP). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form with min-relevant states S_R and $\lambda \in [0,1] \cap \mathbb{Q}$.

The min-witness MILP for $(\mathcal{M}, \lambda)$ is defined as follows:

$$\text{minimize} \sum_{s\in S_R} \sigma(s) \ \text{ such that } \ \mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda), \ \ \mathbf{z} \leq \sigma, \text{and } \ \sigma \in \{0,1\}^{S_R}.$$

Proposition 4.29. *Let $\mathcal{M}$ be an MDP in reachability form, S_R its min-relevant states and $\lambda \in [0,1] \cap \mathbb{Q}$.*

Then, for all $S' \subseteq S_R$: there exists an optimal solution $(\sigma, \mathbf{z})$ of the min-witness MILP for $(\mathcal{M}, \lambda)$ with $\text{supp}(\mathbf{z}) = S'$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\, \text{target}) \geq \lambda$.

Proof. Our goal is to apply Lemma 4.27, which introduces a generic MILP to find points from a nonnegative polytope $\mathcal{P}$ with a maximal amount of zeros. We let $\mathcal{P} = \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ (recall that all points in $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ are nonnegative by definition) and $\Lambda(s) = \{s\}$ for each $s \in S_R$. That is, each state is labeled by a unique label and hence the minimization objective in Lemma 4.27 amounts to minimizing the number of non-zero entries over all points in $\mathcal{P}$, i.e., we have $\Lambda(\mathbf{z}) = \text{supp}(\mathbf{z})$

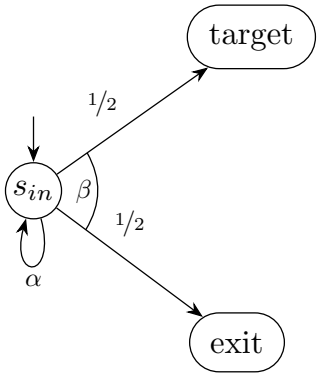


Figure 4.3: An example MDP $\mathcal{M}$ with two enabled actions α and β in the initial state. The expected number of visits of s_{in} can be made arbitrarily large with randomized schedulers. For a similar reason, the set of Farkas certificates $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ is unbounded for any $\lambda \geq 1/2$ (see Example 4.30).

(where $\Lambda(\mathbf{z})$ is defined as in Lemma 4.27) for all vectors $\mathbf{z} \in \mathbb{R}^{S_R}_{\geq 0}$. By additionally setting $\mathbf{u}(s) = 1$ for each $s \in S_R$, we have exactly the min-witness MILP as defined above.

To apply Lemma 4.27 we need to show that $\mathbf{u}(s) = 1$ is indeed an upper bound on $\mathbf{x}(s)$, for all vectors $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$. Let $\mathcal{M}^*$ be the largest min-relevant subsystem of $\mathcal{M}$, as defined in Definition 4.9. As the MDP $\mathcal{M}^*$ is EC-free by construction, $\mathbf{pr}^{\min}_{\mathcal{M}^*}$ is a point-wise upper bound on all vectors in $\mathcal{F}^{\min}_{\mathcal{M}^*,\geq}(\lambda)$ by Lemma 2.9. Furthermore we have $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda) = \mathcal{F}^{\min}_{\mathcal{M}^*,\geq}(\lambda)$. As $\mathbf{pr}^{\min}_{\mathcal{M}^*} \leq \mathbf{1}$, it follows that all vectors in $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ are point-wise bounded from above by one.

Now we are in a position to apply Lemma 4.27 and thereby show the claim.

"$\Longrightarrow$". If $(\sigma, \mathbf{z})$ is an optimal solution of the min-witness MILP with $\mathrm{supp}(\mathbf{z}) = S'$, then $\mathbf{z}$ has minimal support by Lemma 4.27 and hence $\mathcal{M}_{S'}$ is a minimal witnessing subsystem by Corollary 4.25.

"$\Longleftarrow$". If $\mathcal{M}_{S'}$ is a minimal witnessing subsystem, then by Corollary 4.25 there exists a vector $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with minimal support satisfying $\mathrm{supp}(\mathbf{z}) = S'$. But then, by Lemma 4.27, there exists σ such that $(\sigma, \mathbf{z})$ is an optimal solution of the min-witness MILP. □

The max-witness program

We turn to the definition of a corresponding MILP for minimal witnesses with respect to maximal reachability probabilities. An issue which arises here is that the corresponding set of Farkas certificates may be unbounded in the presence of proper end components (see Example 4.30). This is in contrast to the case of EC-free MDPs, where the set of solutions of $\mathbf{yA} \leq \delta_{s_{in}}$ is always bounded (Proposition 3.9). Hence, a vector $\mathbf{u}$ as required by the generic MILP defined in Lemma 4.27 may not exist at all.

Example 4.30. Consider the MDP $\mathcal{M}$ as shown in Figure 4.3. If $\mathbf{A}, \mathbf{t}$ are the system matrix and target vector of $\mathcal{M}$, then the system of linear inequalities $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$ is given by

$$\mathbf{y}(s_{in}, \alpha) \;+\; \mathbf{y}(s_{in}, \beta) \;\leq\; 1 \;+\; \mathbf{y}(s_{in}, \alpha) \quad \wedge \quad 1/2 \cdot \mathbf{y}(s, \beta) \;\geq\; \lambda.$$

If $\lambda = 1/2$, then for any $a \in \mathbb{R}$ the vector $\mathbf{y} = \big((s_{in}, \alpha) \mapsto a,\ (s_{in}, \beta) \mapsto 1\big)$ is a solution. Hence the set $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ is not bounded, as it contains precisely the solutions of the above system of linear inequalities.

For a given $a \in \mathbb{R}$, consider the memoryless randomized scheduler $\mathfrak{S}_a$ defined by $\mathfrak{S}_a(s_{in}, \alpha) = \frac{a}{a+1}$ and $\mathfrak{S}_a(s_{in}, \beta) = \frac{1}{a+1}$. The expected number of visits of state-action pair (s_{in}, α) under $\mathfrak{S}_a$ is a. This shows that we can give no upper bound on the expected number of visits when ranging over all memoryless randomized schedulers, if proper end components are present. △

If a scheduler $\mathfrak{S}$ reaches {target, exit} with probability one, then the vector $\mathbf{ev}^{\mathfrak{S}}$ containing the expected number of visits of $\mathfrak{S}$ satisfies $\mathbf{ev}^{\mathfrak{S}}\mathbf{A} = \delta_{s_{in}}$ (Proposition 3.13). Hence, $\mathbf{ev}^{\mathfrak{S}}$ is potentially a Farkas certificate contained in $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ (the definition of $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ requires additionally that $\mathbf{ev}^{\mathfrak{S}}\mathbf{t} \geq \lambda$ should hold). As we now deal with MDPs which are not necessarily EC-free, the set of vectors $\mathbf{ev}^{\mathfrak{S}}$ one gets when ranging over all such schedulers $\mathfrak{S}$ may be unbounded. This holds even if we consider only memoryless randomized schedulers (see Example 4.30). However, for the sake of proving that a subsystem is a witness, it is enough to consider only memoryless deterministic schedulers which satisfy the above property (see Proposition 2.10).

For an MDP $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ in reachability form, let us define for each enabled state-action pair $(s, \alpha) \in \mathcal{E}$:

$$\mathbf{u}_{ev}(s,\alpha) = \max_{\substack{\mathfrak{S} \in MD(\mathcal{M}),\\ \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond\{\text{target},\text{exit}\})=1}} \mathbf{ev}^{\mathfrak{S}}(s,\alpha) \tag{4.1}$$

This vector contains for each (s, α) the maximum expected number of visits of (s, α) when ranging over all memoryless and deterministic schedulers which reach {target, exit} with probability one. For the MDP defined in Figure 4.3, we have $\mathbf{u}_{ev}(s_{in}, \alpha) = 0$ and $\mathbf{u}_{ev}(s_{in}, \beta) = 1$, since no memoryless deterministic scheduler which reaches {target, exit} with probability one can choose action α.

The set $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ restricted to vectors which are bounded from above by $\mathbf{u}_{ev}$ still contains enough points to find all minimal witnessing subsystems, as the following lemma shows.

Lemma 4.31. *Let $\mathcal{M}$ be an MDP in reachability form and $\lambda \in [0, 1]$. Then for all $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ there exists $\mathbf{y}' \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ satisfying $\mathbf{y}' \leq \mathbf{u}_{ev}$, $|\text{supp}(\mathbf{y}')| \leq |\text{supp}(\mathbf{y})|$ and* state-supp$(\mathbf{y}') \subseteq$ state-supp$(\mathbf{y})$.

Proof. Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ with enabled state-action pairs $\mathcal{E}$, $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ and consider $S' =$ state-supp$(\mathbf{y})$. By statement (2.) of Theorem 4.23, the subsystem $\mathcal{M}' = \mathcal{M}_{S'}$ induced by S' is a witness for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$. Then, there exists a memoryless and deterministic scheduler $\mathfrak{S}'$ for $\mathcal{M}'$ which satisfies $\Pr^{\mathfrak{S}'}_{\mathcal{M}',s_{in}}(\Diamond\{\text{target}, \text{exit}\}) = 1$ and $\Pr^{\mathfrak{S}'}_{\mathcal{M}'}(\Diamond \text{target}) \geq \lambda$ (see Proposition 2.10). Let $\mathcal{E}' = \{(s, \alpha) \in \mathcal{E} \mid s \in S'\}$, $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$, $\mathbf{t}' = \mathbf{t}|_{\mathcal{E}'}$ be the system matrix and target vector of $\mathcal{M}'$, and $\mathbf{ev}^{\mathfrak{S}'} \in \mathbb{R}^{\mathcal{E}'}$ be the expected number of visits in $\mathcal{M}'$ under $\mathfrak{S}'$. It follows that $\Pr^{\mathfrak{S}'}_{s_{in}}(\Diamond \text{target}) = \mathbf{ev}^{\mathfrak{S}'} \cdot \mathbf{t}' \geq \lambda$ holds, and, from Proposition 3.13, that $\mathbf{ev}^{\mathfrak{S}'} \cdot \mathbf{A}' = \delta_{s_{in}}$ holds. Hence, by definition, we have $\mathbf{ev}^{\mathfrak{S}'} \in \mathcal{F}^{\max}_{\mathcal{M}',\geq}(\lambda)$. Observe that state-supp$(\mathbf{ev}^{\mathfrak{S}'}) \subseteq S'$ holds trivially, as the domain of $\mathbf{ev}^{\mathfrak{S}'}$ is $\mathcal{E}'$.

Let $\mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be defined by $\mathbf{y}'(s, \alpha) = \mathbf{ev}^{\mathfrak{S}'}(s, \alpha)$ if $(s, \alpha) \in \mathcal{E}'$, and $\mathbf{y}'(s, \alpha) = 0$ otherwise. It follows from Lemma 4.22 that $\mathbf{y}' \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$. In particular, this implies

$$|\text{supp}(\mathbf{y}')| \;=\; |\text{supp}(\mathbf{ev}^{\mathfrak{S}'})| \;\leq\; |S'| \;\leq\; |\text{supp}(\mathbf{y})|,$$

as $\mathbf{ev}^{\mathfrak{S}'}$ has at most one non-zero entry per state of $\mathcal{M}'$, which follows from the fact that $\mathfrak{S}'$ is memoryless and deterministic. Furthermore, we have

$$\text{state-supp}(\mathbf{y}') \;=\; \text{state-supp}(\mathbf{ev}^{\mathfrak{S}'}) \;\subseteq\; S' \;=\; \text{state-supp}(\mathbf{y}).$$

It remains to show that $\mathbf{y}' \leq \mathbf{u}_{ev}$ holds. Consider any memoryless and deterministic scheduler $\mathfrak{S}$ for $\mathcal{M}$ which satisfies $\mathfrak{S}(s) = \mathfrak{S}'(s)$ for all $s \in S'$. Clearly we have $\mathbf{y}' \leq \mathbf{ev}^{\mathfrak{S}'} \leq \mathbf{ev}^{\mathfrak{S}}$, and hence also $\mathbf{y}' \leq \mathbf{u}_{ev}$ by definition. □

To define the mixed-integer linear program which computes minimal witnessing subsystems for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$, we depend on an upper bound $\mathbf{u}$ of $\mathbf{u}_{ev}$ which has to be computed a priori. Having computed such a bound one can use the above lemma, which shows that it is enough to consider Farkas certificates bounded by $\mathbf{u}_{ev}$.

Definition 4.32 (max-witness MILP). Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form, $\lambda \in [0,1] \cap \mathbb{Q}$ and $\mathcal{E}$ the enabled state-action pairs of $\mathcal{M}$. Furthermore, let $\mathbf{u} \in \mathbb{Q}^{\mathcal{E}}$ be such that $\mathbf{u}_{ev} \leq \mathbf{u}$, where $\mathbf{u}_{ev}$ is as defined in Equation (4.1).

The max-witness MILP for $(\mathcal{M}, \lambda)$ using $\mathbf{u}$ is defined as follows:

$$\text{minimize} \sum_{s \in S} \sigma(s) \;\text{ such that }\; \mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda),\ \sigma \in \{0,1\}^S \text{ and}$$
$$\mathbf{y}(s,\alpha) \leq \mathbf{u}(s,\alpha) \cdot \sigma(s) \;\text{ for all }\; (s,\alpha) \in \mathcal{E}.$$

Proposition 4.33. *Let $\mathcal{M}, \mathcal{E}, \lambda$ and $\mathbf{u}$ be as in Definition 4.32.*

Then, for all $S' \subseteq S$: there exists an optimal solution $(\sigma, \mathbf{y})$ of the max-witness MILP for $(\mathcal{M}, \lambda)$ using $\mathbf{u}$ such that state-supp$(\mathbf{y}) = S'$ *if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$.*

Proof. Let us define the labeling function $\Lambda : \mathcal{E} \to 2^S$ which maps each state-action pair to the corresponding state (i.e., $\Lambda(s,\alpha) = \{s\}$, for all $(s,\alpha) \in \mathcal{E}$). The max-witness MILP indeed corresponds to the generic MILP defined in Lemma 4.27 under labeling Λ, if we choose

$$\mathcal{P} = \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda) \cap \{\, \mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}' \leq \mathbf{u} \,\}.$$

Observe that under labeling Λ we have $\Lambda(\mathbf{y}) = \text{state-supp}(\mathbf{y})$ for all $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$, where $\Lambda(\mathbf{y})$ is defined as in Lemma 4.27.

"$\Longrightarrow$". Let $(\sigma, \mathbf{y})$ be an optimal solution of the max-witness MILP such that state-supp$(\mathbf{y}) = S'$. By Lemma 4.27, $\mathbf{y}$ has minimal state-support among all vectors in $\mathcal{P}$. From Lemma 4.31 it follows that $\mathbf{y}$ also has minimal state-support among all vectors in $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$. But then, $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$ by Corollary 4.25.

"$\Longleftarrow$". Let $\mathcal{M}_{S'}$ be a minimal witnessing subsystem. Then, there exists $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda) \cap \{\, \mathbf{y}' \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}' \leq \mathbf{u} \,\}$ with minimal state-support satisfying $S' = \text{state-supp}(\mathbf{y})$, by Corollary 4.25 and Lemma 4.31. But then, by Lemma 4.27, there exists σ such that $(\sigma, \mathbf{y})$ is an optimal solution of the max-witness MILP. □

To use the above MILP one has to first compute an upper bound on $\mathbf{u}_{ev}$. This problem is looked at more carefully in Section 4.2.2, where we show that for the special case of EC-free MDPs (and hence also for Markov chains), $\mathbf{u}_{ev}$ can be computed precisely in polynomial time. The following remark discusses a way to circumvent the computation of $\mathbf{u}_{ev}$ by adding certain disjunctive constraints to the MILP.

Remark 4.34 (bigM technique and indicator constraints). The generic MILP to find vectors with a maximal number of non-zero entries (Lemma 4.27) uses constraints of the form

$$\mathbf{x}(i) \leq \sigma(i) \cdot \mathbf{u}(i),$$

where $\mathbf{u}(i)$ is a known upper bound on $\mathbf{x}(i)$ over all feasible solutions $\mathbf{x}$. This idea of "charging" (if $\sigma(i) = 0$) or "discharging" (if $\sigma(i) = 1$) the constraint $\mathbf{x}(i) \leq 0$ is a well-known technique in

integer programming, and is often referred to as the bigM-technique. Here M (used instead of $\mathbf{u}(i)$) is meant to be an upper bound on all entries of all feasible solutions. A way to specify this type of constraints without knowledge of M has been introduced under the name of *indicator constraints*, which allows formulating the above constraint as follows:

$$\sigma(i) = 0 \implies \mathbf{x}(i) \leq 0.$$

Essentially, *indicator constraints* allow modeling a form of disjunctions, and dedicated procedures to solve MILPs with such constraints present have been studied [BLTW15, BBF+16]. Modern mathematical optimization solvers such as Gurobi and Cplex support indicator constraints. △

Markov chains

For Markov chains, the min-witness MILP and the max-witness MILP can both be used to compute minimal witnessing subsystems. This is because $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target})$ holds if $\mathcal{M}$ is a Markov chain. Although the two sets of Farkas certificates $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ and $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ do not coincide (see Example 3.6), the max- and min-witness MILPs will return the same optimal objective value (namely, the size of a minimal witnessing subsystem). However, as we will see when introducing heuristic approaches based on computing Farkas certificate, algorithms based on the two formulations may behave differently. For Markov chains, $\mathbf{u}_{ev}$ equals the expected number of visits for each state when starting in s_{in} (see Equation (4.1)). Hence, $\mathbf{u}_{ev}$ can be computed by solving a linear equation system (see Lemma 2.15).

Comparison to known methods using MILPs

The problem of finding minimal witnessing subsystems for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda$ has been considered in [WJÁ+14], where a formulation of the problem as a MILP is given. It uses a variable p_s for each state s of $\mathcal{M}$, which represents the probability achieved under some scheduler, and binary variables $\sigma_{(s,\alpha)}$ for each enabled state-action pair. The program ensures that $p_s \leq \sum_{\alpha\in\mathrm{Act}(s)} \sigma_{(s,\alpha)} \leq 1$ for all $s \in S$, which implies that variables $\sigma_{(s,\alpha)}$ induce a memoryless deterministic scheduler. States s satisfying $\sum_{\alpha\in\mathrm{Act}(s)} \sigma_{(s,\alpha)} = 0$ are the ones which are excluded in the corresponding subsystem. The core of the program are the constraints $p_{s_{in}} \geq \lambda$ and

$$p_s \leq (1 - \sigma_{(s,\alpha)}) + \sum_{s'\in S} P(s,\alpha,s') \cdot p_{s'} \quad \text{for all } (s,\alpha) \in \mathcal{E},$$

where $\mathcal{E}$ is the set of enabled state-action pairs. As $p_s \leq 1$ is ensured independently of these constraints, they are equivalent to the indicator constraints (see Remark 4.34):

$$(\sigma_{(s,\alpha)} = 1) \implies p_s \leq \sum_{s'\in S} P(s,\alpha,s') \cdot p_{s'} \quad \text{for all } (s,\alpha) \in \mathcal{E}.$$

Intuitively, variables $\sigma_{(s,\alpha)}$ determine a memoryless deterministic scheduler, and the above constraints make sure that no value p_s exceeds the reachability probability that state s achieves under that scheduler. Finally, the number of states s such that p_s is non-zero is minimized in a similar way as is done in Lemma 4.27.

In order to correctly treat end components, the MILP also includes another $|S|^2$ binary variables $t_{s,s'}$, for each pair $s, s' \in S$ together with constraints which make sure that every selected state contains a path to {target, exit} under the scheduler induced by σ-variables (see

constraints ((8g) - (8i)) in [WJÁ+14]). This MILP is fundamentally different from the max-witness MILP (Definition 4.32), as the latter uses variables which correspond to the expected number of visits of state-action pairs, rather than variables for the achieved probability in a state. The max-witness MILP requires only $|S|$ binary variables, rather than $|\mathcal{E}| + |S|^2$. A disadvantage of using the max-witness MILP is that it requires an upper bound $\mathbf{u}$ of $\mathbf{u}_{ev}$.

One should note that the max-witness MILP does not require any special treatment with respect to end components (once an upper bound on $\mathbf{u}_{ev}$ is known), in contrast to the solution in [WJÁ+14]. Intuitively, the reason for this is that the "value" (i.e., the maximal probability achieved) of a solution is determined by expression $\mathbf{yt}$ (see the definition of $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ in Definition 3.23). But no state-action pair (s,α) satisfying $\mathbf{t}(s,\alpha) > 0$ is part of a proper end component. Hence, in this formulation, artificially increasing values $\mathbf{y}$ by staying inside a proper end component does not lead to a (spuriously) higher value. The MILP for DTMCs given in [WJÁ+14] coincides with the min-witness MILP (Definition 4.28) in the special case of DTMCs.

The weighted and labeled witness problems

The max- and min-witness MILPs can be extended to solve the label-based and weighted witness problems (see Definition 4.13). In the MILPs we have seen, an assignment to the binary vector $\sigma \in \{0,1\}^S$ represents a subsystem. To handle the weighted problem one only needs to adapt the objective function such that, for each assignment of σ, it returns the total weight of the corresponding subsystem. If $wgt : S \to \mathbb{N}$ is a weight function, then the objective function $\sum_{s\in S} \sigma(s)$ (which counts the number of included states) is replaced by $\sum_{s\in S} wgt(s)\cdot\sigma(s)$.

For the label-based problem, one can use a binary variable per label to count how many of the labels are "present" in a subsystem. If $\Lambda : S \to 2^L$ is a labeling function of the states into a finite set of labels L, then the new objective function is $\sum_{l\in L}\sigma(l)$ and the constraints of the MILPs are adapted such that a positive entry of the Farkas certificate in some state s (respectively state-action pair (s,α)) forces all labels in $\Lambda(s)$ to have a one-entry in σ.

Definition 4.35. Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form with min-relevant states S_R, enabled state-action pairs $\mathcal{E}$, $\lambda \in [0,1]\cap\mathbb{Q}$ and $\mathbf{u}$ be such that $\mathbf{u}_{ev} \leq \mathbf{u}$, where $\mathbf{u}_{ev}$ is defined as in Equation (4.1). Let $wgt : S \to \mathbb{N}$ be a weight function on $\mathcal{M}$ and $\Lambda : S \to 2^L$ a labeling of $\mathcal{M}$ into a finite set of labels L.

The weighted min- and max-witness MILPs ((1.) and (3.)), and the label-based min- and max-witness MILPs ((2.) and (4.)) are defined as follows:

(1.) minimize $\displaystyle\sum_{s\in S_R} wgt(s)\cdot\sigma(s)$ such that $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, $\mathbf{x} \leq \sigma$ and $\sigma \in \{0,1\}^{S_R}$.

(2.) minimize $\displaystyle\sum_{l\in L}\sigma(l)$ such that $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, $\sigma \in \{0,1\}^L$ and $\mathbf{x}(s) \leq \sigma(l)$ for all $s \in S_R$ and $l \in \Lambda(s)$.

(3.) minimize $\displaystyle\sum_{s\in S} wgt(s)\cdot\sigma(s)$ such that $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$, $\sigma \in \{0,1\}^S$ and $\mathbf{y}(s,\alpha) \leq \mathbf{u}(s,\alpha)\cdot\sigma(s)$ for all $(s,\alpha) \in \mathcal{E}$.

(4.) minimize $\displaystyle\sum_{l\in L}\sigma(l)$ such that $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$, $\sigma \in \{0,1\}^L$ and $\mathbf{y}(s,\alpha) \leq \mathbf{u}(s,\alpha)\cdot\sigma(l)$ for all $(s,\alpha) \in \mathcal{E}$ and $l \in \Lambda(s)$.

Proposition 4.36. *Optimal solutions of the weighted and label-based min-and max-witness MILPs as defined in Definition 4.35 correspond to minimal witnessing subsystems with respect to the corresponding notions of size.*

Proof. (1.) "$\Longrightarrow$": Suppose $(\sigma, \mathbf{x}) \in \{0,1\}^{S_R} \times \mathbb{Q}^{S_R}$ is an optimal solution of the MILP (1.) and let $S_1 = \{s \in S_R \mid \sigma(s) = 1\}$. Suppose, for contradiction, that there exists a witnessing subsystem $\mathcal{M}'$ of $\mathcal{M}$ such that $wgt(\mathcal{M}') < wgt(\mathcal{M}_{S_1})$, and let $S_2 \subseteq S_R$ be such that $\mathcal{M}_{S_2} = \mathcal{M}'$ (we may assume that $\mathcal{M}'$ is of this form). Consider the vector $\sigma' \in \{0,1\}^{S_R}$ defined by $\sigma'(s) = 1$ iff $s \in S_2$. As $\mathcal{M}'$ is witnessing, we find a Farkas certificate $\mathbf{x}' \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\mathrm{supp}(\mathbf{x}') \subseteq S_2$, by Theorem 4.23. Clearly $(\sigma', \mathbf{x}')$ is a feasible solution of the MILP, and furthermore we have

$$\sum_{s \in S_R} wgt(s) \cdot \sigma'(s) \;=\; \sum_{s \in S_2} wgt(s) \;=\; wgt(\mathcal{M}') \;<\; wgt(\mathcal{M}_{S_1}) \;=\; \sum_{s \in S_R} wgt(s) \cdot \sigma(s),$$

which shows that $(\sigma', \mathbf{x}')$ achieves a better value in the objective function and contradicts the assumption that $(\sigma, \mathbf{x})$ is an optimal solution.

"$\Longleftarrow$": Let $\mathcal{M}'$ be a witnessing subsystem with states $S_1 \cup \{\text{target}, \text{exit}\}$ such that $wgt(\mathcal{M}')$ is minimal. By Theorem 4.23 there exists $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\mathrm{supp}(\mathbf{x}) \subseteq S_1$. Define $\sigma \in \{0,1\}^{S_R}$ such that $\sigma(s) = 1$ iff $s \in S_1$. Suppose that $(\sigma, \mathbf{x})$ is not an optimal solution of the MILP and let $(\sigma', \mathbf{x}')$ be a better solution. Let $S_2 = \{s \in S_R \mid \sigma'(s) = 1\}$. The subsystem $\mathcal{M}_{S_2}$ is a witnessing subsystem, by Theorem 4.23. As in the other case, we have $wgt(\mathcal{M}_{S_2}) = \sum_{s \in S_R} wgt(s) \cdot \sigma'(s) < \sum_{s \in S_R} wgt(s) \cdot \sigma(s) = wgt(\mathcal{M}')$. This contradicts the weight-minimality of $\mathcal{M}'$.

(2.) Observe that the MILP has the structure of the generic MILP defined in Lemma 4.27, using $\mathcal{P} = \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, $\mathbf{u} = \mathbf{1}$ and the labeling function Λ. Hence, by the same lemma, optimal solutions of the MILP correspond to points $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that

$$|\Lambda(\mathbf{x})| \;=\; |\{l \in L \mid \text{ there exists } s \text{ s.t. } \mathbf{x}(s) > 0 \text{ and } l \in \Lambda(s)\}|$$

is minimal. On the other hand, for each $S_1 \subseteq S_R$ we have: there exists $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $|\Lambda(\mathbf{x})|$ is minimal and $\mathrm{supp}(\mathbf{x}) = S_1$ if and only if $\mathcal{M}_{S_1}$ is a witnessing subsystem with a minimal number of appearing labels. This is shown below.

"$\Longrightarrow$": Consider $\mathbf{x}$ as above and suppose, for contradiction, that there exists a witnessing subsystem $\mathcal{M}_{S_2}$ such that $|\Lambda(\mathcal{M}_{S_2})| < |\Lambda(\mathcal{M}_{S_1})|$. Then, we find $\mathbf{x}_2 \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with $\mathrm{supp}(\mathbf{x}_2) \subseteq S_2$ by Theorem 4.23 and hence $|\Lambda(\mathbf{x}_2)| \leq |\Lambda(\mathcal{M}_{S_2})| < |\Lambda(\mathcal{M}_{S_1})| = |\Lambda(\mathbf{x})|$. But this contradicts the fact that $\mathbf{x}$ is a point in $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ which minimizes $|\Lambda(\cdot)|$.

"$\Longleftarrow$": Assume that $\mathcal{M}_{S_1}$ is a witnessing subsystem with a minimal number of appearing labels. By Theorem 4.23 we find $\mathbf{x} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\mathrm{supp}(\mathbf{x}) \subseteq S_1$. We claim that $\mathbf{x}$ minimizes $|\Lambda(\cdot)|$ among all points in $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$. Suppose, for contradiction, that there exists $\mathbf{x}' \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with $|\Lambda(\mathbf{x}')| < |\Lambda(\mathbf{x})|$ and let $S_2 = \mathrm{supp}(\mathbf{x}')$. But then, $\mathcal{M}_{S_2}$ is a witnessing subsystem (by Theorem 4.23) and $|\Lambda(\mathcal{M}_{S_2})| = |\Lambda(\mathbf{x}')| < |\Lambda(\mathbf{x})| = |\Lambda(\mathcal{M}_{S_1})|$. This contradicts our assumption that $\mathcal{M}_{S_1}$ minimizes $|\Lambda(\cdot)|$ over all witnessing subsystems of $\mathcal{M}$.

Hence the solutions of (2.) correspond to witnessing subsystems of $\mathcal{M}$ for $\mathbf{Pr}^{\min}(\Diamond\,\text{target}) \geq \lambda$ which include a minimal number of labels.

The correctness proofs for MILPs (3.) and (4.) are analogous. □

4.2.2 Computing upper bounds on $\mathbf{u}_{ev}$

To use the max-witness MILP we need to be able to compute an upper bound $\mathbf{u}$ on $\mathbf{u}_{ev}$ (as defined in Equation (4.1)). The fact that the definition of $\mathbf{u}_{ev}$ ranges over all memoryless deterministic schedulers which *do not realize a proper end component* indicates a difference to, e.g., the computation of maximal expected total reward. In the latter problem choosing a maximizing action locally in each state is sufficient, but this is not true in the former problem. Rather, there the choice in any state depends on which actions are chosen in other states in order to avoid completing a proper end component, and hence this choice is not local.

We first show how a generic bound on $\mathbf{u}_{ev}$ can be computed in polynomial time, which depends only on the number of states of the MDP and its least non-zero transition probability. The idea used to derive the bound is well-known, and we spell it out here only for completeness.

Lemma 4.37. *Let $\mathcal{M} = (S \cup \{\text{target}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form whose least non-zero transition probability is ϵ, $n = |S|$ and let $\mathbf{u}_{ev}$ be defined as in Equation* (4.1)*. Then:*

$$\max_{(s,\alpha)\in\mathcal{E}} \mathbf{u}_{ev}(s,\alpha) \;\leq\; n/\epsilon^{2n}$$

Proof. Let $T = \{\text{target}, \text{exit}\}$ for this proof. We show that n/ϵ^{2n} is an upper bound on the expected number of steps that any memoryless and deterministic scheduler satisfying $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond T) = 1$ needs before reaching T. Then, n/ϵ^{2n} is clearly also an upper bound on $\max_{(s,\alpha)\in\mathcal{E}} \mathbf{u}_{ev}(s,\alpha)$.

Let $\mathfrak{S}$ be a memoryless and deterministic scheduler for $\mathcal{M}$ such that $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond T) = 1$. By assumption, from each state of $\mathcal{M}$ there exists a path reaching T of length at most n and probability at least ϵ^n. Consequently, the probability of not reaching T in at most n steps (henceforth denoted $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Box^{\leq n}\,\overline{T})$) is at most $1-\epsilon^n$. We will use the notation $\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond^{=i}\,T)$ to denote the probability of reaching T for the first time in exactly T steps. Now we calculate:

$$\sum_{i\geq 1} i\cdot \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond^{=i}\,T) \;=\; \sum_{i\geq 0}\sum_{j=1}^{n}(in+j)\cdot \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond^{=(in+j)}\,T) \;\leq\; \sum_{i\geq 0} n(i+1)\cdot \sum_{j=1}^{n}\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond^{=(in+j)}\,T)$$

$$\leq\; \sum_{i\geq 0} n(i+1)\cdot \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Box^{\leq in}\,\overline{T}) \;\leq\; \sum_{i\geq 0} n(i+1)\cdot(1-\epsilon^n)^i \;=\; n/\epsilon^{2n}.$$

In the calculation we use $\sum_{j=1}^{n}\Pr^{\mathfrak{S}}_{\mathcal{M}}(\Diamond^{=(in+j)}\,T) \leq \Pr^{\mathfrak{S}}_{\mathcal{M}}(\Box^{\leq in}\,\overline{T})$, which follows for all $i \geq 0$ from the fact that both states in T are absorbing. □

It follows that the vector $\mathbf{u} \in \mathbb{Q}^{\mathcal{E}}$ defined by $\mathbf{u}(s,\alpha) = n/\epsilon^{2n}$, where n is the number of states of the given MDP and ϵ is its least non-zero transition probability, can be used as an upper bound for the max-witness MILP. This value may be very large, however, and in practice this leads to two problems: The MILP may be numerically unstable, and its continuous relaxation is weaker, which makes it harder to solve. Hence, computing a tighter bound on $\mathbf{u}_{ev}$ is an important problem in this context.

Computing $\mathbf{u}_{ev}$ in MDPs with small end components

We will now present an algorithm to compute $\mathbf{u}_{ev}$ exactly which runs in time $O(|\text{Act}|^K)\cdot \text{poly}(|\mathcal{M}|)$, where K is the maximal number of states of any maximal end component (MEC) in the MDP $\mathcal{M}$. Using it we can compute $\mathbf{u}_{ev}$ in polynomial time if the size of all end components

is bounded from above by a constant. The algorithm is based on an explicit enumeration of memoryless and deterministic schedulers inside maximal end components.

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form with enabled state-action pairs $\mathcal{E}$, and let $(E_1, A_1), \ldots, (E_k, A_k)$ be the maximal end components of $\mathcal{M}$, excluding those induced by "target" and "exit". Our goal is to compute the value $\mathbf{u}_{ev}(t, \beta)$, for some $(t, \beta) \in \mathcal{E}$. Let (E, A) be the unique maximal end component of $\mathcal{M}$ such that $t \in E$. Let $\mathfrak{S}$ be a memoryless and deterministic scheduler which assigns to each state $s \in E$ an action in $\text{Act}(s)$. We will assume that for all $s \in E$ we have $\Pr^{\mathfrak{S}}_{\mathcal{M},s}(\Diamond \neg E) = 1$, which means that with probability one the end component is left from every state in E under $\mathfrak{S}$. If this condition is satisfied, we call $\mathfrak{S}$ an *internal scheduler* of (E, A).

Given an internal scheduler $\mathfrak{S}$ of (E, A) we construct the MDP $\mathcal{N}^{\mathfrak{S}}$ as follows. First, we disable all actions α in a state $s \in E$ satisfying $\alpha \neq \mathfrak{S}(s)$. By our assumption that each state leaves (E, A) with probability one under $\mathfrak{S}$, and (E, A) is a maximal end component, it follows that no state in E is included in a proper end component after removing these actions. Now we construct the *goal*-directed quotient of maximal end components of the resulting MDP (see Section 2.2.2 for the definition). An example of this construction is given in Figure 4.4. In the following we will identify states $\{s\}$ of $\mathcal{N}^{\mathfrak{S}}$ (which correspond to singleton maximal end components) with s.

To compute $\mathbf{u}_{ev}(t, \beta)$ we can enumerate all internal schedulers $\mathfrak{S}$ of (E, A), compute the maximal expected number of visits of (t, β) in $\mathcal{N}^{\mathfrak{S}}$ and take the maximum of these values.

Lemma 4.38. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form. For all enabled state-action pairs (t, β) of $\mathcal{M}$ we have*

$$\mathbf{u}_{ev}(t, \beta) \quad = \quad \max_{\mathfrak{S}} \ \mathbb{E}^{\max}_{\mathcal{N}^{\mathfrak{S}}}(\diamondplus\{\text{target}, \text{exit}\}),$$

where $\mathfrak{S}$ ranges over all internal schedulers of the maximal end component of t and the reward function assigns one to (t, β) and zero to all other state-action pairs.

Proof. The proof proceeds by first showing (1) $\mathbf{u}_{ev}(t, \beta) \leq \max_{\mathfrak{S}} \ \mathbb{E}^{\max}_{\mathcal{N}^{\mathfrak{S}}}(\diamondplus\{\text{target}, \text{exit}\})$ and then (2) $\mathbf{u}_{ev}(t, \beta) \geq \max_{\mathfrak{S}} \ \mathbb{E}^{\max}_{\mathcal{N}^{\mathfrak{S}}}(\diamondplus\{\text{target}, \text{exit}\})$.

(1) Let $\mathfrak{S}_1$ be a memoryless deterministic scheduler of $\mathcal{M}$ satisfying $\Pr^{\mathfrak{S}_1}_{\mathcal{M}}(\Diamond\{\text{target}, \text{exit}\}) = 1$. First, we construct a scheduler $\mathfrak{S}_2$ for $\mathcal{M}$ such that $\mathbf{ev}^{\mathfrak{S}_2}(t, \beta) \geq \mathbf{ev}^{\mathfrak{S}_1}(t, \beta)$ and for every maximal end component of $\mathcal{M}$ excluding the one which contains t, there is a unique state in which $\mathfrak{S}_2$ chooses an external action.

To this end, consider a maximal end component (E, A) of $\mathcal{M}$ such that $t \notin E$ and $\mathfrak{S}_1$ chooses external actions for multiple states in E. We choose a state $s \in E$ such that $\mathfrak{S}_1(s)$ is external and

$$\sum_{s' \in S} P(s, \mathfrak{S}_1(s), s') \cdot \mathbb{E}^{\mathfrak{S}_1}_{\mathcal{M},s'}(\diamondplus\{\text{target}, \text{exit}\})$$

is maximal among all states in E for which $\mathfrak{S}_1$ chooses an external action. We can adapt $\mathfrak{S}_1$ such that inside the end component (E, A) it chooses internal actions in all states apart from s, while maintaining the property that $\mathfrak{S}_1$ leaves E with probability one from all states in E. The expected number of times (t, β) is visited after this transformation is at least as high as before, as $t \notin E$. Applying this to all maximal end components (E, A) such that $t \notin E$ yields the scheduler $\mathfrak{S}_2$.

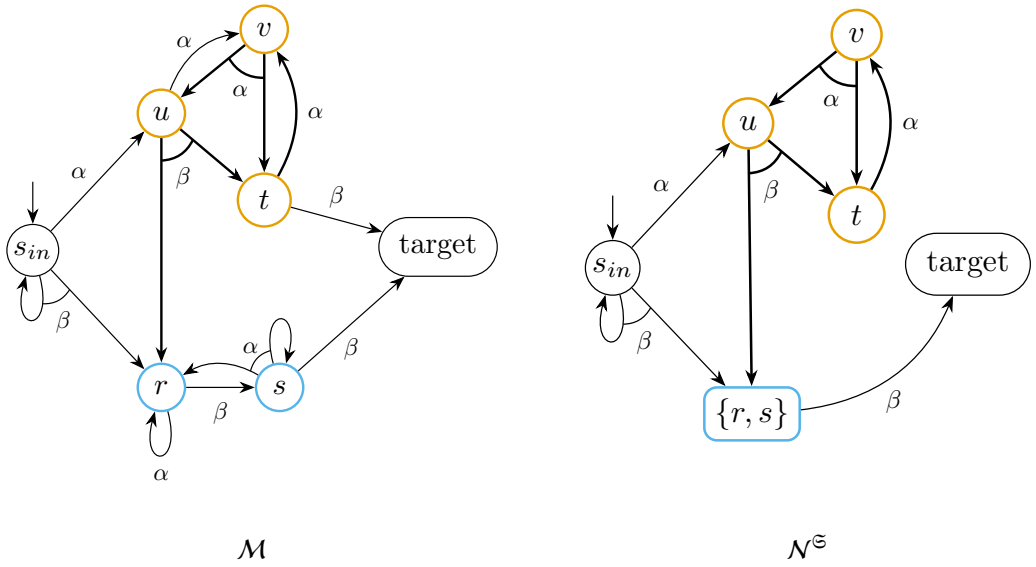


Figure 4.4: An MDP $\mathcal{M}$ with two MECs indicated by the colors. Concrete probabilities are omitted. The bold transitions define an internal scheduler $\mathfrak{S}$ for the upper MEC. The corresponding MDP $\mathcal{N}^{\mathfrak{S}}$ excludes all actions *not* chosen by $\mathfrak{S}$ in the upper MEC, and collapses all other MECs, as in the standard quotient construction. The MDP $\mathcal{N}^{\mathfrak{S}}$ is EC-free.

Now let (E, A) be the maximal end component which includes t and let $\mathfrak{S}$ be the internal scheduler of (E, A) one gets by restricting $\mathfrak{S}_2$ to its choices inside (E, A). The choices of external actions by $\mathfrak{S}_2$ in all other MECs (which are unique due to the above transformation) induce a memoryless deterministic scheduler $\mathfrak{S}_3$ of $\mathcal{N}^{\mathfrak{S}}$ such that

$$\mathbf{ev}_{\mathcal{M}}^{\mathfrak{S}_2}(t, \beta) \quad = \quad \mathbb{E}_{\mathcal{N}^{\mathfrak{S}}}^{\mathfrak{S}_3}(\diamondsuit\{\text{target}, \text{exit}\}),$$

which concludes the proof of this case.

(2) Let (E, A) be the maximal end component which includes t, let $\mathfrak{S}$ be an internal scheduler of (E, A) and $\mathfrak{S}_1$ be a memoryless deterministic scheduler of $\mathcal{N}^{\mathfrak{S}}$. We can restrict ourselves to such schedulers by Proposition 2.14. Consider the scheduler $\mathfrak{S}_2$ for $\mathcal{M}$ which one gets as follows. For states in E, $\mathfrak{S}_2$ copies the choice of $\mathfrak{S}$. In all other maximal end components $\mathfrak{S}_2$ copies $\mathfrak{S}_1$ if the corresponding external action is available in the current state, and otherwise chooses internal actions (in a memoryless and deterministic way) which make sure that the corresponding external state is reached with probability one. Then $\mathfrak{S}_2$ is memoryless and deterministic, and the expected number of visits to (t, β) in $\mathcal{M}$ under $\mathfrak{S}_2$ is $\mathbb{E}_{\mathcal{N}^{\mathfrak{S}}}^{\mathfrak{S}_1}(\diamondsuit\{\text{target}, \text{exit}\})$. □

The number of internal schedulers (which are deterministic and memoryless by definition) of a maximal end component (E, A) is bounded by $|\text{Act}|^{|E|}$. Hence the above lemma yields a polynomial time procedure to compute $\mathbf{u}_{ev}$ if the number of states in any end component is bounded from above by a constant.

Proposition 4.39. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form and let K be an upper bound on the number of states in any end component of $\mathcal{M}$. Then, $\mathbf{u}_{ev}$ can be computed in time* $\text{poly}(|\mathcal{M}|) \cdot |\text{Act}|^K$.

Algorithm 1: k-step quotient sum heuristic ($\mathrm{QS}_k(\mathbf{M}, \mathbf{b})$)

Input: Matrix $\mathbf{M} \in \mathbb{Q}^{m \times n}$, vector $\mathbf{b} \in \mathbb{Q}^m$, natural number k.
Output: Solution of $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$ with small support.
`/* Check whether` $\mathbf{Mx} \geq \mathbf{b}$ `has a nonnegative solution. */`
1 **if** $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$ *is unsatisfiable* **then return**;
`/* Initial objective function. */`
2 $\mathbf{o}_1 := (1, \ldots, 1)$;
3 **for** $i = 1$ **to** k **do**
`/* Find optimal solution of LP under current objective function. */`
4 $\mathrm{LP}_i := \min \mathbf{o}_i \cdot \mathbf{x}$ such that $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$;
5 $\mathbf{x}_i :=$ solve_lp(LP_i);
`/* Define large constant` C`. */`
6 Choose C such that $C > \max\{1/\mathbf{x}_i(j) \mid 1 \leq j \leq n,\ \mathbf{x}_i(j) > 0\}$;
`/* Update objective function using previous solution. */`
7 **for** $j = 1$ **to** n **do**
8 **if** $\mathbf{x}(j) > 0$ **then** $\mathbf{o}_{i+1}(j) = 1/\mathbf{x}_i(j)$ **else** $\mathbf{o}_{i+1}(j) = C$;
9 **end**
10 **end**
11 **return** $\mathbf{x}_k$

If the MDP one considers is EC-free, then the above procedure boils down to simply computing the maximal expected total reward in the MDP under the reward function which is one for (t, β), and zero otherwise. This would require solving a linear program for each pair $(t, \beta) \in \mathcal{E}$. An upper bound which holds for all state-action pairs is given by the maximal expected number of steps taken in $\mathcal{M}$ before reaching $\{\text{target}, \text{exit}\}$. Computing this value only requires solving a single linear program. For Markov chains, $\mathbf{u}_{ev}$ corresponds to the expected number of visits of the unique scheduler, and can hence be computed precisely by solving a single linear equation system.

4.2.3 A heuristic based on linear programming

So far we have considered *exact* methods to compute minimal witnessing subsystems, i.e., methods which are guaranteed to find a minimal solution. As the corresponding decision problem is NP-complete (see Theorem 4.16) already in the case of acyclic Markov chains, we cannot expect to find efficient algorithms for it. In this section, we present a heuristic which is based on iteratively solving a sequence of linear programs, whose underlying systems of linear inequalities are the ones defining Farkas certificates.

The main tool is again Theorem 4.23, which relates Farkas certificates with small support to witnessing subsystems with few states. We present a generic LP-based heuristic called the *k-step quotient-sum heuristic* (Algorithm 1), or simply *quotient-sum heuristic*, which aims to find solutions of a given set of linear inequalities with many zeros. It takes as input a system of linear inequalities described by matrix $\mathbf{M}$ and vector $\mathbf{b}$, and a natural number k which specifies the number of iterations that the algorithm should run. In line 2, the initial objective function coefficients $\mathbf{o}_1$ are defined, which assign equal weight to every variable. Then, in the main loop, the linear program LP_i under the current objective $\mathbf{o}_i$ is defined and solved (lines 4-5). This

linear program finds a vector minimizing $\mathbf{o}_i \cdot \mathbf{x}$ under the condition that $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$ holds.

The produced solution $\mathbf{x}_i$ is used to define the new objective function coefficients $\mathbf{o}_{i+1}$ (line 8). In the update, a large value C is assigned to all variables which already have value zero in the current solution (i.e., for j satisfying $\mathbf{x}_i(j) = 0$). The remaining variables are assigned a new objective value which is inversely proportional to their value in the previous solution. The underlying idea is that if a variable has a small value already, then it should be more likely that a solution exists in which this variable has value equal to zero. A large coefficient in the new objective function $\mathbf{o}_{i+1}$ means that solutions with even smaller value (in best case with value zero) of this variable are preferred in the next iteration. Correctness of the procedure does not depend on the specific choice of C, and a large C is meant to discourage solutions whose support includes a dimension which was already zero in a foregoing iteration.

A useful observation in this context is that for all inputs $\mathbf{M}$ and $\mathbf{b}$, the LPs constructed in Algorithm 1 are not unbounded. This follows from the fact that the objective vector $\mathbf{o}_i$ is always nonnegative in all entries.

Lemma 4.40. *For all $\mathbf{M} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{o} \in \mathbb{R}^n_{\geq 0}$, the linear program*

$$\min \mathbf{o} \cdot \mathbf{x} \quad \textit{such that} \quad \mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$$

either has an optimal solution, or no feasible solution at all.

Proof. Let $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n_{\geq 0} \mid \mathbf{Mx} \geq \mathbf{b}\}$ and suppose that there exists $\mathbf{p}_0, \mathbf{p}_1 \in \mathbb{R}^n$ such that for all $t \geq 0$ we have $\mathbf{p}_0 + t\mathbf{p}_1 \in \mathcal{P}$. It follows that $\mathbf{p}_0, \mathbf{p}_1$ are nonnegative. But then, as $\mathbf{o}$ is also nonnegative, for all $t, t' \geq 0$ with $t \leq t'$ we have $\mathbf{o}(\mathbf{p}_0 + t\mathbf{p}_1) \leq \mathbf{o}(\mathbf{p}_0 + t'\mathbf{p}_1)$, and thus the objective value *increases* along the line defined by $\mathbf{p}_0, \mathbf{p}_1$ with growing t. Hence, there is no infinite line through $\mathcal{P}$ with arbitrary low objective value in the linear program. □

As the vectors $\mathbf{x}_i$ produced in line 5 are always nonnegative, it follows that the coefficient vector $\mathbf{o}_i$ remains nonnegative throughout Algorithm 1. Hence, the above lemma applies to all linear programs LP_i, which means that an optimal solution is always found in line 4.

Accounting for labels. The quotient-sum heuristic can be adapted to also take labels into account, as shown in Algorithm 2. Here, a labeling function $\Lambda : \{1, \ldots, n\} \to 2^L$ into a finite set of labels L is also given as input. An additional vector σ of variables is introduced in the constructed linear programs, with the constraint that $\mathbf{x}(i) \leq \sigma(l)$ should hold for all $i \in \{1, \ldots, n\}$ and $l \in \Lambda(i)$. The objective function now minimizes a weighted sum over σ, where the weights are again adapted in each iteration. This implies that in any optimal solution, $\sigma(l)$ will be zero exactly if all variables $\mathbf{x}(i)$ which satisfy $l \in \Lambda(i)$ have value zero.

Applying the heuristic to the witness problems

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form. Algorithms 1 and 2 can be applied to find small witnessing subsystems for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda$ and $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda$ as follows. For minimal reachability probabilities, we first construct the system of linear inequalities defining the set of Farkas certificates $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ (for simplicity, we will also use $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ to describe the linear inequalities). Applying Algorithm 1 (with arbitrary natural number k) yields a vector $\mathbf{z} = \mathrm{QS}_k(\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda))$ which satisfies $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ and, by Theorem 4.23, the subsystem $\mathcal{M}_{\mathrm{supp}(\mathbf{z})}$ induced by the non-zero entries of $\mathbf{z}$ is a witnessing subsystem for

Algorithm 2: labeled k-step quotient sum heuristic ($\mathrm{QS}_k(\mathbf{M}, \mathbf{b}, \Lambda)$)

Input: Matrix $\mathbf{M} \in \mathbb{Q}^{m \times n}$, vector $\mathbf{b} \in \mathbb{Q}^m$, natural number k, labeling function $\Lambda : [n] \to 2^L$.

Output: Solution of $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$ which hits few labels.

```
/* initial objective (one coefficient per label in L)                    */
```

1 $\mathbf{o}_1 := (1, \ldots, 1) \in \mathbb{Q}^L$;
2 **for** $i = 1$ **to** k **do**
3 $\quad \mathrm{LP}_i := \min \mathbf{o}_i \cdot \sigma$ such that $\mathbf{Mx} \geq \mathbf{b} \wedge \mathbf{x} \geq \mathbf{0}$ and $\mathbf{x}(i) \leq \sigma(l)$ for all $i \in \{1, \ldots, n\}$ and $l \in \Lambda(i)$;
4 $\quad (\sigma_i, \mathbf{x}_i) := \mathrm{solve\_lp}(\mathrm{LP}_i)$;
5 $\quad$ Choose C such that $C > \max\{1/\sigma_i(l) \mid l \in L,\ \sigma_i(l) > 0\}$;
6 $\quad$ **for** $l \in L$ **do**
7 $\quad\quad$ **if** $\sigma_i(l) > 0$ **then** $\mathbf{o}_{i+1}(l) = 1/\sigma_i(l)$ **else** $\mathbf{o}_{i+1}(l) = C$;
8 $\quad$ **end**
9 **end**
10 **return** $\mathbf{x}_k$

$\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$. In the case of maximal reachability probabilities, we can apply the heuristic in analogous fashion. Recall that vectors in $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ have dimension $|\mathcal{E}|$, which is the number of enabled state-action pairs of $\mathcal{M}$. Again, for any natural number k, the result of applying the quotient-sum heuristic $\mathbf{y} = \mathrm{QS}_k(\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda))$ satisfies $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ and $\mathcal{M}_{\text{state-supp}(\mathbf{y})}$ is a witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$, by Theorem 4.23.

Recall that $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ is unsatisfiable iff $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) < \lambda$ holds by Theorem 3.24. Hence, $\mathrm{QS}_k(\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda))$ returns no solution (i.e., it returns in line 1) iff $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) < \lambda$ holds, and the analogous statement holds for the maximal reachability probability. As a consequence, $\mathrm{QS}_k(\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda))$ can also be viewed as a *counterexample generating model checking procedure* for the property $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) < \lambda$. If it holds, the algorithm returns nothing in line 1, and otherwise a vector is returned from which a (hopefully small) witnessing subsystem for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$ can be produced. In any case, the returned vector is a Farkas certificate which proves that the property is violated. This algorithm *does not* produce a certificate in case the property is satisfied; this would require solving another linear program to find a Farkas certificate for $\mathbf{Pr}^{\min}(\Diamond \text{target}) \geq \lambda$ (see Definition 3.23).

We say that the heuristic stabilizes after iteration i if $\mathrm{QS}_i\big(\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)\big) = \mathrm{QS}_{i+1}\big(\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)\big)$ holds. Usually this happens already after two-five iterations and sometimes the solution after the first iteration already induces a very small witnessing subsystem. See Section 4.2.5 for further information and the corresponding experimental results.

It should be pointed out that the heuristics as described above do not depend on the MDP being EC-free (although the maximal end components have to be precomputed to define $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, see Definition 3.23). In particular, no upper bound on $\mathbf{u}_{ev}$ (defined in Equation (4.1)) is required, as compared to the MILP formulations to compute *minimal* witnessing subsystems for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$.

The quality of the heuristic is measured by how many non-zero entries the returned solution has, respectively how many labels are induced by its support. An extensive evaluation of the heuristic when applied to the witness problems is given in Section 4.2.5, where alternative initial

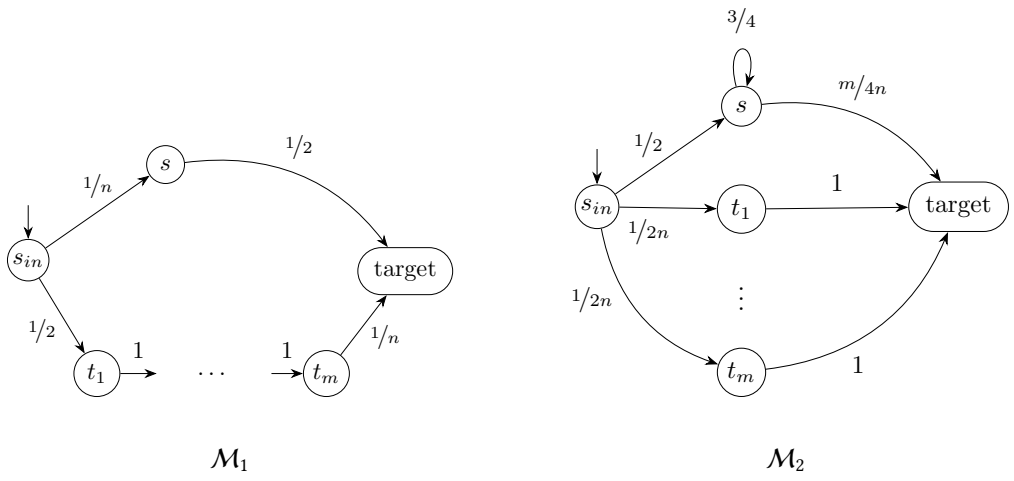


Figure 4.5: Markov chains for which the quotient-sum heuristic runs into a local optimum. Here, n and m are natural numbers, and we assume that $6 \leq 2m < n$ holds. For all states the missing probabilities are added to a transition to "exit", which is not drawn.

objective functions (rather than just taking $\mathbf{o}_1 = (1, \ldots, 1)$) are also considered. In the following, the limits of this approach are discussed by studying certain constructed instances.

Limitations of the heuristic

Unsurprisingly, the heuristic may get stuck in local optima. We will now consider in more detail how the heuristic works for the two Markov chains depicted in Figure 4.5. These Markov chains can also be used to show that the two possible instantiations of the quotient-sum heuristic for Markov chains (using $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ and $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, respectively) may yield results which are arbitrarily far away from one another.

Consider the Markov chain $\mathcal{M}_1$ depicted in Figure 4.5, where n, m are natural numbers satisfying $6 \leq 2m < n$. Let $\mathbf{A}, \mathbf{t}$ be its system matrix and target-vector, and let $\lambda = 1/2n$. The system of linear inequalities $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$, which defines the set of Farkas certificates $\mathcal{F}^{\min}_{\mathcal{M}_1,\geq}(\lambda)$, is given by

$$\begin{aligned} &\mathbf{z}(s_{in}) \leq 1/n \cdot \mathbf{z}(s), && \mathbf{z}(s_{in}) \leq 1/2 \cdot \mathbf{z}(t_1), \qquad \mathbf{z}(s) \leq 1/2, \qquad \mathbf{z}(s_{in}) \geq \lambda \\ &\mathbf{z}(t_m) \leq 1/n, && \mathbf{z}(t_i) \leq \mathbf{z}(t_{i+1}) \quad \text{for all } i \in \{1, \ldots, m-1\}. \end{aligned}$$

The first iteration of the quotient-sum heuristic instantiated by this system of linear inequalities computes a solution of the linear program: *minimize* $(1, \ldots, 1) \cdot \mathbf{z}$ *such that* $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$. This optimal solution is given by the vector

$$\mathbf{z}_1 = \big(s_{in} \mapsto 1/2n,\ s \mapsto 0,\ t_1 \mapsto 1/n,\ \ldots,\ t_m \mapsto 1/n \big).$$

The corresponding witnessing subsystem is the one induced by $\mathrm{supp}(\mathbf{z}_1) = \{s_{in}, t_1, \ldots, t_m\}$, and hence has $m + 1$ states. Another feasible solution of the LP is

$$\mathbf{z}_2 = \big(s_{in} \mapsto 1/2n,\ s \mapsto 1/2,\ t_1 \mapsto 0,\ \ldots,\ t_m \mapsto 0 \big).$$

It corresponds to the witnessing subsystem induced by $\{s_{in}, s\}$, which is arbitrarily smaller than the previous witness with growing m. However, comparing the objective values of vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ in the linear program above yields:

$$(1,\ldots,1)\cdot\mathbf{z}_1 \;=\; {}^{m}\!/_{n} + {}^{1}\!/_{2n} \;<\; {}^{1}\!/_{2} + {}^{1}\!/_{2n} \;=\; (1,\ldots,1)\cdot\mathbf{z}_2.$$

Here we used our assumption that $2m < n$. This means that in the first iteration of the heuristic $\mathrm{QS}_k\big(\mathcal{F}^{\min}_{\mathcal{M}_1,\geq}(\lambda)\big)$, the vector $\mathbf{z}_1$ will be preferred over $\mathbf{z}_2$ even though $\mathbf{z}_2$ induces a smaller witnessing subsystem. In the second iteration, the objective function is updated to be the point-wise inverse of $\mathbf{z}_1$, with the exception of the coefficient of s, which is assigned a large number C (we assume $C > 2m$ in the following):

$$\mathbf{o}_2 = \big(\, s_{in} \mapsto 2n,\ s \mapsto C,\ t_1 \mapsto n,\ \ldots,\ t_m \mapsto n \,\big).$$

Under this objective function again the feasible solution $\mathbf{z}_1$ is preferred over $\mathbf{z}_2$, as $\mathbf{o}_2 \cdot \mathbf{z}_1 = 1 + m < 1 + {}^{C}\!/_{2} = \mathbf{o}_2 \cdot \mathbf{z}_2$. One can check that again $\mathbf{z}_1$ is the optimal solution of the new LP, and hence the heuristic repeats itself and further iterations do not yield any improvement. Consequently, the witnessing subsystem computed by $\mathrm{QS}_k\big(\mathcal{F}^{\min}_{\mathcal{M}_1,\geq}(\lambda)\big)$ is the one induced by $\{s, t_1, \ldots, t_m\}$ for any $k \geq 1$.

As $\mathcal{M}_1$ is a Markov chain, we can also use the algorithm $\mathrm{QS}_k\big(\mathcal{F}^{\max}_{\mathcal{M}_1,\geq}(\lambda)\big)$ to compute witnesses for $\Pr_{\mathcal{M}_1}(\Diamond\,\mathrm{target}) \geq \lambda$. In this case, the first iteration solves the linear program: *minimize* $\mathbf{y}\cdot(1,\ldots,1)^T$ *such that* $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$. Spelling out the underlying system of linear inequalities yields:

$$\begin{array}{lll} \mathbf{y}(s_{in}) \leq 1, & \mathbf{y}(s) \leq {}^{1}\!/_{n}\cdot\mathbf{y}(s_{in}), & {}^{1}\!/_{n}\cdot\mathbf{y}(t_m) + {}^{1}\!/_{2}\cdot\mathbf{y}(s) \geq \lambda, \\ \mathbf{y}(t_1) \leq {}^{1}\!/_{2}\cdot\mathbf{y}(s_{in}), \text{ and} & \mathbf{y}(t_{i+1}) \leq \mathbf{y}(t_i) \ \text{ for all } i \in \{1,\ldots,m{-}1\}. & \end{array}$$

The minimal solution vector of the above LP is

$$\big(\, s_{in} \mapsto 1,\ s \mapsto {}^{1}\!/_{n},\ t_1 \mapsto 0,\ \ldots,\ t_m \mapsto 0 \,\big),$$

which corresponds to the minimal witnessing subsystem induced by $\{s_{in}, s\}$. Hence, for the Markov chain $\mathcal{M}_1$ using $\mathrm{QS}_k\big(\mathcal{F}^{\max}_{\mathcal{M}_1,\geq}(\lambda)\big)$ yields the optimal solution already after one iteration, while $\mathrm{QS}_k\big(\mathcal{F}^{\min}_{\mathcal{M}_1,\geq}(\lambda)\big)$ is far away from the optimum for any number of iterations k.

The Markov chain $\mathcal{M}_2$ (also shown in Figure 4.5) represents the opposite situation. Let $\mathbf{A}, \mathbf{t}$ be the system matrix and target vector of $\mathcal{M}_2$ and define $\lambda = {}^{m}\!/_{2n}$, again under the assumption that $6 \leq 2m < n$ holds. The system of linear inequalities $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$ used in $\mathrm{QS}_k\big(\mathcal{F}^{\max}_{\mathcal{M}_2,\geq}(\lambda)\big)$ is:

$$\begin{array}{lll} \mathbf{y}(s_{in}) \leq 1, & & {}^{m}\!/_{4n}\cdot\mathbf{y}(s) + \displaystyle\sum_{1\leq i\leq m}\mathbf{y}(t_i) \geq \lambda \\ \mathbf{y}(s) \leq {}^{1}\!/_{2}\cdot\mathbf{y}(s_{in}) + {}^{3}\!/_{4}\cdot\mathbf{y}(s), & \text{and} & \mathbf{y}(t_i) \leq {}^{1}\!/_{2n}\cdot\mathbf{y}(s_{in}) \text{ for all } i \in \{1,\ldots,m\}. \end{array}$$

The minimal solution vector of the linear program *minimize* $\mathbf{y}\cdot(1,\ldots,1)^T$ *such that* $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$ is given by:

$$\mathbf{y}_1 = \big(\, s_{in} \mapsto 1,\ s \mapsto 0,\ t_1 \mapsto {}^{1}\!/_{2n},\ \ldots,\ t_m \mapsto {}^{1}\!/_{2n} \,\big).$$

It corresponds to the witnessing subsystem induced by $\{s_{in}, t_1, \ldots, t_m\}$. A Farkas certificate with smaller support is

$$\mathbf{y}_2 = \big(s_{in} \mapsto 1,\ s \mapsto 2,\ t_1 \mapsto 0,\ \ldots,\ t_m \mapsto 0 \big).$$

However, as in the other example, the objective value of vector $\mathbf{y}_1$ (which is $1 + m/2n$) is less than the objective value of vector $\mathbf{y}_2$ (which is 3), given our assumption that $2m < n$ holds. Similarly to the previous example, one finds that after the first iteration $\mathrm{QS}_k\big(\mathcal{F}^{\max}_{\mathcal{M}_{2,\geq}}(\lambda)\big)$ repeats itself.

Now consider the heuristic $\mathrm{QS}_k\big(\mathcal{F}^{\min}_{\mathcal{M}_{2,\geq}}(\lambda)\big)$ based on the system of linear inequalities $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$. When spelled out, this yields:

$$\mathbf{z}(s_{in}) \leq 1/2 \cdot \mathbf{z}(s) + \sum_{1 \leq i \leq m} 1/2n \cdot t_i, \qquad \mathbf{z}(s_{in}) \geq \lambda$$
$$\mathbf{z}(s) \leq 3/4 \cdot \mathbf{z}(s) + m/4n, \qquad \text{and} \qquad \mathbf{z}(t_i) \leq 1 \text{ for all } i \in \{1, \ldots, m\}.$$

A minimal vector of the linear program *minimize* $(1, \ldots, 1) \cdot \mathbf{z}$ *such that* $\mathbf{Az} \leq \mathbf{t} \wedge \mathbf{z}(s_{in}) \geq \lambda$ is given by:

$$\big(s_{in} \mapsto m/2n,\ s \mapsto m/n,\ t_1 \mapsto 0,\ \ldots,\ t_m \mapsto 0 \big),$$

and this vector corresponds to the minimal witness induced by $\{s_{in}, s\}$.

This shows that the two heuristics $\mathrm{QS}_k\big(\mathcal{F}^{\max}_{\mathcal{M}_{2,\geq}}(\lambda)\big)$ and $\mathrm{QS}_k\big(\mathcal{F}^{\min}_{\mathcal{M}_{2,\geq}}(\lambda)\big)$ based on the two different sets of Farkas certificates can be arbitrarily better than the other one in the case of Markov chains.

4.2.4 The tool Switss

Switss[1] is a tool which implements a large part of the algorithms presented in this thesis. It is written in python and uses the PuLP[2] library to model linear optimization problems. This enables easily interfacing with various mathematical optimization solvers in the back end, such as Gurobi [Gur22], Cplex[3], Cbc[4] and GLPK[5]. Alternatively, Switss also allows using Gurobi's python interface directly.

The tool includes algorithms for the exact computation of minimal witnessing subsystems using the MILP-based approaches described in this thesis, an implementation of the quotient-sum heuristic and a module to compute and validate Farkas certificates. Additionally, it provides methods to visualize MDPs and MDP-subsystems and to run and evaluate benchmarks. What has not been implemented is the computation of small and minimal witnessing subsystems for invariants (see Section 4.4), and the algorithm to compute upper bounds on $\mathbf{u}_{ev}$ for MDPs which are not EC-free but have only small end components (see Lemma 4.38). Whenever no upper bound is given and the MDP is not EC-free, we fall back to a formulation using indicator constraints (see Remark 4.34).

[1] `https://www.github.com/simonjantsch/switss`
[2] `https://coin-or.github.io/pulp/`
[3] `https://www.ibm.com/analytics/cplex-optimizer`
[4] `https://projects.coin-or.org/Cbc`
[5] `https://www.gnu.org/software/glpk/`

4.2.5 Experimental results

This section reports on a number of experiments which have been performed using Switss. All computations were run on a computer with two Intel E5-2680 processors having 8 cores each at 2.70 GHz running Linux, with a total of 378 GBs of RAM. If not specified otherwise, each computation was assigned 4 cores and ran under a timeout of 20 minutes and a memory out of 30 GBs. We have configured Switss to use Gurobi version 9.5 [Gur22] to solve the underlying LPs and MILPs. However, the experiments are set up such that one can easily reproduce them using the open-source solver Cbc (although this might lead to different results, in particular with respect to the computation times). All experimental data, together with the version of Switss used to produce the data and all scripts used to run the benchmarks and evaluate the raw data are available [Jan22b].

The benchmarks we use are standard benchmarks from the Prism benchmark suite [KNP12]. Most of them are parametrized by two parameters N and K and we will write "protocol-name_N_K" to distinguish different instances. We use Prism to construct sparse matrix representations of the benchmarks. These are loaded into Switss, which converts them into reachability form with respect to the considered reachability properties. Essentially, this is a preprocessing step which ensures that there is a single target state, and identifies all states which cannot reach it with the state "exit". Tables 4.1 and 4.2 report on the sizes and reachability probabilities of the resulting models (after transformation into reachability form). We start with a short description of the benchmarks.

crowds The crowds protocol [RR98] was designed to allow for anonymous usage of the web, accomplished by routing data randomly through other connected devices. It is modeled as a Markov chain in the Prism benchmark suite, and parametrized by N, the number of non-adversarial crowd members, and K, which is the number of protocol runs. The property we consider is to reach a state in which a corrupt crowd member directly succeeds the original sender more than once.

brp The bounded retransmission protocol is designed to transmit a file consisting of N chunks through an unreliable channel [HSV93]. Each chunk is retransmitted at most K times. It is modeled as a Markov chain, and the property we consider is to reach a state in which the receiver reports an uncertainty on the success of the transmission.

leader Given a ring of N processes, the synchronous leader election protocol enables jointly electing a unique leader among the processes [IR90]. In every round, each process draws a number uniformly from the range $\{1, \ldots, K\}$, where K is another parameter of the protocol. If some number is drawn by exactly one process, then the process with the highest number satisfying this criterion is selected to be the leader. Otherwise, a new round begins. This protocol is modeled as a Markov chain, and the property we consider is to reach a state in which a leader has been elected successfully.

Table 4.1: Properties of Markov chain benchmarks, after transformation into reachability form. Probabilities are rounded to three decimal places.

	crowds			brp			leader
	(2,6)	(2,8)	(5,8)	(16,2)	(32,2)	(1024,2)	(6,6)
states	434	832	27 847	499	995	31 747	234 210
Pr(◊ target)	0.375	0.532	0.310	2.645×10^{-5}	2.644×10^{-5}	2.576×10^{-5}	1.000

Table 4.2: Properties of MDP benchmarks, after transformation into reachability form. The minimal and maximal probabilities of reaching "target" are one in all MDP benchmarks.

	cons			firewire		csma	
	(2,4)	(4,2)	(4,4)	3	30	(2,6)	(3,2)
states	528	22 656	22 656	4093	138 130	66 718	36 850
state-action pairs	784	60 544	60 544	5519	302 654	66 788	38 456

consensus This benchmark is an MDP model of the randomized consensus algorithm described in [AH90]. The goal of this protocol is to form consensus quickly between N processes on one out of two possible outcome values. The parameter K is any number above one and is a technical ingredient of the protocol. We consider the property of reaching a state in which the protocol terminates.

csma The CSMA/CD protocol is a network protocol for transmitting messages in a local area network which uses collision detection. The parameter N indicates the number of stations in the network and K is a technical parameter which determines how long a station waits on average until attempting to retransmit, in case that a collision was detected. We consider the property of reaching a state in which all stations delivered their messages.

firewire This benchmark models the Tree Identity Protocol of the IEEE1394 High Performance Serial Bus, which is also called FireWire. It is a leader election protocol which is run whenever a new member joins the network. The parameter N determines a delay, i.e., the time needed for a message to be transmitted between members of the network. We consider the property of reaching a state in which a leader was elected successfully.

Computing minimal witnesses

To compute minimal witnesses (we will focus on state-minimality) we have proposed two formulations based on mixed-integer linear programs, one for minimal probabilities (Definition 4.28), and one for maximal reachability probabilities (Definition 4.32).

Table 4.3 shows selected experimental results obtained by solving the max- and min-witness programs for several benchmarks, including both Markov chains and Markov decision processes. For Markov chains, the min-witness and the max-witness programs can both be used, and as

Table 4.3: Experiments on computing minimal witnessing subsystems using the MILP-based approaches for different benchmarks and thresholds. The table entries contain the running times in seconds required to solve the problem and the number of states of the computed subsystem in brackets. Timeouts (configured to 20 minutes) are indicated by "TO". The rows named ltlsubsys present the results of solving the alternative MILP-formulation for maximal reachability probabilities proposed in [WJÁ+14].

	$\lambda =$	0.05	0.11	0.33	0.4
crowds_2_8	min-MILP	< 1 (29)	3.0 (57)	516.1 (191)	TO (-)
	max-MILP	2.4 (29)	3.0 (57)	294.2 (191)	TO (-)
	$\lambda =$	2×10^{-6}	8×10^{-6}	1×10^{-5}	1.2×10^{-5}
brp_32_2	min-MILP	2.5 (196)	4.0 (215)	1.4 (218)	TO (-)
	max-MILP	< 1 (196)	< 1 (215)	< 1 (218)	TO (-)
	$\lambda =$	0.1	0.3	0.6	0.9
consensus_2_4	min-MILP	13.6 (166)	4.9 (233)	1.0 (308)	1.4 (420)
	max-MILP	TO (-)	TO (-)	TO (-)	TO (-)
	ltlsubsys (max)	TO (-)	TO (-)	TO (-)	TO (-)
	$\lambda =$	0.1	0.3	0.6	0.9
firewire_3	min-MILP	2.0 (240)	2.2 (479)	2.5 (1619)	1.8 (4093)
	max-MILP	10.2 (85)	TO (-)	TO (-)	TO (-)
	ltlsubsys (max)	5.5 (85)	903.1 (248)	TO (-)	TO (-)

expected, they always return the same size for minimal witnesses. All MDP benchmarks are EC-free, and the required upper bound on $\mathbf{u}_{ev}$ (see Definition 4.32) is computed by solving the linear program:

$$\textit{maximize} \sum_{(s,\alpha)\in\mathcal{E}} \mathbf{y}(s,\alpha) \quad \textit{such that} \quad \mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{y} \geq \mathbf{0}.$$

This linear program is bounded by Proposition 3.9 for EC-free MDPs, and its optimal value is an upper bound on all entries of $\mathbf{u}_{ev}$. For Markov chains, the vector containing the expected number of visits per state-action pair yields such an upper bound and can be obtained by solving a linear equation system. See Section 4.2.2 for details on computing upper bounds on $\mathbf{u}_{ev}$.

We want to highlight the following observations. First, the computation times may vary substantially with the threshold λ. In particular, for brp_32_2, we can see that a very small increase in the threshold may change the computation times from under five seconds to running into the timeout of 20 minutes. Second, in the case of MDPs, the min- and max-witness programs may behave very differently. It appears from these experiments that the min-witness MILP is generally easier to solve. Third, the MILP proposed in [WJÁ+14], which computes minimal witnesses for lower bounds on the maximal reachability probability, performs a bit better than the max-witness MILP in these experiments, as it solves the firewire_3 benchmark for threshold 0.1.

The quotient-sum heuristic

We now turn to an evaluation of the quotient-sum heuristic, which was introduced in Section 4.2.3. Figure 4.6 shows the results of applying it to two Markov chain benchmarks. As observed before, for Markov chains we can instantiate the heuristic with either the inequalities defining $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ (plotted on the left in Figure 4.6), or those defining $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ (plotted on the right in Figure 4.6). The plot $\mathfrak{m}$-QSHeur$_i$ shows the size of the subsystem returned after i iterations of the quotient-sum heuristic for increasing thresholds λ using the system of inequalities $\mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\geq}(\lambda)$. We can see that while the formulation using $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ produces smaller subsystems for the benchmark crowds_2_8, the opposite is true (for most λ) in the benchmark brp_1024_2.

The figures shows that running multiple iterations of the heuristic can indeed bring down the size of the produced subsystem substantially. Furthermore, the heuristic tends to stabilize already after very few iterations. In Figure 4.6, the difference between QSHeur$_2$ and QSHeur$_3$ is already very small in most cases.

An interesting phenomenon are the "spikes" in the left plot for crowds_2_8 (using $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$) and the right plot for brp_1024_2 (using $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$). Here, *increasing* the threshold λ (i.e., the lower bound on the probability) may produce subsystems with *fewer* states. This is counterintuitive, as a witnessing subsystem for threshold λ is also witnessing for any $\lambda' \leq \lambda$. Recall that the size of the subsystem corresponds to the number of non-zero entries of the vector produced by the heuristic. Increasing λ implies moving one of the defining linear inequalities further away from zero in the underlying system of linear inequalities.

Consider the spike observed in the range of thresholds between $\lambda = 0.02$ and $\lambda = 0.13$ in Figure 4.6a. The results after two or more iterations (plotted as max-QSHeur$_i$, with $i \geq 2$) mostly increase in this range, but decrease again drastically for $\lambda = 0.15$. If we look at the results of QSHeur$_1$ for the same range (recall that this is the result of the first iteration, which is used to compute the objective function coefficients for the second iteration), we find that they remain almost constant. It appears that the initial iteration guides the search into a region of the polyhedron which is "good" for low thresholds in the range, but not for the larger ones. When the threshold increases to above 0.13, the initial LP yields another solution (it "jumps" in the plot), which changes the objective of the second iteration favorably, yielding a smaller subsystem. From this observation we learn that the "direction" of the initial minimization objective may affect the outcome of the heuristic significantly. Different alternative initial objectives are evaluated later in this section.

Comparison with the exact methods. Figure 4.7 compares the results of the quotient-sum heuristic against the size of a minimal subsystem as computed by the MILP-approach. Here we use the result of the quotient-sum heuristic after three iterations. One can see that in these experiments the result of the better performing instances of the heuristic is very close to the actual minimum.

Comparison with other heuristic approaches. The tool Comics [JÁV+12] also implements heuristics to compute small witnessing subsystems for Markov chains. They are based on algorithms which iteratively include more and more paths to a subsystem (starting with the subsystem including only s_{in}) until the threshold condition is met. This ensures that in the end, the computed subsystem is a witness. To check whether the condition is met, a model checking step is included in each iteration. Two variants of this idea are implemented in Comics:

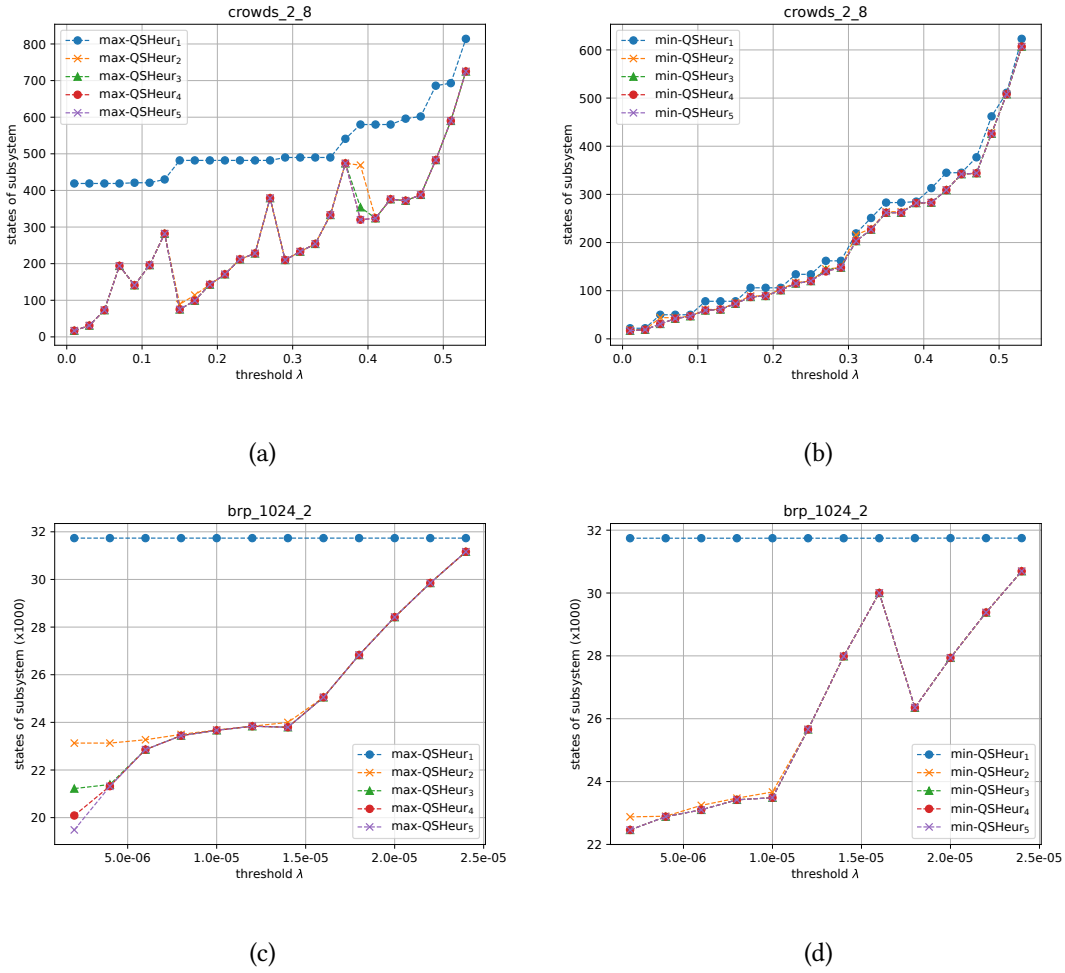


Figure 4.6: Results produced by the quotient-sum heuristic for two Markov chain benchmarks, with increasing thresholds. Two instances of the heuristic are considered: on the left the system of inequalities defining $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ is used, whereas $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ is used on the right. The plot $\mathfrak{m}$-QSHeur$_i$ contains the result produced by iterating the quotient-sum heuristic i times over $\mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\geq}(\lambda)$. The plotted value is the size of the subsystem induced by the computed Farkas certificate (i.e., the size of the support of the certificate).

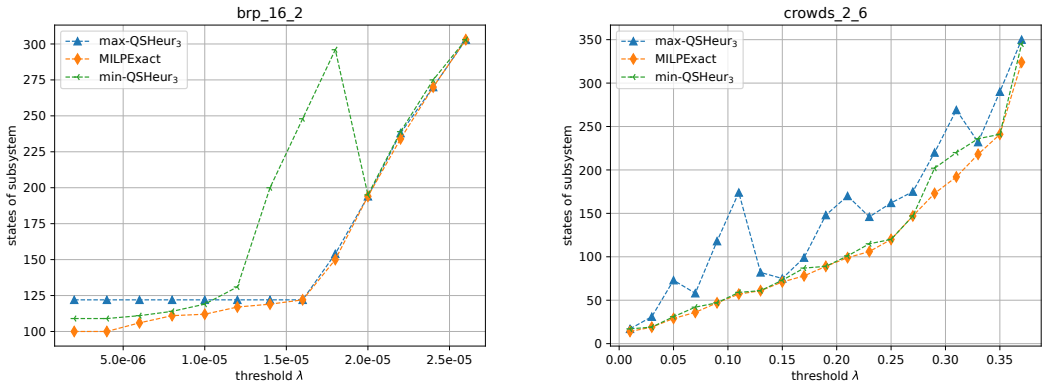


Figure 4.7: Comparison of subsystem sizes computed by the quotient-sum heuristic in three iterations against the minimal witnessing subsystem. Here, $\mathfrak{m}$-QSHeur$_3$ is the quotient-sum heuristic (running for three iterations) which uses the inequalities defining $\mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\geq}(\lambda)$, and MILPExact plots the size of a minimal witness as computed using the MILP approach.

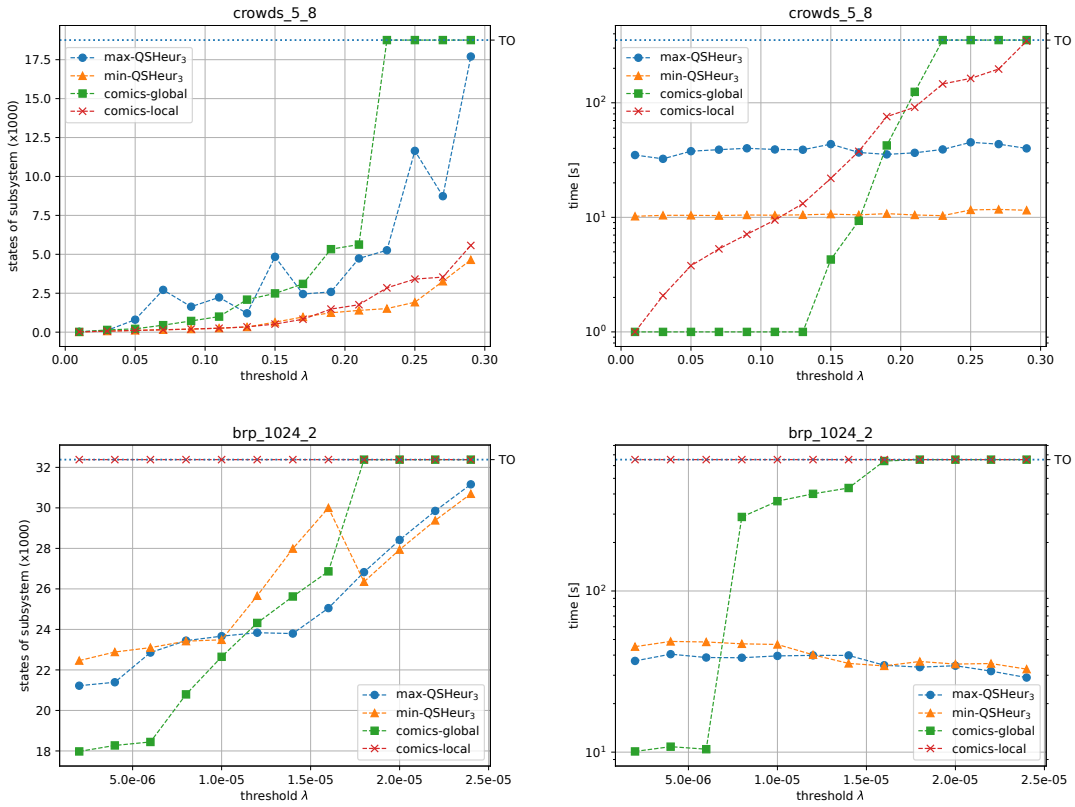


Figure 4.8: Sizes (left) and computation times (right) of witnessing subsystems computed using the quotient-sum heuristic and Comics on two Markov chain benchmarks.

the "global-search" finds preferably short paths to "target" carrying a lot of probability in each iteration, whereas the "local-search" attempts to connect states already included in the current subsystem with few additional states.

Figure 4.8 shows the result of Comics as compared with the quotient-sum heuristic for two different Markov chain benchmarks. On the left, we see the sizes of subsystems returned by the different heuristics for increasing thresholds, and on the right we see the corresponding computation times. The results of the quotient-sum heuristics after three iterations are plotted, using $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ and $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ respectively. For Comics, we use the computation time reported by the tool, and we plot all times under one second on the one-second line.

When considering the sizes of computed subsystems, we can see that the better configuration of the quotient-sum heuristic performs comparably to the better configuration of Comics. The only large difference appears for small thresholds of brp_1024_2, where the "global-search" performs very well in comparison to the other methods. For the computation times, one can see that while the quotient-sum heuristic is not influenced by increasing thresholds, the computation times of Comics increase drastically. While Comics is very fast for small thresholds, it runs into the timeout for thresholds closer to the actual probability of the system. This phenomenon can be explained by the different nature of the algorithms. With increasing thresholds, Comics needs to add more paths to generate a witness, which leads to more iterations. As each iteration includes a model checking phase, it is natural that computation times increase with an increase in the number of iterations. On the other hand, the threshold λ is just a number appearing in the linear programs that the quotient-sum heuristic solves, and does not affect the number of iterations or the time required to solve the LPs in these experiments.

QS-heuristic with alternative initial objectives. In the quotient-sum heuristic (see Algorithm 1) the coefficients $\mathbf{o}_1$ of the initial objective function were chosen to be $(1, \ldots, 1)$. Therefore, the first iteration computes a Farkas certificate with minimal sum-of-entries. This choice is natural, but it is not necessary for the correctness of the procedure. Any nonnegative vector of coefficients can be chosen, and it is easy to adapt the initial objective in Storm.

We now evaluate different choices of initial objectives. The idea is to make coefficients relatively small for states (respectively state-action pairs) which we expect to be part of a minimal witnessing subsystem, i.e., those that are "important" for the property at hand. Coefficients of states which are not important should be relatively high, because this favors solutions in which the value of these states equals zero. We will consider two alternative initial coefficient vectors. For the first one, consider the linear program

$$\textit{maximize} \quad \mathbf{yt} \quad \textit{such that} \quad \mathbf{yA} \leq \delta_{s_{in}}. \tag{4.2}$$

A solution $\mathbf{y}$ of Equation (4.2) corresponds to the expected number of visits of a scheduler that maximizes the probability of reaching target in EC-free MDPs, and hence state-action pairs with a high value in a solution of this LP can be considered important. Given an optimal solution $\mathbf{y}$ to the above linear program, we define $\mathbf{inve} \in \mathbb{Q}^{\mathcal{E}}_{\geq 0}$ by

$$\mathbf{inve}(s,\alpha) = \begin{cases} \frac{1}{\mathbf{y}(s,\alpha)} & \text{if } \mathbf{y}(s,\alpha) > 0 \\ C & \text{otherwise,} \end{cases}$$

where $C > \max\{1/\mathbf{y}(s,\alpha) \mid (s,\alpha) \in \mathcal{E}, \mathbf{y}(s,\alpha) > 0\}$ is a large constant. Second, we consider the

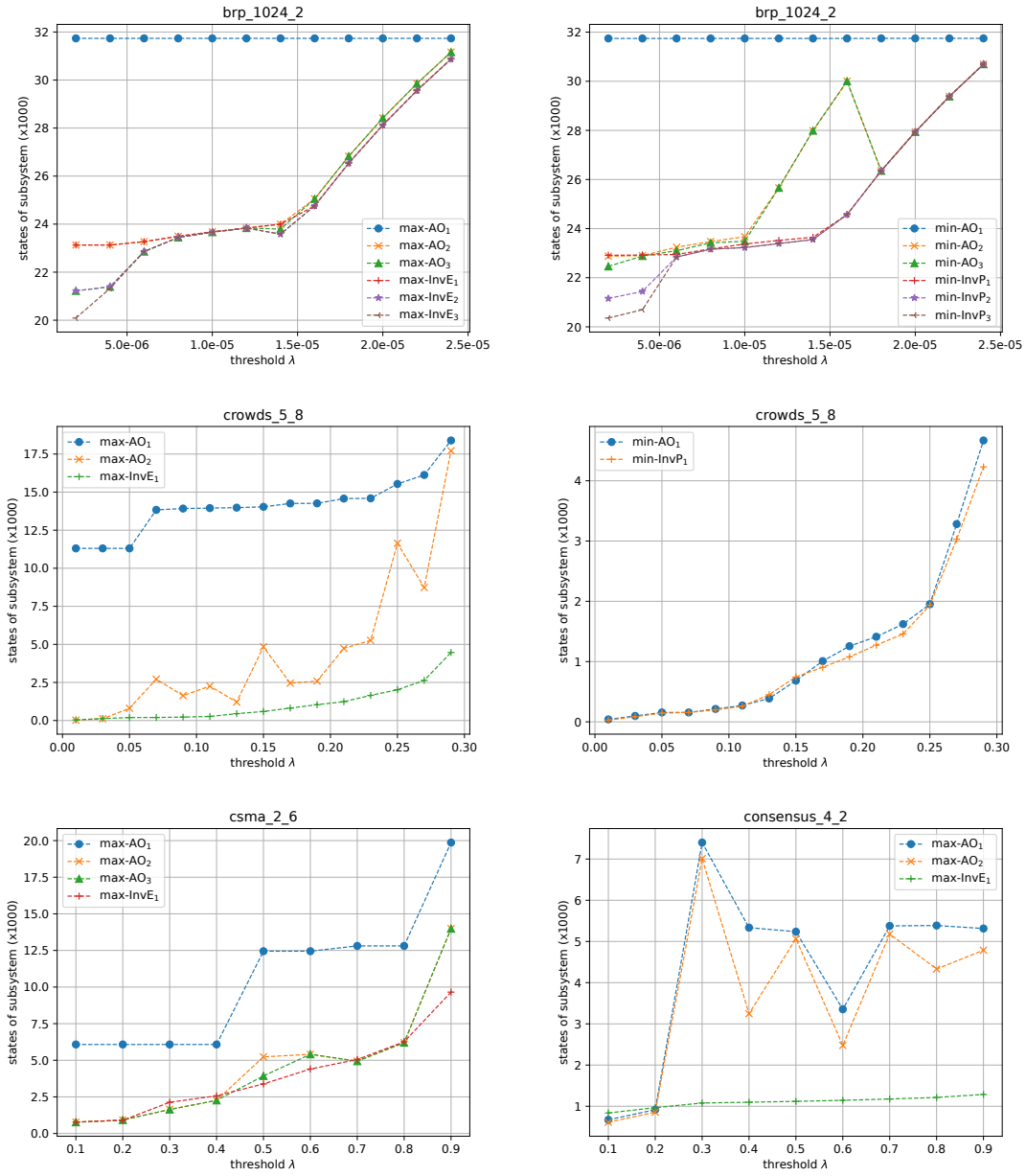


Figure 4.9: Sizes of witnessing subsystems produced by the quotient-sum heuristic using different initial objective functions in the first iteration. Results are plotted for the first iteration after which the heuristic stabilizes.

Table 4.4: Number of labels included in subsystems computed by label-based minimization approaches for the bounded retransmission protocol benchmark for four thresholds. The result of the quotient-sum heuristic using either min- or max-formulation after three iterations is shown along with the minimum achievable (lines called "exact"), as computed by the MILP approach. For the larger benchmark, using the max-formulation yields suboptimal results, and runs into timeouts for large thresholds. The min-formulation always produced optimal results.

	$\lambda =$	0.97	0.978	0.982	0.986	0.994
brp_32_8	min-QSHeur$_3$	2	3	3	3	3
	max-QSHeur$_3$	2	3	3	3	3
	exact	2	3	3	3	3
brp_1024_8	min-QSHeur$_3$	3	4	4	4	4
	max-QSHeur$_3$	3	6	4	4	4
	exact	3	4	4	4	4

vector **invp** $\in \mathbb{Q}^S$ defined as the point-wise inverse (in the same way as above) of the vector $\mathbf{pr}^{\min}$ containing the minimal reachability probabilities for each state.

Figure 4.9 compares the quotient-sum heuristic under different initial objectives. The standard initialization is denoted by AO (for "all-ones"), while the results of choosing **inve** (respectively **invp**) as initial objectives are denoted by InvE (respectively InvP). As before, subscripts (e.g. AO$_i$) indicate how many iterations of the heuristic were run, and a prefix "$\mathfrak{m}$-" (e.g. max-AO) indicates that $\mathcal{F}^{\mathfrak{m}}_{\mathcal{M},\geq}(\lambda)$ is used as underlying system of inequalities (with $\mathfrak{m} \in \{\min, \max\}$). In Figure 4.9 we only plot the result of the iteration of the quotient-sum heuristics after which the results stabilize.

The most interesting observation of Figure 4.9 is that the "spike" phenomenon discussed before can indeed be relieved by changing the initial objective. While the standard initialization AO induces such spikes when using $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ for brp_1024_2 and when using $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ in crowds_5_8, this is not the case when initializing with **invp** or **inve**, respectively. Furthermore, the heuristic often stabilizes faster with the alternative initializations. For example, using **inve** in the quotient-sum heuristic over $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ for crowds_5_8, csma_2_6 and consensus_4_2 produces a very small witnessing subsystem already after one iteration, and the heuristic stabilizes thereafter. Generally, it appears that **invp** works better for formulations using $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$, while **inve** works better for formulations using $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$, and hence these are the configurations which are shown in Figure 4.9.

The lower part of Figure 4.9 showcases two MDP benchmarks. It shows that for MDPs, using the alternative initializations may yield much better results. For consensus_4_2, the standard initialization yields large spikes, while, in comparison, initializing with **inve** yields very small subsystems for all thresholds. We have only included experiments on the maximal probability here, as the differences between initializations are not as large for minimal probabilities.

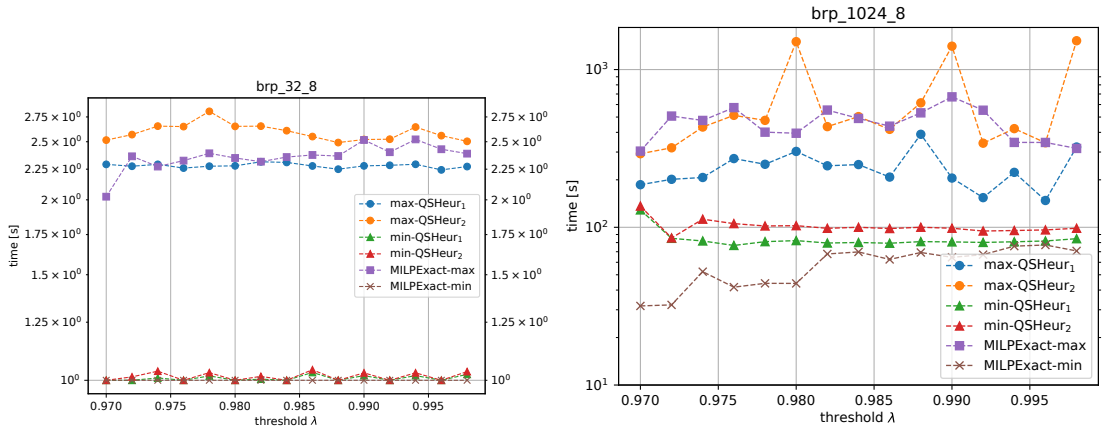


Figure 4.10: Computation times used for the label-based minimization approaches on the bounded retransmission protocol benchmark. All algorithms based on the inequalities $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ perform significantly better than those based on $\mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$.

Label-based minimization

The bounded retransmission protocol is designed to send N chunks of data with at most K attempted retransmissions per chunk. A natural question in this context is the following. For a given probability λ, what is the least value of K which guarantees successfully sending all chunks with probability at least λ? This question will serve as an inspiration for us to consider label-based minimization.

Let $\mathcal{M}$ be the Markov chain in reachability form one gets from the model brp_N_K along with the reachability objective of successfully sending all chunks. Observe that this is not the same target we considered in other experiments involving the bounded retransmission protocol. The Prism model of the system includes a variable `nrtr` whose value corresponds to the number of retries which have already been attempted for some chunk of data. We consider the labeling Λ induced by the value of this variable by setting $\Lambda(s) = i$ iff the value of `nrtr` is i in state s.

If there exists a witnessing subsystem $\mathcal{M}'$ for $\Pr_{\mathcal{M}}(\Diamond\,\text{target}) \geq \lambda$ which includes only the labels $\{0, \ldots, i\}$, for some $i < K$, then this means that "target" is reached with probability at least λ even if the number of attempted retries is bounded by i for each chunk. As the label i+1 is only reached through states with label i in this model, any useful subsystem includes all labels from an interval $\{0, 1, \ldots, i\}$. Our algorithms produce only such subsystems and henceforth we will only speak about the number of labels included in a subsystem. Table 4.4 shows the number of labels included in subsystems produced by the label-based version of the quotient-sum heuristic (see Algorithm 2) and exact minimization as computed by the MILP approach (see Definition 4.35). The heuristic based on the system of inequalities $\mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ always produces the minimal result, while the other instance produces the wrong result for the second threshold in the larger instance of the protocol, where it returns 6, while 4 labels suffice.

Figure 4.10 takes a closer look on the computation times for these instances. We can see that the algorithms using the min-formulations are always faster, and the exact approaches (which solve MILPs) do not require more time in general than the heuristic approaches. To explain this, recall that the number of binary variables in the MILP correspond to the number of labels which are considered. Hence, these programs contain only eight binary variables, which is much less

than the number of states. On the other hand, the labeled quotient-sum heuristic adds further variables to the linear program, and includes much fewer non-zero coefficients in the objective function. As these experiments show, such changes may make the LPs harder to solve.

It follows that while the MILPs do not scale well enough to handle state-based minimization even for systems with over a few hundred states, they may well be a valuable alternative when one is interested in label-minimal subsystems and the number of labels is not too large.

Summary of experimental results. To sum up the experimental results, Tables 4.5 and 4.7 present computation times and sizes of computed subsystems of all considered heuristics for models of various size. The main results of all conducted experiments are as follows.

- The MILP-based approaches to compute minimal witnesses scale to systems with at most a few hundred states. Their computation time varies significantly with the threshold, and their performance is comparable to existing approaches.
- The quotient-sum heuristic often computes subsystems which are close to the optimum, and is competitive with existing approaches. Sometimes, it produces results which are far off and adapting the initial objective function can help in such cases. Usually, it stabilizes within a few iterations. Its computation time is largely invariant to the threshold, and it scales to systems of about 10^5 states.
- For label-based minimization, the quotient-sum heuristic is able to compute small witnessing subsystems but, in the presence of few labels, may require relatively high computation times. In this case the MILP-based approaches may even be faster.

Table 4.5: Results of the different heuristic approaches for MDPs. The table entries include the time in seconds of computing the result, and the size of the computed subsystem in brackets (timeouts are marked by -). The results for the quotient-sum heuristic are always those obtained after three iterations. The thresholds λ_i are defined in Table 4.6.

		λ_1	λ_2	λ_3	λ_4
csma_2_6 $\lvert S\rvert = 66\,718$ $\lvert\mathcal{E}\rvert = 66\,788$	min-qs-ao	24.8 (441)	25.1 (1458)	25.5 (3718)	26.7 (10 012)
	min-qs-invp	51.8 (441)	53.9 (1458)	54.2 (3718)	54.2 (10 012)
	max-qs-ao	104.0 (768)	109.8 (1637)	89.0 (5408)	103.9 (13 987)
	max-qs-inve	68.8 (641)	68.8 (2092)	69.7 (4403)	80.0 (9653)
csma_3_2 $\lvert S\rvert = 36\,850$ $\lvert\mathcal{E}\rvert = 38\,456$	min-qs-ao	17.1 (8170)	17.7 (11 568)	16.9 (23 098)	17.0 (28 535)
	min-qs-invp	31.9 (8170)	32.4 (11 568)	34.2 (23 098)	34.0 (28 535)
	max-qs-ao	39.5 (1463)	53.4 (1596)	49.1 (1650)	47.1 (1747)
	max-qs-inve	40.3 (1607)	40.2 (1754)	43.9 (5061)	43.5 (7847)
consensus_2_4 $\lvert S\rvert = 528$ $\lvert\mathcal{E}\rvert = 784$	min-qs-ao	< 1 (201)	< 1 (292)	< 1 (308)	< 1 (420)
	min-qs-invp	< 1 (201)	< 1 (292)	< 1 (308)	< 1 (420)
	max-qs-ao	< 1 (104)	< 1 (106)	< 1 (119)	< 1 (128)
	max-qs-inve	< 1 (158)	< 1 (166)	< 1 (186)	< 1 (270)
consensus_4_4 $\lvert S\rvert = 43\,136$ $\lvert\mathcal{E}\rvert = 115\,840$	min-qs-ao	194.1 (9035)	171.6 (13 308)	48.1 (16 692)	202.0 (35 256)
	min-qs-invp	222.6 (9035)	171.4 (13 308)	80.9 (16 692)	227.6 (35 256)
	max-qs-ao	271.7 (2070)	533.4 (7122)	555.7 (4549)	577.0 (6631)
	max-qs-inve	209.5 (2615)	199.5 (2836)	196.9 (3136)	236.3 (4219)
firewire_3 $\lvert S\rvert = 4093$ $\lvert\mathcal{E}\rvert = 5519$	min-qs-ao	1.6 (240)	1.7 (479)	1.9 (1619)	2.2 (4093)
	min-qs-invp	3.9 (240)	3.9 (479)	4.1 (1619)	4.4 (4093)
	max-qs-ao	3.6 (85)	4.6 (251)	4.5 (578)	4.6 (578)
	max-qs-inve	4.6 (91)	5.0 (256)	5.0 (422)	5.0 (591)
firewire_30 $\lvert S\rvert = 138\,130$ $\lvert\mathcal{E}\rvert = 302\,654$	min-qs-ao	111.9 (618)	113.2 (1235)	433.1 (36 068)	347.9 (138 130)
	min-qs-invp	196.9 (618)	194.8 (1235)	489.6 (36 068)	397.9 (138 130)
	max-qs-ao	242.6 (85)	267.8 (249)	-	-
	max-qs-inve	275.5 (98)	260.9 (263)	254.3 (434)	254.8 (618)

Table 4.6: Threshold values used in Tables 4.5 and 4.7.

	λ_1	λ_2	λ_3	λ_4
brp	2.0×10^{-6}	1.0×10^{-5}	1.8×10^{-5}	2.0×10^{-5}
crowds	0.05	0.15	0.21	0.29
leader	0.1	0.3	0.7	0.9
csma	0.1	0.3	0.6	0.9
consensus	0.1	0.3	0.6	0.9
firewire	0.1	0.3	0.6	0.9

Table 4.7: Results of the different heuristic approaches for Markov chains. The table entries include the time in seconds of computing the result, and the size of the computed subsystem in brackets (timeouts are marked by -). The results for the quotient-sum heuristic are always those obtained after three iterations. The thresholds λ_i are defined in Table 4.6.

		λ_1	λ_2	λ_3	λ_4
brp_32_2 $\lvert S\rvert = 995$	min-qs-ao	< 1 (205)	< 1 (356)	< 1 (627)	< 1 (491)
	min-qs-invp	< 1 (196)	< 1 (218)	< 1 (447)	< 1 (491)
	max-qs-ao	< 1 (218)	< 1 (218)	< 1 (446)	< 1 (490)
	max-qs-inve	< 1 (218)	< 1 (218)	< 1 (446)	< 1 (490)
	comics-gl	< 1 (197)	< 1 (219)	< 1 (451)	< 1 (492)
	comics-lo	< 1 (197)	< 1 (824)	< 1 (995)	< 1 (995)
brp_1024_2 $\lvert S\rvert = 31\,747$	min-qs-ao	45.1 (22 461)	46.6 (23 486)	36.5 (26 352)	35.2 (27 938)
	min-qs-invp	68.5 (20 357)	56.7 (23 228)	57.3 (26 352)	57.2 (27 938)
	max-qs-ao	36.8 (21 221)	39.6 (23 666)	33.7 (26 827)	34.4 (28 418)
	max-qs-inve	48.4 (20 089)	52.0 (23 666)	45.1 (26 539)	45.5 (28 125)
	comics-gl	10.1 (17 976)	360.9 (22 648)	-	-
	comics-lo	-	-	-	-
crowds_2_8 $\lvert S\rvert = 832$	min-qs-ao	< 1 (31)	< 1 (73)	< 1 (101)	< 1 (148)
	min-qs-invp	< 1 (31)	< 1 (73)	< 1 (101)	< 1 (188)
	max-qs-ao	< 1 (73)	< 1 (75)	< 1 (171)	< 1 (210)
	max-qs-inve	< 1 (33)	< 1 (103)	< 1 (129)	< 1 (207)
	comics-gl	< 1 (41)	< 1 (120)	< 1 (177)	< 1 (286)
	comics-lo	< 1 (30)	< 1 (74)	< 1 (104)	< 1 (149)
crowds_5_8 $\lvert S\rvert = 27\,847$	min-qs-ao	10.4 (113)	10.7 (640)	10.5 (1397)	11.5 (4645)
	min-qs-invp	21.2 (113)	21.4 (743)	20.7 (1271)	21.3 (4230)
	max-qs-ao	37.8 (801)	43.5 (4838)	36.6 (4735)	40.0 (17 704)
	max-qs-inve	32.0 (182)	37.3 (592)	30.3 (1233)	35.8 (4469)
	comics-gl	< 1 (211)	4.3 (2489)	124.6 (5626)	-
	comics-lo	3.8 (110)	21.9 (519)	91.5 (1761)	345.7 (5569)
leader_6_6 $\lvert S\rvert = 234\,210$	min-qs-ao	-	454.6 (70 220)	411.2 (165 301)	129.4 (211 242)
	min-qs-invp	-	539.7 (70 220)	482.0 (165 301)	219.1 (211 242)
	max-qs-ao	-	-	-	-
	max-qs-inve	448.1 (37 811)	454.7 (70 006)	434.2 (163 509)	430.9 (210 524)
	comics-gl	11.3 (23 400)	251.0 (70 238)	-	-
	comics-lo	-	-	-	-

4.3 Witnessing subsystems for the expected total reward

Two notions of witnessing subsystems for threshold properties on the expected total reward in Markov chains were introduced in [QJD+15]. One of them is analogous to the one for probabilistic reachability constraints as defined in Section 4.1. Again, transitions can be redirected to "exit" to form subsystems, and for the expected total reward criterion this means that the possibility of collecting more reward in the future ends. Hence, the expected reward achieved in a subsystem can never increase with respect to the original Markov chain. If a subsystem satisfies a lower-bounded threshold constraint on the expected total reward when starting in s_{in}, it is called a witness for the property.

The second type of subsystem considered in [QJD+15] does not alter the transition structure of the Markov chain, but rather allows altering the reward function which determines how much reward a state contributes per visit. We will only consider the first notion here, and show that the correspondence between witnessing subsystems and Farkas certificates which was studied for probabilistic reachability constraints holds also for constraints on the expected total reward. Again, this yields novel MILP formulations for the problem of computing minimal witnesses. We also generalize the work in [QJD+15] by covering MDPs.

As in Section 3.2, where we considered Farkas certificates for constraints on the expected total reward, we will assume MDPs to be in *reward reachability form* (see Definition 2.12). This means that from each state the minimal probability to reach "exit" is one (or equivalently, that "exit" induces the only reachable proper end component). We use the same notion of subsystems considered in the previous sections (see Definition 4.1). It is essentially the definition of subsystems as given in [QJD+15, Definition 7] with the difference that we assume "exit" to be a single state (rather than a set of states) and allow individual transitions to be redirected, rather than only allowing all transitions of a state to be redirected together. In contrast to Section 3.2, we only consider nonnegative reward functions in this section.

Remark 4.41. We can now explain why we use the state "exit" rather than "target" when defining the expected total reward criterion. Namely, it allows us to use a unified definition of subsystem. For probabilistic reachability constraints it is important to distinguish the two states, and to redirect edges to "exit" in the definition of subsystems. For the expected total reward criterion the distinction is not important, but it is important that we redirect edges to the state in which rewards are collected. This makes sure that paths escaping this state forever have maximal probability zero in all subsystems. △

In the same way as for probabilistic reachability constraints the expected total reward cannot be higher in a subsystem than in the original system. The corresponding lemma can be proved by minor adaptations to the proof of Proposition 4.4.

Lemma 4.42. *Let $\mathcal{M}$ be an MDP in nonnegative reward reachability form and $\mathcal{M}'$ be a subsystem of $\mathcal{M}$. Then:*

$$\mathbb{E}^{\max}_{\mathcal{M}'}(\Diamond \mathrm{exit}) \leq \mathbb{E}^{\max}_{\mathcal{M}}(\Diamond \mathrm{exit}) \quad \textit{and} \quad \mathbb{E}^{\min}_{\mathcal{M}'}(\Diamond \mathrm{exit}) \leq \mathbb{E}^{\min}_{\mathcal{M}}(\Diamond \mathrm{exit}).$$

Proof. We first make the following observations:

- All finite paths in $\mathcal{M}'$ which do not end in "exit" are also finite paths in $\mathcal{M}$, and
- for all finite paths π in $\mathcal{M}$ which end in "exit", there exists a prefix π' of π such that π' exit is a finite path in $\mathcal{M}'$.

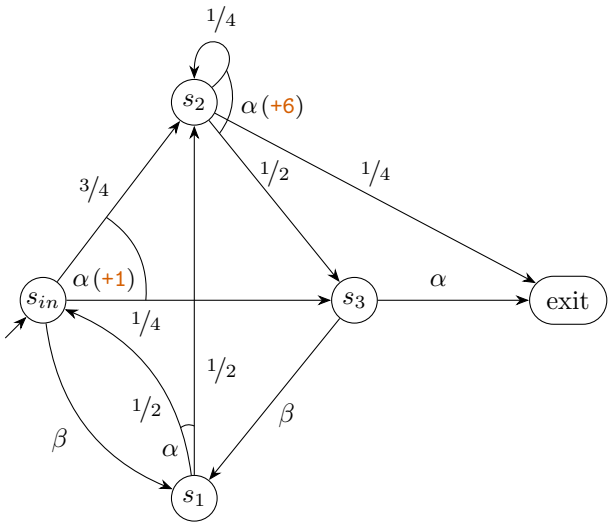


Figure 4.11: An MDP $\mathcal{M}$ in nonnegative reward reachability form. The reward function rew is defined by $\mathrm{rew}(s_{in}, \alpha) = 1, \mathrm{rew}(s_2, \alpha) = 6$ and zero for all other state-action pairs.

Using these facts one can, in the same way as in Proposition 4.4, extend each scheduler $\mathfrak{S}'$ of $\mathcal{M}'$ to a scheduler $\mathfrak{S}$ of $\mathcal{M}$ satisfying $\mathbb{E}^{\mathfrak{S}'}_{\mathcal{M}'}(\diamondsuit \mathrm{exit}) \leq \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(\diamondsuit \mathrm{exit})$. This relies on the fact that all states have a scheduler which reaches "exit" with probability one, and the reward function is nonnegative. Furthermore, each scheduler $\mathfrak{S}$ of $\mathcal{M}$ can be restricted to form a scheduler $\mathfrak{S}'$ of $\mathcal{M}'$ which again satisfies $\mathbb{E}^{\mathfrak{S}'}_{\mathcal{M}'}(\diamondsuit \mathrm{exit}) \leq \mathbb{E}^{\mathfrak{S}}_{\mathcal{M}}(\diamondsuit \mathrm{exit})$. □

Definition 4.43 (Witnesses for expected rewards). Let $\mathcal{M}$ be an MDP in nonnegative reward reachability form. Furthermore, let $\lambda \geq 0$ and $\mathcal{M}'$ be a subsystem of $\mathcal{M}$.

- $\mathcal{M}'$ is called a witness for $\mathbb{E}^{\max}_{\mathcal{M},s_{in}}(\diamondsuit \mathrm{exit}) \geq \lambda$ if $\mathbb{E}^{\max}_{\mathcal{M}',s_{in}}(\diamondsuit \mathrm{exit}) \geq \lambda$ holds.
- $\mathcal{M}'$ is called a witness for $\mathbb{E}^{\min}_{\mathcal{M},s_{in}}(\diamondsuit \mathrm{exit}) \geq \lambda$ if $\mathbb{E}^{\min}_{\mathcal{M}',s_{in}}(\diamondsuit \mathrm{exit}) \geq \lambda$ holds.

Example 4.44. Consider the MDP $\mathcal{M}$ in Figure 4.11, let $\mathbf{A}$ be its system matrix and $\mathbf{r} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be defined by $\mathbf{r}(s, \alpha) = \mathrm{rew}(s, \alpha)$. A Farkas certificate for $\mathbb{E}^{\max}_{\mathcal{M},s_{in}}(\diamondsuit \mathrm{exit}) \geq 7$ is given by:

$$\mathbf{y}_1 = \big((s_{in}, \alpha) \mapsto 1,\ (s_{in}, \beta) \mapsto 0,\ (s_1, \alpha) \mapsto 0,\ (s_2, \alpha) \mapsto 1,\ (s_3, \alpha) \mapsto 0,\ (s_3, \beta) \mapsto 0 \big).$$

This is because it satisfies $\mathbf{yA} \leq \delta_{s_{in}}$ and $\mathbf{yr} = \mathbf{y}(s_{in}, \alpha) + 6 \cdot \mathbf{y}(s_2, \alpha) = 7$. The corresponding witnessing subsystem is the one induced by state-supp$(\mathbf{y}_1) = \{s_{in}, s_2\}$. Indeed, one can check that even if states s_1 and s_3 were identified with "exit" (i.e., they would contribute zero reward), then the maximal expect reward from s_{in} would still be 7. A Farkas certificate for $\mathbb{E}^{\max}_{\mathcal{M},s_{in}}(\diamondsuit \mathrm{exit}) \geq 8$ is:

$$\mathbf{y}_2 = \big((s_{in}, \alpha) \mapsto 0,\ (s_{in}, \beta) \mapsto 2,\ (s_1, \alpha) \mapsto 2,\ (s_2, \alpha) \mapsto 4/3,\ (s_3, \alpha) \mapsto 0,\ (s_3, \beta) \mapsto 0 \big).$$

It corresponds to the subsystem induced by $\{s_{in}, s_1, s_2\}$ and to the memoryless deterministic scheduler which chooses β in s_{in}.

For the minimal expected reward, any subsystem which does not include s_1 has value zero due to the scheduler which chooses β in s_{in} (recall that all actions remain enabled in subsystems,

and "missing" transitions are redirected to "exit"). However, the subsystem $\{s_{in}, s_1, s_2\}$ is a witness for $\mathbb{E}^{\min}_{\mathcal{M},s_{in}}(\oplus\,\mathrm{exit}) \geq 7$, as certified by the following Farkas certificate:

$$\mathbf{z}_1 = \big(\ s_{in} \mapsto 7,\ s_1 \mapsto 7,\ s_2 \mapsto 8,\ s_3 \mapsto 0\ \big).$$

It satisfies $\mathbf{A}\mathbf{z}_1 \leq \mathbf{r}$ and $\mathbf{z}(s_{in}) \geq 7$. Observe that the subsystem $\{s_{in}, s_1, s_2\}$ is not a witness for $\mathbb{E}^{\min}_{\mathcal{M},s_{in}}(\oplus\,\mathrm{exit}) \geq 8$, as the scheduler which chooses α in s_{in} achieves only value 7. △

The following proposition relates Farkas certificates for the expected total reward criterion (see Propositions 3.32 and 3.33) to witnessing subsystems which are induced by their support. It is the analogon of Theorem 4.23 for probabilistic reachability constraints. Its proof is simplified by the fact that MDPs in reward reachability form are EC-free.

Proposition 4.45. *Let $\mathcal{M} = (S \cup \{\mathrm{exit}\}, \mathrm{Act}, s_{in}, P, \mathrm{rew})$ be an MDP in nonnegative reward reachability form, $\mathbf{A}$ be the system matrix of $\mathcal{M}$, $\lambda \geq 0$ and $\mathbf{r} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be defined by $\mathbf{r}(s,\alpha) = \mathrm{rew}(s,\alpha)$ for all $(s,\alpha) \in \mathcal{E}$. Fix a subset $S' \subseteq S$.*

1. *There exists a vector $\mathbf{z} \in \mathbb{R}^{S}_{\geq 0}$ satisfying $\mathbf{A}\mathbf{z} \leq \mathbf{r} \wedge \mathbf{z}(s_{in}) \geq \lambda$ such that $\mathrm{supp}(\mathbf{z}) \subseteq S'$ if and only if $\mathbb{E}^{\min}_{\mathcal{M}_{S'}}(\oplus\,\mathrm{exit}) \geq \lambda$ holds.*
2. *There exists a vector $\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ satisfying $\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\,\mathbf{r} \geq \lambda$ such that $\text{state-supp}(\mathbf{y}) \subseteq S'$ if and only if $\mathbb{E}^{\max}_{\mathcal{M}_{S'}}(\oplus\,\mathrm{exit}) \geq \lambda$ holds.*

Proof. Let $\mathcal{E}' = \{(s,\alpha) \in \mathcal{E} \mid s \in S'\}$. As observed before, the system matrix $\mathbf{A}'$ of the induced subsystem $\mathcal{M}_{S'}$ is the restriction of the system matrix of $\mathcal{M}$ to states in S', i.e., $\mathbf{A}' = \mathbf{A}|_{\mathcal{E}' \times S'}$. Additionally, define $\mathbf{r}' = \mathbf{r}|_{\mathcal{E}'}$. We only show (1.), as (2.) can be shown analogously.

(1.) "$\Longrightarrow$". By Lemma 4.22, the vector $\mathbf{z}' = \mathbf{z}|_{S'}$ satisfies $\mathbf{A}'\mathbf{z}' \leq \mathbf{r}' \wedge \mathbf{z}'(s_{in}) \geq \lambda$, from which $\mathbb{E}^{\min}_{\mathcal{M}_{S'}}(\oplus\,\mathrm{exit}) \geq \lambda$ follows by Proposition 3.32.

"$\Longleftarrow$". If $\mathbb{E}^{\min}_{\mathcal{M}_{S'}}(\oplus\,\mathrm{exit}) \geq \lambda$ holds, then the vector $\mathbf{z}' = \mathbf{ex}^{\min} \in \mathbb{R}^{|S'|}$ containing the expected total reward achieved starting from any state in $\mathcal{M}_{S'}$ is a solution of $\mathbf{A}'\mathbf{z}' \leq \mathbf{r}' \wedge \mathbf{z}'(s_{in}) \geq \lambda$ (see also Proposition 3.32). As $\mathcal{M}$ is in nonnegative reward reachability form, $\mathbf{ex}^{\min}$ is nonnegative. But then the vector $\mathbf{z} \in \mathbb{R}^{S}$ defined by $\mathbf{z}(s) = \mathbf{z}'(s)$ for all $s \in S'$, and $\mathbf{z}(s) = 0$ otherwise, satisfies $\mathbf{A}\mathbf{z} \leq \mathbf{r} \wedge \mathbf{z}(s_{in}) \geq \lambda$, by Lemma 4.22. □

As a corollary, it follows that the (state-)support minimal vectors satisfying these systems of linear inequalities correspond to minimal witnessing subsystems for the corresponding expected reward constraints.

Corollary 4.46. *Let $\mathcal{M} = (S \cup \{\mathrm{exit}\}, \mathrm{Act}, s_{in}, P, \mathrm{rew})$ be an MDP in nonnegative reward reachability form, $\mathbf{A}$ be the system matrix of $\mathcal{M}$, $\lambda \geq 0$ and $\mathbf{r} \in \mathbb{R}^{\mathcal{E}}_{\geq 0}$ be defined by $\mathbf{r}(s,\alpha) = \mathrm{rew}(s,\alpha)$ for all $(s,\alpha) \in \mathcal{E}$. Define*

$$\mathcal{P}_{\min} = \{\mathbf{z} \in \mathbb{R}^{S}_{\geq 0} \mid \mathbf{A}\mathbf{z} \leq \mathbf{r} \wedge \mathbf{z}(s_{in}) \geq \lambda\} \quad \text{and} \quad \mathcal{P}_{\max} = \{\mathbf{y} \in \mathbb{R}^{\mathcal{E}}_{\geq 0} \mid \mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \wedge \mathbf{y}\mathbf{r} \geq \lambda\}.$$

Then for all $S' \subseteq S$ we have

1. *There exists a vector $\mathbf{z} \in \mathcal{P}_{\min}$ with minimal support and $\mathrm{supp}(\mathbf{z}) = S'$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for $\mathbb{E}^{\min}_{\mathcal{M}}(\oplus\,\mathrm{exit}) \geq \lambda$.*
2. *There exists a vector $\mathbf{y} \in \mathcal{P}_{\max}$ with minimal state-support and $\text{state-supp}(\mathbf{y}) = S'$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for $\mathbb{E}^{\max}_{\mathcal{M}}(\oplus\,\mathrm{exit}) \geq \lambda$.*

With the above corollary in place, we can derive MILPs whose optimal solutions correspond to minimal witnessing subsystems for lower bounds on the expected total reward criterion in an analogous way as for probabilistic reachability constraints. The assumption that MDPs in reward reachability form are EC-free simplifies the computation of the upper bound required for the generic MILP defined in Lemma 4.27. In this setting the minimal expected reward vector serves as an upper bound for vectors satisfying $\mathbf{Az} \leq \mathbf{r}$, and the vector $\mathbf{u}_{ev}$, containing the maximal expected number of visits over all memoryless deterministic schedulers (see Equation (4.1)), serves as an upper bound for vectors satisfying $\mathbf{yA} \leq \delta_{s_{in}}$. As the MDP is EC-free here by assumption, the latter value can be computed in polynomial time (see Section 4.2.2).

Definition 4.47 (MILPs for expected total reward). Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P, \text{rew})$ be an MDP in nonnegative reward reachability form with system matrix $\mathbf{A}$ and enabled state-action pairs $\mathcal{E}$. Furthermore, let $\mathbf{r} \in \mathbb{Q}_{\geq 0}^{\mathcal{E}}$ be defined by $\mathbf{r}(s,\alpha) = \text{rew}(s,\alpha)$ for all $(s,\alpha) \in \mathcal{E}$ and let $\lambda \geq 0$.

- The min-witness MILP for expected rewards is given by: minimize $\sum_{s \in S} \sigma(s)$ such that

$$\mathbf{Az} \leq \mathbf{r}, \quad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{z}(s_{in}) \geq \lambda \quad \text{and} \quad \mathbf{z}(s) \leq \sigma(s) \cdot \mathbf{ex}^{\min}(s) \text{ for all } s \in S.$$

- The max-witness MILP for expected rewards is given by: minimize $\sum_{s \in S} \sigma(s)$ such that

$$\mathbf{yA} \leq \delta_{s_{in}}, \quad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{y} \cdot \mathbf{r} \geq \lambda \quad \text{and} \quad \mathbf{y}(s,\alpha) \leq \sigma(s) \cdot \mathbf{u}_{ev}(s,\alpha) \text{ for all } (s,\alpha) \in \mathcal{E}.$$

In both MILPs, σ is a vector of binary variables of dimension S.

Proposition 4.48. *Let $\mathcal{M}, \mathbf{A}, \mathcal{E}, \mathbf{r}$ and λ be as in the above definition, and $S \cup \{\text{exit}\}$ be the states of $\mathcal{M}$. Then, for all $S' \subseteq S$:*

- *There exists an optimal solution $(\sigma, \mathbf{z})$ of the min-witness MILP for expected rewards with $\text{supp}(\mathbf{z}) = S'$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for the property $\mathbb{E}^{\min}_{\mathcal{M}}(\diamondsuit \text{exit}) \geq \lambda$, and*
- *There exists an optimal solution $(\sigma, \mathbf{y})$ of the max-witness MILP for expected rewards with $\text{state-supp}(\mathbf{y}) = S'$ if and only if $\mathcal{M}_{S'}$ is a minimal witnessing subsystem for the property $\mathbb{E}^{\max}_{\mathcal{M}}(\diamondsuit \text{exit}) \geq \lambda$.*

Proof. Consider the first statement. By Lemma 2.13 every solution of $\mathbf{Az} \leq \mathbf{r}$ is bounded from above by the vector $\mathbf{ex}^{\min}$. It follows that the max-witness MILP is an instance of the generic MILP to compute vectors with minimal support (see Lemma 4.27). The claim follows by combining Corollary 4.46 and Lemma 4.27.

For the second statement, we first observe that as $\mathcal{M}$ is EC-free, the maximum expected number of visits of each state-action pair is attained by a memoryless deterministic scheduler by Proposition 2.14, and therefore bounded. In particular, this implies that solutions of $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yr} \geq \lambda$ are point-wise bounded from above by $\mathbf{u}_{ev}$. Hence the max-witness MILP is an instance of the generic MILP defined in Lemma 4.27, under the labeling function $\Lambda : \mathcal{E} \to 2^S$ which maps each state-action pair to the corresponding state (i.e., $\Lambda(s,\alpha) = \{s\}$ for all $(s,\alpha) \in \mathcal{E}$). It follows from Lemma 4.27 that optimal solutions of the MILP correspond to vectors $\mathbf{y}$ with minimal state support, and the claim follows by applying Corollary 4.46. □

4.4 Witnessing subsystems for invariants

So far, we have only considered witnessing subsystems for probabilistic reachability constraints expressing *lower bounds* on the minimal or maximal reachability probability. Such a witness shows that the probability of reaching the target is *at least as high* as the threshold (either for some, or for all schedulers).

In this section we deal with the question of witnessing lower bounds on the optimal probability of *never reaching a target set* (or, equivalently, of staying in a given set of states forever). Such properties are called *invariants*. They correspond to standard probabilistic reachability constraints expressing *upper bounds* on the optimal reachability probabilities. This can be seen through the equivalences

$$\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond T) < \lambda \iff \mathbf{Pr}^{\min}_{\mathcal{M}}(\Box \overline{T}) \geq 1{-}\lambda \quad \text{and} \quad \mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond T) < \lambda \iff \mathbf{Pr}^{\max}_{\mathcal{M}}(\Box \overline{T}) \geq 1{-}\lambda,$$

where $\mathcal{M} = (S, \mathrm{Act}, s_{in}, P)$ is an MDP, $T \subseteq S$ and $\overline{T} = S \setminus T$.

In the following, we will define witnessing subsystems for properties of the form $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Box \overline{T}) \geq \lambda$ and $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Box \overline{T}) \geq \lambda$, and discuss how they can be computed. We will assume that the set T of states which should be avoided consists of the single, absorbing state "exit". Apart from that, we consider arbitrary MDPs.

The standard definition of subsystem (see Definition 4.1), which implicitly redirects edges to "exit", does not depend on the existence of the state "target". Hence, the same notion of subsystem can be used here. We get a result which is analogous to Proposition 4.4, namely that the probability of satisfying the property cannot increase when passing to a subsystem. Intuitively, the idea is that redirecting a transition to "exit" represents a worst case assumption for the property $\Box \neg \,\mathrm{exit}$. Here "$\neg\,\mathrm{exit}$" represents the set of all states excluding "exit". The proposition can be proven in the same way as Proposition 4.4.

Proposition 4.49. *Let $\mathcal{M} = (S \cup \{\mathrm{exit}\}, \mathrm{Act}, s_{in}, P)$ be an MDP and $\mathcal{M}'$ be a subsystem of $\mathcal{M}$ with states $S' \cup \{\mathrm{exit}\}$. Then, for all $s \in S'$ we have:*

$$\mathbf{Pr}^{\min}_{\mathcal{M}',s}(\Box \neg \,\mathrm{exit}) \leq \mathbf{Pr}^{\min}_{\mathcal{M},s}(\Box \neg \,\mathrm{exit}) \quad \textit{and} \quad \mathbf{Pr}^{\max}_{\mathcal{M}',s}(\Box \neg \,\mathrm{exit}) \leq \mathbf{Pr}^{\max}_{\mathcal{M},s}(\Box \neg \,\mathrm{exit}).$$

As supported by the above proposition, we call a subsystem $\mathcal{M}'$ of $\mathcal{M}$ *a witness* for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Box \neg \,\mathrm{exit}) \geq \lambda$ (respectively for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Box \neg \,\mathrm{exit}) \geq \lambda$) if $\mathcal{M}'$ itself satisfies the property.

Remark 4.50. The reachability constraint $\mathbf{Pr}^{\min}(\Diamond\,\mathrm{target}) < \lambda$ corresponds, by the equivalences above, to the property $\mathbf{Pr}^{\max}(\Box \neg \,\mathrm{target}) \geq 1 - \lambda$. The reason that we consider witnessing subsystems for $\mathbf{Pr}^{\max}(\Box \neg \,\mathrm{exit}) \geq \lambda$ here (rather than $\mathbf{Pr}^{\max}(\Box \neg \,\mathrm{target}) \geq \lambda$) is that in this setting we can use the same notion of subsystem (which redirects edges to "exit"). If one is interested in witnesses for properties of the form $\mathbf{Pr}^{\mathfrak{m}}(\Diamond\,\mathrm{target}) < \lambda$, one has to first swap the roles of "target" and "exit". △

Now, we turn to the question of computing minimal (or small) witnessing subsystems in the above sense. Let us first consider the MDP $\mathcal{M}_r$ one gets by collapsing all reachable states which cannot reach "exit" into a state called "target". If $\mathcal{M}_r$ is EC-free, then the problem can be reduced to the computation of witnesses for standard (lower-bounded) probabilistic reachability constraints. This is because in this case $\Pr^{\mathfrak{S}}_{\mathcal{M}_r}(\Box \neg \,\mathrm{exit}) = \Pr^{\mathfrak{S}}_{\mathcal{M}_r}(\Diamond\,\mathrm{target})$ holds for all schedulers $\mathfrak{S}$ of $\mathcal{M}_r$. Hence, a subsystem is a witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}_r}(\Box \neg \,\mathrm{exit}) \geq \lambda$ if and only if it is a witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}_r}(\Diamond\,\mathrm{target}) \geq \lambda$, where $\mathfrak{m} \in \{\min, \max\}$. In particular, this implies that computing

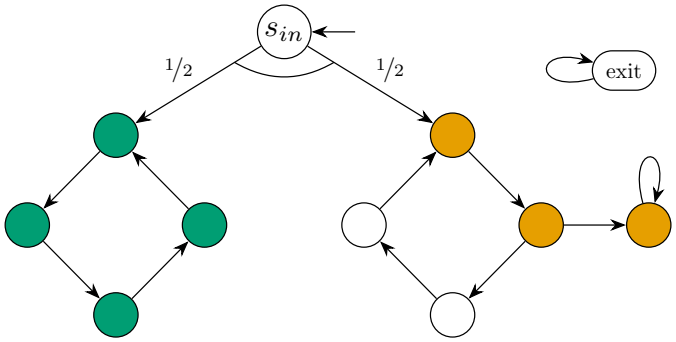


Figure 4.12: An MDP $\mathcal{M}$ in which the only probabilistic choice is in the initial state, which chooses with probability $1/2$ between continuing on the left or on the right. All other choices are nondeterministic. The state "exit" is unreachable in $\mathcal{M}$. The green states (in the left cycle) indicate a minimal witnessing subsystem for the property $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Box\neg \text{exit}) \geq 1/2$. On the other hand, the orange states (on the right) indicate a minimal witnessing subsystem for the property $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Box\neg \text{exit}) \geq 1/2$.

minimal witnesses for non-reachability properties remains NP-hard, as the hardness result in the standard case (Theorem 4.16) applies to Markov chains already.

So the interesting questions arise for MDPs with proper end components. Intuitively the status of end components changes when considering "$\Box\neg$ exit" in comparison to "$\Diamond$ target": Whereas realizing a proper end component (which does not include "exit") is desirable for the former property, it is not for the latter. Consider the MDP $\mathcal{M}$ in Figure 4.12 and the properties $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}}(\Box\neg \text{exit}) \geq 1/2$, with $\mathfrak{m} \in \{\min, \max\}$. For both properties it suffices to demonstrate that one of the probabilistic choices in the initial state leads to a state which avoids "exit" forever with (minimal or maximal) probability one. The smallest witnessing subsystem for $\mathfrak{m} = \min$ includes the entire left cycle. On the other hand, for $\mathfrak{m} = \max$ we do not have to take into account all schedulers, and hence the smallest witnessing subsystem in this case is induced by the shortest path to the rightmost absorbing state. The minimal probability of $\Box\neg$ exit is zero in the subsystem induced by the orange states, as there exists a scheduler which leaves the subsystem with probability one.

The example illustrates that to satisfy $\Box\neg$ exit a scheduler has to reach and realize proper end components (which exclude exit). Furthermore, for minimal probabilities, only subsystems which are "closed under end components" (this will be made precise below) are relevant. In the following, we will separately consider the cases of minimal and maximal probabilities.

The minimal probability of avoiding "exit" forever

The minimal probability of avoiding "exit" forever is above λ if λ is a lower bound on the probability of this property under all schedulers of $\mathcal{M}$. It follows that if any internal transition of an end component is redirected to "exit", then in the resulting subsystem all states in this end component have minimal probability zero of achieving the property. This is because there exists a scheduler which ensures that this transition is eventually taken (and therefore "exit" is reached) with probability one.

This observation implies that it is never useful for a subsystem to partially include an end component when considering minimal probabilities. Hence, we can consider the quotient

of maximal end components and thereby reduce the problem to the case of lower-bounded probabilistic reachability constraints.

Let $\mathcal{M} = (S \cup \{\text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in which "exit" is an absorbing state, and consider the property $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Box\neg\,\text{exit}) \geq \lambda$ for some $\lambda \in [0,1]$. Let $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be the maximal end components of $\mathcal{M}$ excluding the ones induced by "exit", and let

$$\mathcal{N} \;=\; \mathcal{M}^{\text{target}}_{/\mathcal{D}} \;=\; (\{E_1, \ldots, E_k, \text{exit}, \text{target}\},\ (S \times \text{Act}) \cup \{\tau\},\ [s_{in}]_{\mathcal{D}},\ P_{\mathcal{D}})$$

be the target-directed MEC-quotient of $\mathcal{M}$ as defined in Section 2.2.2. Here "target" is a fresh state introduced in the construction.

In $\mathcal{N}$, each maximal end component of $\mathcal{M}$ is represented by a state, and all proper end components have an additional τ-transition which moves to "target" with probability one. The τ-transition represents the fact that by realizing the proper end component, a scheduler indeed avoids "exit" forever. The MDP $\mathcal{N}$ is in reachability form, as $\mathbf{Pr}^{\max}_{\mathcal{N},s}(\Diamond\,\text{target}) > 0$ holds for all states s of $\mathcal{N}$ (excluding "exit") and furthermore, it is EC-free.

Our aim is to show that the subsystems $\mathcal{M}'$ of $\mathcal{M}$ which satisfy $\mathbf{Pr}^{\min}_{\mathcal{M}'}(\Box\neg\,\text{exit}) \geq \lambda$ essentially correspond to subsystems $\mathcal{N}'$ of $\mathcal{N}$ which satisfy $\mathbf{Pr}^{\min}_{\mathcal{N}'}(\Diamond\,\text{target}) \geq \lambda$. To this end, let us call a subsystem $\mathcal{M}' = (S' \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P')$ of $\mathcal{M}$ *useful*, if for all proper end components (E, A) of $\mathcal{M}$ we have either $S' \cap E = \emptyset$ or $E \subseteq S'$. Furthermore, if $E \subseteq S'$ holds, then all transitions of internal actions of (E, A) should be included fully in the subsystem. To explain this notion, assume that $\emptyset \subset (S' \cap E) \subset E$ holds for some proper end component (E, A). Then, for all states $s \in S' \cap E$ we have $\mathbf{Pr}^{\min}_{\mathcal{M}',s}(\Box\neg\,\text{exit}) = 0$. This is because some transition of the proper end component E is not included in the subsystem, which means that it now leads to "exit". The same is true if we have $E \subseteq S'$ but some (formerly) internal action of the end component has positive probability to reach "exit" in the subsystem. In both cases all remaining states in the end component can also be removed from the subsystem without decreasing the minimal probability of avoiding "exit" forever in any state. As a consequence, there is always a *useful* (label- or weight-) minimal witnessing subsystem of $\mathcal{M}$.

If $\mathcal{M}$ is equipped with a labeling function $\Lambda : S \to 2^L$ or a weight function $wgt : S \to \mathbb{N}$, we consider the new labeling and weight functions $\Lambda_{\mathcal{N}}$ and $wgt_{\mathcal{N}}$ for $\mathcal{N}$ defined by

$$\Lambda_{\mathcal{N}}(E) \;=\; \bigcup_{s \in E} \Lambda(s) \qquad \text{and} \qquad wgt_{\mathcal{N}}(E) \;=\; \sum_{s \in E} wgt(s)$$

for all $E \in \{E_1, \ldots, E_k\}$.

Proposition 4.51. *There exists a one-to-one correspondence h between subsystems of $\mathcal{N}$ and useful subsystems of $\mathcal{M}$ such that $\Lambda_{\mathcal{N}}(\mathcal{N}') = \Lambda(h(\mathcal{N}'))$, $wgt_{\mathcal{N}}(\mathcal{N}') = wgt(h(\mathcal{N}'))$ and $\mathbf{Pr}^{\min}_{\mathcal{N}'}(\Diamond\,\text{target}) = \mathbf{Pr}^{\min}_{h(\mathcal{N}')}(\Box\neg\,\text{exit})$ holds for all subsystems $\mathcal{N}'$ of $\mathcal{N}$.*

Proof. Given a subsystem $\mathcal{N}'$ of $\mathcal{N}$ we construct a useful subsystem $h(\mathcal{N}')$ of $\mathcal{M}$ by including all states and internal actions of (states representing) maximal end components included in $\mathcal{N}'$. Any transitions of external actions which are excluded in $\mathcal{N}'$ are also excluded in $h(\mathcal{N}')$. This construction clearly preserves the labeling and weight functions, and any useful subsystem of $\mathcal{M}$ corresponds uniquely to a subsystem of $\mathcal{N}$ via this correspondence. Every scheduler $\mathfrak{S}$ of $\mathcal{N}'$ naturally induces a scheduler $\mathfrak{S}'$ of $h(\mathcal{N}')$ which satisfies $\Pr^{\mathfrak{S}}_{\mathcal{N}'}(\Diamond\,\text{target}) = \Pr^{\mathfrak{S}'}_{h(\mathcal{N}')}(\Box\neg\,\text{exit})$. Here, if $\mathfrak{S}$ chooses a τ-action for a proper end component (E, A) in $\mathcal{N}'$, $\mathfrak{S}'$ simply chooses to never leave E. Likewise, a scheduler $\mathfrak{S}$ of $h(\mathcal{N}')$ induces a scheduler $\mathfrak{S}'$ of $\mathcal{N}'$ with

$\Pr^{\mathfrak{S}'}_{\mathcal{N}'}(\Diamond\,\text{target}) = \Pr^{\mathfrak{S}}_{h(\mathcal{N}')}(\Box\neg\,\text{exit})$. As before, if $\mathfrak{S}$ chooses to realize an end component, $\mathfrak{S}'$ takes the corresponding τ-action. This concludes the proof. □

As we only have to consider useful subsystems of $\mathcal{M}$ when looking for minimal witnessing subsystems for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Box\neg\,\text{exit}) \geq \lambda$, we can instead search for minimal witnessing subsystems of $\mathcal{N}$ for the property $\mathbf{Pr}^{\min}_{\mathcal{N}}(\Diamond\,\text{target}) \geq \lambda$. The latter problem has been treated extensively in foregoing sections.

The maximal probability of avoiding "exit" forever

In contrast to the previous section, we cannot collapse maximal end components when considering minimal witnesses for properties of the form $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Box\neg\,\text{exit}) \geq \lambda$. This is because a scheduler may not have to visit all states of a maximal end component C. Rather, it might realize an end component strictly included in C, or just pass through C to realize another end component (see Figure 4.12). In such a case collapsing maximal end components does not maintain the information of *how many states* (or labels) have to be included in a witnessing subsystem.

Let $\mathcal{M}$ be an MDP as in the previous section and $\mathcal{D} = \{(E_1, A_1), \ldots, (E_k, A_k)\}$ be the maximal end components of $\mathcal{M}$ *which are also proper* (i.e., such that $A_i(s) \neq \emptyset$ for all $s \in E_i$). Furthermore, let $S_E = \{s_E \mid s \in \bigcup_{1\leq i\leq k} E_i\}$ be a set of copies of the states of $\mathcal{M}$ which are included in some proper end component. We will construct an MDP which consists of $\mathcal{M}$ plus a copy of each proper maximal end component. In the copy, only internal actions will be accessible. To enter the copies, we add new actions α_E, which are copies of internal actions α. The copies of actions and states will be denoted by a suffix $_E$, that is, $s_E \in S_E$ denotes the copy of state $s \in S$, and so on.

A state-action pair (s, α) is said to be *internal*, if for some $(E_i, A_i) \in \mathcal{D}$ we have $s \in E_i$ and $\alpha \in A_i(s)$. Consider the MDP

$$\mathcal{N} = (S \cup S_E \cup \{\text{exit}\},\ \text{Act} \cup \{\alpha_E \mid \alpha \in \text{Act}\},\ s_{in},\ P'),\ \text{where}$$

- $P'(s, \alpha, s') = P(s, \alpha, s')$ for all $s, s' \in S \cup \{\text{exit}\}$ and $\alpha \in \text{Act}(s)$,
- $P'(s, \alpha_E, s'_E) = P(s, \alpha, s')$ for all $s, s' \in S$ and $\alpha \in \text{Act}(s)$, if (s, α) is internal,
- $P'(s_E, \alpha_E, s'_E) = P(s, \alpha, s')$ for all $s, s' \in S$ and $\alpha \in \text{Act}(s)$, if (s, α) is internal, and
- all other triples are assigned probability zero.

The idea is to model the two choices that a scheduler of $\mathcal{M}$ has inside a proper end component. Either it stays inside the end component forever, in which case action α_E can be chosen in $\mathcal{N}$. The remaining path will then remain inside the copy of the corresponding maximal end component in S_E thereafter. Or it can choose the original action and stay inside S.

We will consider the *core* of $\mathcal{N}$, denoted by $\mathcal{N}^C$, in which states S_E of $\mathcal{N}$ are identified with "target". It is defined as follows:

$$\mathcal{N}^C = (S \cup \{\text{target}, \text{exit}\}, \text{Act} \cup \{\alpha_E \mid \alpha \in \text{Act}\}, s_{in}, P_C),\ \text{where}$$

- $P_C(s, \alpha, s') = P'(s, \alpha, s')$ for all $s, s' \in S \cup \{\text{exit}\}$ and $\alpha \in \text{Act}$, and
- $P_C(s, \alpha_E, \text{target}) = 1$ for all $s \in S$ and $\alpha \in \text{Act}$, if (s, α) is internal.

Hence, each state included in a proper end component has an action leading to "target" immediately. The aim is to construct a system of linear inequalities whose solutions (with small state-support) correspond to (small) witnessing subsystems for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Box\neg \text{exit}) \geq \lambda$. To this end we will use:

1. Farkas certificates which show that "target" is reachable in $\mathcal{N}^C$ with probability at least λ.
2. An equation system $\mathbf{y}_i\mathbf{A}_i = \mathbf{0}$ for each proper maximal end component (E_i, A_i), whose solutions correspond to different ways of realizing that end component, and
3. Inequalities linking the above two, forcing certain entries in $\mathbf{y}_i$ to be non-zero.

The second point above should be compared with Lemma 3.8 and Remark 3.29, which discuss how solutions of similar systems of linear (in-)equalities relate to proper end components.

Now, let us define this system of linear inequalities precisely. Let $\mathbf{A}, \mathbf{t}$ be the system matrix and target vector of $\mathcal{N}^C$, and $\mathbf{A}_i$ be the system matrix of the maximal end component (E_i, A_i) (with $1 \leq i \leq k$), defined as follows for all $s, t \in E_i$ and $\alpha \in A_i(s)$:

$$\mathbf{A}_i((s,\alpha), t) = \begin{cases} 1 - P(s,\alpha,s) & \text{if } s = t, \\ -P(s,\alpha,t) & \text{if } s \neq t. \end{cases}$$

Let $\mathbf{y}$ be a vector of variables of dimension $|\mathcal{E}|$ and $\mathbf{y}_i$ be a vector containing one entry per enabled, internal state-action pair of the maximal end component (E_i, A_i). Consider the following system of linear inequalities:

$$\begin{aligned}
&\mathbf{y}\mathbf{A} \leq \delta_{s_{in}} \ \wedge \ \mathbf{y}\mathbf{t} \geq \lambda, \\
&\mathbf{y}_i\mathbf{A}_i = \mathbf{0} && \text{for all } 1 \leq i \leq k, \text{ and} \\
&\mathbf{y}(s,\alpha_E) \cdot P(s,\alpha,s') \leq \sum_{\beta \in \mathrm{Act}(s')} \mathbf{y}_i(s'_E, \beta_E) && \text{for all internal } (s,\alpha) \text{ and } s' \in S, \text{and} \\
&\mathbf{y} \geq \mathbf{0}, \ \mathbf{y}_i \geq \mathbf{0} && \text{for all } 1 \leq i \leq k.
\end{aligned}$$

The state-support of a vector $\mathbf{Y} = (\mathbf{y}, \mathbf{y}_1, \ldots, \mathbf{y}_k)$ is defined as

$$\text{state-supp}(\mathbf{Y}) = \{\ s \in S \ \mid \ s \in \mathrm{supp}(\mathbf{y}) \ \text{ or } \ s_E \in \mathrm{supp}(\mathbf{y}_i) \ \text{ for some } \ 1 \leq i \leq k\ \}.$$

Proposition 4.52. *For all $S' \subseteq S$, there exists a solution $\mathbf{Y}$ to the above system of linear inequalities with* state-supp$(\mathbf{Y}) \subseteq S'$ *if and only if* $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Box\neg \text{exit}) \geq \lambda$ *holds.*

Proof. "$\Longrightarrow$". Let $\mathbf{Y}$ be a solution to the above system of linear inequalities and let $S' =$ state-supp$(\mathbf{Y})$. We claim that the induced subsystem $\mathcal{M}_{S'}$ satisfies $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Box\neg \text{exit}) \geq \lambda$. Define $\mathbf{Y} = \{\mathbf{y}, \mathbf{y}_1, \ldots, \mathbf{y}_k\}$ and consider the two sets

$$S_1 = \{\ s \in S \ \mid \ s \in \text{state-supp}(\mathbf{y})\ \} \quad \text{and} \quad S_2 = \{\ s \in S \ \mid \ s_E \in \bigcup_{1 \leq i \leq k} \text{state-supp}(\mathbf{y}_i)\ \}.$$

We have $S_1 \cup S_2 = S'$, but the two sets are not necessarily disjoint. As $\mathbf{A}, \mathbf{t}$ are the system matrix and target vector of $\mathcal{N}^C$ (which is the *core* of $\mathcal{N}$, defined above), the vector $\mathbf{y}$ is a Farkas certificate for $\mathbf{Pr}^{\max}_{\mathcal{N}^C}(\Diamond \text{target}) \geq \lambda$ (see Definition 3.23). In $\mathcal{N}^C$, the state "target" represents the set of states S_E in $\mathcal{N}$. More precisely, as the only actions which move to S_E in $\mathcal{N}$ are of the form

(s, α_E), we have

$$\mathbf{yt} \;=\; \sum_{\substack{(s,\alpha_E)\,s.t.\\ (s,\alpha)\in\mathcal{E}}} \mathbf{y}(s,\alpha_E)\cdot P(s,\alpha,s') \;\geq\; \lambda.$$

Due to the constraint

$$\mathbf{y}(s,\alpha_E)\cdot P(s,\alpha,s') \;\leq\; \sum_{\beta\in\mathrm{Act}(s')} \mathbf{y}_i(s'_E,\beta_E)$$

we have $s' \in S_2$ whenever $\mathbf{y}(s,\alpha_E)\cdot P(s,\alpha,s') > 0$ holds for some $(s,\alpha) \in \mathcal{E}$. Hence, $\mathbf{y}$ is a certificate for the stronger claim that the maximal probability of reaching S_2 is above λ in $\mathcal{M}_{S'}$. The constraint $\mathbf{y}_i\mathbf{A}_i = \mathbf{0}$ implies that the states state-supp$(\mathbf{y}_i)$ form a disjoint set of end components, by Lemma 3.8. In particular, this means that after reaching S_2 there exists a scheduler in $\mathcal{M}_{S'}$ which stays inside S_2 forever. Altogether, this lets us construct a scheduler which avoids "exit" forever in $\mathcal{M}_{S'}$ with probability at least λ.

"$\Longleftarrow$". Suppose that $\mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Box\neg \text{exit}) \geq \lambda$ holds and let $\mathfrak{S}$ be a memoryless deterministic scheduler which achieves this probability. As invariance properties are complements of reachability properties, optimal probabilities to satisfy an invariant are always attained by some MD-scheduler. Let $S_2 \subseteq S'$ be the recurrent states (those included in some BSCC) in the induced Markov chain of $\mathcal{M}_{S'}$ under $\mathfrak{S}$, but excluding the state "exit". The assumption implies that S_2 is reached under $\mathfrak{S}$ in $\mathcal{M}$ with probability at least λ. Then, the set S_2, together with the actions chosen by $\mathfrak{S}$ in S_2, induces a disjoint set of end components in $\mathcal{M}$ (namely one for each BSCC of the induced Markov chain). It follows from statement (2.) of Lemma 3.8 that for each maximal end component (E_i, A_i) which intersects S_2 we find a vector $\mathbf{y}_i$ satisfying $\mathbf{y}_i\mathbf{A}_i = \mathbf{0}$, such that $\bigcup_{1\leq i\leq k}$ state-supp$(\mathbf{y}_i) = S_2$.

Let $S_1 = S' \setminus S_2$, which includes all states in S' which are not part of a BSCC in the Markov chain induced by $\mathfrak{S}$. Consider the scheduler $\mathfrak{S}'$ for $\mathcal{N}^C$ derived from $\mathfrak{S}$ as follows. For states $s \in S_1$, we let $\mathfrak{S}'(s) = \mathfrak{S}(s)$, and for states $s \in S_2$ we let $\mathfrak{S}'(s) = \alpha_E$, where $\alpha = \mathfrak{S}(s)$. If $s \in S_2$ holds then $\mathfrak{S}(s)$ is guaranteed to be internal, as s is contained in a BSCC under $\mathfrak{S}$.

As S_2 is reached with probability at least λ in $\mathcal{M}_{S'}$ under $\mathfrak{S}$, it follows that {target} is reached with probability at least λ under $\mathfrak{S}'$ in $\mathcal{N}^C_{S'}$ (the subsystem of $\mathcal{N}^C$ induced by S'). It follows from Theorem 4.23 that there exists a nonnegative solution of $\mathbf{yA} \leq \delta_{s_{in}} \wedge \mathbf{yt} \geq \lambda$ such that state-supp$(\mathbf{y}) \subseteq S'$ (recall that $\mathbf{A}, \mathbf{t}$ are the system matrix and target vector of $\mathcal{N}^C$). As S_2 is reached with probability at least λ in $\mathcal{M}_{S'}$ under $\mathfrak{S}$, such a solution can be found which additionally satisfies: if $\mathbf{y}(s,\alpha_E) > 0$, then for all $s' \in S$ with $P(s,\alpha,s') > 0$ we have $s' \in S_2$, and therefore $s' \in$ state-supp$(\mathbf{y}_i)$ for some $1 \leq i \leq k$. But then we can find $K \geq 0$ such that

$$\mathbf{y}(s,\alpha_E)\cdot P(s,\alpha,s') \;\leq\; K\cdot \sum_{\beta\in\mathrm{Act}(s')} \mathbf{y}_i(s'_E,\beta_E)$$

holds for all $(s,\alpha,s') \in S\times\mathrm{Act}\times S$. It follows that $\mathbf{Y} = (\mathbf{y}, K\cdot\mathbf{y}_1, \ldots, K\cdot\mathbf{y}_k)$ satisfies the system of linear inequalities defined above. We have state-supp$(\mathbf{Y}) \subseteq S'$, which concludes the proof. □

The above proposition paves the way for algorithms to compute minimal witnessing subsystems for invariant properties in the same way as has been described for the case of reachability. In particular, the generic mixed-integer linear program defined in Lemma 4.27, whose optimal solutions correspond to the minimal-support solutions of the underlying system of linear inequalities, can be used. Likewise, the quotient-sum heuristic can be applied.

Chapter 5

Probabilistic systems with low tree width

A standard way of dealing with computationally hard problems is to consider restricted, but important, classes of instances for which the problems become tractable. In graph theory, a particularly prominent restriction is to assume that graphs are "similar to trees". This has been captured formally by the notion of *tree width* [RS86, Bod97]. The tree width of a graph is a number which quantifies how similar to a tree it is. A large class of NP-complete problems become tractable for classes of graphs of bounded tree width [Cou90, Bod97], and natural graphs, such as the control flow graphs of many imperative languages, have bounded tree width [Tho98, GMT02].

The restriction of bounded tree width has been considered for probabilistic systems [CŁ13, CIP15, ACG+20]. These papers address classical problems in probabilistic model checking such as computing the maximal end components, the set of states with maximal reachability probability one, optimal mean-payoff values and reachability probabilities in Markov chains. For all these problems, algorithms with improved time and space requirements are given for models with bounded tree width.

This chapter considers the problem of computing minimal witnessing subsystems for probabilistic models whose underlying graph has low tree width. We first study the case that the underlying graph *is a tree* (transitions to "target" and "exit" are not included in the underlying graph structure here). In this case, we show that the problem of computing weight-minimal witnesses in Markov chains is solvable in polynomial time, given that the weights are encoded in unary. In particular, this generalizes the problem of computing state-minimal witnesses.

Encouraged by this result, we tackle the problem of computing state-minimal witnesses in probabilistic systems with *low tree width*. We introduce a novel notion of tree width, called *directed tree-partition width* for directed graphs. It is a strong notion in the sense that classes of directed graphs with bounded directed tree-partition width have bounded width with respect to all known notions of tree width for directed graphs.

The main result of this chapter states that the (corresponding decision-) problem of computing minimal witnesses is NP-complete for a class of Markov chains with directed tree-partition width six. This complements the hardness result for acyclic Markov chains proved in Theorem 4.16. It follows that we cannot hope for algorithms which run in polynomial time even

for Markov chains with constant directed tree-partition width, and hence also not for Markov chains with bounded width with respected to any other known measure of tree width.

To prove this result, we introduce an intermediate problem called the *d-dimensional matrix-pair chain problem*. It can be described geometrically as a one-player game in n rounds. Starting with an initial vector, in every round the player chooses one of two $d \times d$ matrices (which may be different in each round), which is then multiplied with the current vector. The goal is to ultimately (that is, after n rounds) end up with a point inside a predefined halfspace. We show that this problem is NP-complete for fixed d, and then reduce it to the witness problem for Markov chains with fixed directed path-partition width. The complexity results proved in this chapter are summed up in the following list. We will call a Markov chain *tree structured* if its underlying graph is a tree (this will be defined precisely later).

- A weight-minimal witnessing subsystem can be computed in polynomial time for tree structured Markov chains, given that the weights are encoded in unary (Proposition 5.4).
- If weights are encoded in binary, then the weighted witness problem is NP-complete for tree structured Markov chains. Furthermore, the labeled witness problems is also NP-complete for tree structured Markov chains (Proposition 5.6).
- The d-dimensional matrix-pair chain problem is NP-complete for $d = 2$ (Proposition 5.12) and also for $d = 3$ under the additional restriction that all matrices and vectors in the input contain only nonnegative values (Proposition 5.13).
- The witness problem is NP-complete for Markov chains with directed path-partition width six (Theorem 5.18).

It remains open whether the matrix-pair chain problem is NP-hard for $d = 2$ with the assumption of nonnegative matrices and vectors as input.

While the above results show that polynomial time algorithms cannot be hoped for even for Markov chains with constant tree width, it is still possible that one can design algorithms which make use of this special structure of such systems to yield better results in practice, despite being exponential in worst case. The final part of this chapter describes such an algorithm. It works bottom-up along the tree structure, computes partial subsystems and remembers only those which may be necessary to form a (global) witnessing subsystem. An experimental study shows that this algorithm outperforms the MILP-based approach for certain benchmarks for which well-structured tree decompositions can be computed easily.

Related work

While for undirected graphs there is one universally accepted and standard notion of tree width, this is not the case for directed graphs [GHK+16]. Several such notions have been proposed [Ree99, JRST01, Saf05]. The notion we introduce is most related to the *tree-partition width* which has been studied for undirected graphs [See85, Ede86, DO96, Woo09]. However, the result that deciding whether a tree-partition of a given size exists is NP-hard is not easily transferable from the undirected to the directed case (the corresponding theorem for undirected graphs is [Ede86, Theorem 2.2]).

Algorithms in the context of probabilistic model checking for systems with low tree width are also considered in [CŁ13, CIP15, ACG+20]. The notion of tree width which is used in these papers is the standard notion for undirected graphs. All problems they address are

polynomial-time solvable in the first place, and the contribution lies in providing algorithms with significantly better running times using the structure provided by systems with small tree width.

Outline

We start by showing that the weighted witness problem with unary weights is solvable in polynomial time for tree structured Markov chain (Section 5.1.1). If weights are encoded in binary, or if the goal is to minimize labels, the problem becomes NP-complete (Section 5.1.2). Then, we introduce the directed tree- and path-partition width (Section 5.2). We go on to study the d-dimensional matrix-pair chain problem (Section 5.3.1) and use it to show NP-completeness of the witness problem for Markov chains with bounded directed path-partition width (Section 5.3.2). Finally, we describe an algorithm which makes use of a given directed tree decomposition to compute a minimal witness (Section 5.4).

Relation to published work

A polynomial-time algorithm to compute minimal witnesses in tree structured Markov chains was described in [FJB20], in joint work with Florian Funke and Christel Baier. The algorithm given in this chapter is essentially the same, but has been generalized to compute weight-minimal witnesses in pseudo-polynomial time. Most of the remaining results in this chapter, apart from the NP-hardness results for weight-minimal (with binary encoding) and label-minimal witnesses in tree structured Markov chains, have been published in [JPB21]. This paper is joint work with Jakob Piribauer and Christel Baier, and has been presented at GandALF 2021.

5.1 The witness problem for Markov chains with tree structure

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be an MDP in reachability form. For the purpose of this chapter, the underlying graph of $\mathcal{M}$ will be defined as

$$\mathcal{U}_{\mathcal{M}} = \big(S, \quad \{(s,t) \in S \times S \mid \text{there exists } \alpha \in \text{Act}(s) \text{ such that } P(s,\alpha,t) > 0\}\big).$$

Here we do not take into account states target and exit, in contrast to the general definition of underlying graph. We say that $\mathcal{M}$ has *tree structure* if $\mathcal{U}_{\mathcal{M}}$ is a directed tree, i.e., all vertices have indegree at most one. The reason for excluding target and exit from the underlying graph is that we do not want transitions to these states to influence whether a system is tree structured.

First, we give a polynomial time algorithm for the unary weighted witness problem (in which weights are encoded in unary) for tree structured Markov chains. As a consequence, the standard witness problem (which asks whether any smallest witnessing subsystem has at most k states) is solvable in polynomial time for such Markov chains.

5.1.1 An algorithm for tree structured Markov chains and unary weights

We will first describe an algorithm for Markov chains with binary tree structure, and then show that the problem for arbitrary tree structure can be reduced to this special case. By binary tree structure we mean that additionally to being tree structured, each vertex in the underlying graph has at most two successors. Recall that the weighted witness problem takes as input

a Markov chain $\mathcal{M}$ with weight function wgt, a natural number K and a rational λ and asks whether a subsystem $\mathcal{M}'$ of $\mathcal{M}$ exists such that $\Pr_{\mathcal{M}',s_{in}}(\Diamond \text{target}) \geq \lambda$ and $wgt(\mathcal{M}') \leq K$.

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ be a Markov chain with binary tree structure, and let $wgt : S \to \mathbb{N}$ be a weight function for $\mathcal{M}$. For the algorithmic problems in this section, we assume that wgt is encoded using unary encoding, and we define $W = wgt(\mathcal{M}) = \sum_{s \in S} wgt(s)$. We show how, for any rational $\lambda \in [0,1]$, one can compute a weight-minimal witnessing subsystem of $\mathcal{M}$ in time polynomial in $|\mathcal{M}|$ and W (given that such a subsystem exists).

Let $\mathcal{M}_q$ be the Markov chain one gets by taking $\mathcal{M}$, making state q the new initial state and removing all unreachable states. We define a function $\text{ac} : S \times \{0, \ldots, W\} \to [0,1]$ which will return for state q and number w the *maximal reachability probability achievable* by a subsystem of $\mathcal{M}_q$ with weight at most w. Computing the function "ac" is enough to solve the weighted witness problem, as the size of a minimal witnessing subsystem of $\mathcal{M}$ for threshold λ is given by the minimal $w \in \{0, \ldots, W\}$ satisfying $\text{ac}(s_{in}, w) \geq \lambda$.

We now give a recursive definition of the function ac, using the fact that $\mathcal{M}$ has binary tree structure. In Lemma 5.2 we show that this function indeed matches the interpretation given above. First, we define $\text{ac}(q, w) = 0$ for all $q \in S$ and $w < wgt(q)$. Now we distinguish whether q is a leaf, has a single successor q' which is reached with probability μ, or two successors q_1, q_2 reached with probability μ_1 and μ_2. For all $i \in \{0, \ldots, W - wgt(q)\}$ define:

$$\begin{aligned}
&\text{(leaf)}: && \text{ac}(q, wgt(q)+i) &&= P(q, \text{target}) \\
&\text{(single-suc)}: && \text{ac}(q, wgt(q)+i) &&= P(q, \text{target}) + \mu \cdot \text{ac}(q', i) \\
&\text{(double-suc)}: && && \\
& && \text{ac}\big(q, wgt(q)+i\big) &&= P(q, \text{target}) + \max\{\, \mu_1 \cdot \text{ac}(q_1, j) + \mu_2 \cdot \text{ac}(q_2, i-j) \;\mid\; 0 \leq j \leq i \,\}
\end{aligned} \tag{5.1}$$

Observe that for all $q \in S$ the function $\text{ac}(q, \cdot)$ is monotonically increasing. More precisely, if $w_1 \leq w_2$, then we have $\text{ac}(q, w_1) \leq \text{ac}(q, w_2)$. This holds by definition for leafs q, and is preserved by the recursive definition in Equation (5.1).

Lemma 5.1. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ be a Markov chain with binary tree structure, $wgt : S \to \mathbb{N}$ be a weight function for $\mathcal{M}$ and $W = \sum_{s \in S} wgt(s)$. The function $\text{ac} : S \times W \to [0,1]$ as defined in Equation (5.2) can be computed in time polynomial in $|\mathcal{M}|$ and W.*

Proof. The function ac can be computed bottom up along the tree order as described in Equation (5.1). For each state q with two successors, and value $w \in \{0, \ldots, W\}$, one has to compute the maximum from at most W sums. Hence, to compute all values of ac for state q one needs to compute at most W^2 such sums. Computing the sum can be done in polynomial time in $|\mathcal{M}|$. This needs to be done for all states in S and thus at most $|S| \cdot W^2$ such sums have to be computed and compared. □

Lemma 5.2. *Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ be a Markov chain with binary tree structure, $wgt : S \to \mathbb{N}$ be a weight function for $\mathcal{M}$, $W = \sum_{s \in S} wgt(s)$ and $\text{ac} : S \times W \to [0,1]$ be as defined above.*

Then, for all $q \in S$, $w \in \{0, \ldots, W\}$ and $\lambda \in [0,1]$ we have $\text{ac}(q, w) \geq \lambda$ if and only if there exists a subsystem $\mathcal{M}'$ of $\mathcal{M}_q$ with weight at most w satisfying $\Pr_{\mathcal{M}',q}(\Diamond \text{target}) \geq \lambda$.

Proof. We proceed by induction on the tree order of S. It is enough to show the statement for the case with two successors q_1, q_2 reached with probability μ_1 and μ_2. This is because both

leaf and single-successor cases are special instances thereof, with $\mu_1 = \mu_2 = 0$ (leaf) and $\mu_2 = 0$ (single-suc).

"$\Longrightarrow$": Assume that $\mathrm{ac}(q, w) \geq \lambda$ for some $w \in \{0, \ldots, W\}$ and $\lambda \in [0, 1]$. We may assume that $w = wgt(q) + i$ for some $i \leq W - wgt(q)$, as otherwise $\mathrm{ac}(q, w) = 0$ holds. By Equation (5.1), we have

$$\mathrm{ac}\big(q, wgt(q) + i\big) \;=\; P(q, \mathrm{target}) + \max \{\, \mu_1 \cdot \mathrm{ac}(q_1, j) + \mu_2 \cdot \mathrm{ac}(q_2, i{-}j) \;\mid\; 0 \leq j \leq i \,\}.$$

Let j^* be such that the maximum is attained in the above expression. By induction hypothesis, there exists a subsystem of $\mathcal{M}_{q_1}$ with weight at most j^* and probability at least $\mathrm{ac}(q_1, j^*)$, and a subsystem of $\mathcal{M}_{q_2}$ with weight at most $i{-}j^*$ and probability at least $\mathrm{ac}(q_2, j^*)$. Attaching these subsystems to state q yields a subsystem of $\mathcal{M}_q$ with weight at most $wgt(q) + i$, and probability at least

$$P(q, \mathrm{target}) + \mu_1 \cdot \mathrm{ac}(q_1, j^*) + \mu_2 \cdot \mathrm{ac}(q_2, i{-}j^*) \quad = \quad \mathrm{ac}(q, wgt(q) + i).$$

"$\Longleftarrow$": Let $\mathcal{M}'$ be a subsystem of $\mathcal{M}_q$ with weight $wgt(q){+}i' \leq wgt(q){+}i = w$ and reachability probability $\lambda' \geq \lambda$. The subsystem $\mathcal{M}'$ can be decomposed into the state q plus some subsystems $\mathcal{M}_1$ of $\mathcal{M}_{q_1}$ and $\mathcal{M}_2$ of $\mathcal{M}_{q_2 y}$, where both could potentially be empty. Let us assume that $\mathcal{M}_1$ contributes weight j and has probability λ_1 of reaching target. This implies that $\mathcal{M}_2$ contributes weight $i'{-}j$. Let us assume that $\mathcal{M}_2$ achieves probability λ_2. By induction hypothesis, we have $\mathrm{ac}(q_1, j) \geq \lambda_1$ and $\mathrm{ac}(q_2, i'{-}j) \geq \lambda_2$. Furthermore, $\mathrm{ac}(q_2, i{-}j) \geq \mathrm{ac}(q_2, i'{-}j)$ holds as $\mathrm{ac}(q_2, \cdot)$ is monotonically increasing. From Equation (5.1) follows:

$$\begin{aligned} \mathrm{ac}(q, wgt(q) + i) \;&\geq\; P(q, \mathrm{target}) + \mu_1 \cdot \mathrm{ac}(q_1, j) + \mu_2 \cdot \mathrm{ac}(q_2, i{-}j) \\ &\geq\; P(q, \mathrm{target}) + \mu_1 \cdot \lambda_1 + \mu_2 \cdot \lambda_2 \;=\; \lambda' \;\geq\; \lambda. \end{aligned}$$

□

Binarization of Markov chains

The algorithm presented above assumes that its input is a Markov chain with binary tree structure. We now show that Markov chains with arbitrary tree structure can always be transformed into this special form, while preserving minimal witnessing subsystems.

Let $\mathcal{M} = (S \cup \{\mathrm{target}, \mathrm{exit}\}, s_{in}, P)$ be a Markov chain with tree structure. The idea is to apply a local transformation to each state $q \in S$ as follows. Let $q_0, \ldots, q_k$ be the $k+1$ successors of q in S, reachable with probabilities $\mu_0, \ldots, \mu_k$. We add $k{-}1$ fresh states $u_1, \ldots, u_{k-1}$, and define their transitions as follows. Here we identify $u_0 := q$.

$$\begin{aligned} P'(u_j, q_j) \;&=\; \frac{\mu_j}{1 - \sum_{0 \leq i < j} \mu_i}, && \text{for } \; 0 \leq j < k \\ P'(u_j, u_{j+1}) \;&=\; 1 - P'(u_j, q_j), && \text{for } \; 0 \leq j < k{-}1 \\ P'(u_{k-1}, q_k) \;&=\; \frac{\mu_k}{1 - \sum_{0 \leq i < k-1} \mu_i} \end{aligned}$$

The idea is shown in Figure 5.1. We call the result of applying this transformation to all states of $\mathcal{M}$ the *binarization* of $\mathcal{M}$, denoted by $\mathcal{B}_\mathcal{M}$. For $0 \leq j < k$ we have

$$\Pr_{\mathcal{B}_\mathcal{M}, q}(\Diamond q_j) \;=\; \Pr_{\mathcal{B}_\mathcal{M}}(q\, u_1 \ldots u_j\, q_j) \;=\; \prod_{0 \leq l < j} \left(1 - \left(\frac{\mu_l}{1 - \sum_{0 \leq i < l} \mu_i}\right)\right) \cdot \frac{\mu_j}{1 - \sum_{0 \leq i < j} \mu_i} \;=\; \mu_j.$$

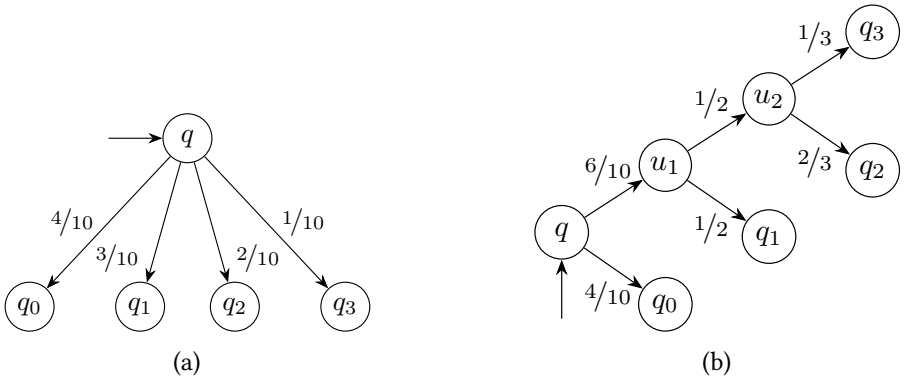


Figure 5.1: A Markov chain (a) and its binarization (b).

This follows from $\prod_{0\leq l<j}\left(1-\left(\frac{\mu_l}{1-\sum_{0\leq i<l}\mu_i}\right)\right) = 1-\sum_{0\leq i<j}\mu_i$, which can be shown by induction on j. For $j=k$, the corresponding path is $q\ u_1\dots u_{k-1}\ q_k$ and the formula is almost the same. The number of states that have to be added is bounded by the number of transitions in $\mathcal{M}$. If $\mathcal{M}$ is equipped with a weight function wgt, or a labeling function Λ, we interpret the same functions as weight (respectively, labeling) functions for $\mathcal{B}$ by assigning to all states of $\mathcal{B}$ that are not in S weight zero (respectively, the empty set of labels).

Take arbitrary set $S'\subseteq S$ and let U' be the states in $\mathcal{B}_{\mathcal{M}}$ which lie on some path between any two states in S'. Consider the induced subsystems $\mathcal{M}'=\mathcal{M}_{S'}$ and $\mathcal{B}'=\mathcal{B}_{S'\cup U'}$. By the above calculation, we have $\Pr_{\mathcal{M}',q}(\Diamond\,\text{target}) = \Pr_{\mathcal{B}',q}(\Diamond\,\text{target})$ for all $q\in S'$. Furthermore, by definition, we have $wgt(\mathcal{M}')=wgt(\mathcal{B}')$ and $\Lambda(\mathcal{M}')=\Lambda(\mathcal{B}')$. As no fresh state u in $\mathcal{B}$ has a direct edge to target, there is always a subsystem of the form $\mathcal{B}_{S'\cup U'}$ among the minimal witnessing subsystems, which implies the following lemma.

Lemma 5.3. *Let $\mathcal{M}=(S\cup\{\text{target},\text{exit}\},s_{in},P)$ be a Markov chain with tree structure, weight function wgt, labeling function Λ and let $\mathcal{B}$ be its binarization.*

Then for all $\lambda\in[0,1]$ and $K\in\mathbb{N}$, $\mathcal{M}$ has a subsystem $\mathcal{M}'$ satisfying $\Pr_{\mathcal{M}'}(\Diamond\,\text{target})\geq\lambda$ and $wgt(\mathcal{M}')\leq K$ (resp. $|\Lambda(\mathcal{M}')|\leq K$) if and only if $\mathcal{B}$ has a subsystem $\mathcal{B}'$ satisfying $\Pr_{\mathcal{B}'}(\Diamond\,\text{target})\geq\lambda$ and $wgt(\mathcal{B}')\leq K$ (resp. $|\Lambda(\mathcal{B}')|\leq K$).

Proposition 5.4. *Given a tree structured Markov chain $\mathcal{M}$ with weight function wgt, $\lambda\in[0,1]$ and $K\in\mathbb{N}$, a weight minimal witnessing subsystem for $\Pr_{\mathcal{M}}(\Diamond\,\text{target})\geq\lambda$ can be computed in polynomial time in $|\mathcal{M}|$ and the sum-of-weights W.*

Proof. Any tree structured Markov chain can be transformed into an equivalent one (with respect to the weighted witness problem) with binary tree structure in polynomial time by Lemma 5.3. For binary tree structured Markov chains the function $\text{ac}: S\times\{0,\dots,W\}\to[0,1]$ can be computed in polynomial time in $|\mathcal{M}|$ and the sum-of-weights W (Lemma 5.1) and we know by Lemma 5.2 that $\text{ac}(s_{in},w)\geq\lambda$ holds if and only if the Markov chain has a witnessing subsystem for λ with weight at most w. □

The standard witness problem, which asks for witnessing subsystem with a minimal amount of states, is a special instance of the weighted witness problem in which each state is given weight one. This gives us the following corollary.

Corollary 5.5. *The witness problem for tree structured Markov chains can be solved in polynomial time.*

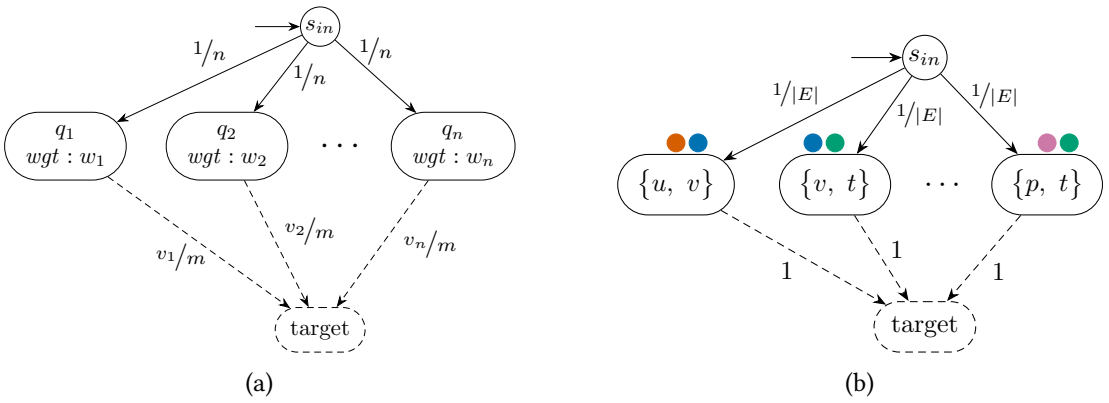


Figure 5.2: NP-hardness of the labeled and weighted witness problem for tree structured Markov chains, where weights are encoded in binary. Transitions to "exit" are omitted. As states "target" and "exit" and their incoming transition are not included in the underlying graph $\mathcal{U}_{\mathcal{M}}$ (and hence dashed), both Markov chains are tree structured. In (a), the structure of the reduction from knapsack to the weighted witness problem is sketched, whereas (b) sketches the reduction from clique to the labeled witness problem. Here states represent edges of some undirected graph, and each state is labeled by colors indicating which vertices participate in the corresponding edge.

5.1.2 NP-hardness with labels or binary weights

We now show that the labeled witness problem is NP-complete for tree structured Markov chains, and the same holds for the weighted witness problem if weights are encoded in binary. These problems are already in NP for arbitrary MDPs, so it remains to show NP-hardness. Let us first recall the definition of the knapsack problem, which is a classical NP-complete problem [Kar72].

The *knapsack problem* takes as input a tuple (n, w, v, W, V), where $n \in \mathbb{N}$ is the number of items, $w : \{1, \dots, n\} \to \mathbb{Q}$ defines the *weight* of each item, $v : \{1, \dots, n\} \to \mathbb{Q}$ defines the *value* of each item, $W \in \mathbb{Q}$ is the maximum allowed weight and $V \in \mathbb{Q}$ is the minimum required value. All numbers of the input are encoded in binary. The problem is to decide whether there exists a subset $N \subseteq \{1, \dots, n\}$ such that

$$\sum_{i \in N} w(i) \leq W \qquad \text{and} \qquad \sum_{i \in N} v(i) \geq V.$$

We call a subset $N \subseteq \{1, \dots, n\}$ satisfying the above property a *solution* of the knapsack instance. The following proposition shows how to reduce the knapsack problem in polynomial time to the weighted witness problem for tree structured Markov chains. It is essential here that the weights of the Markov chain are encoded in binary. Furthermore, the clique problem is polynomially reduced to the labeled witness problem for tree structured Markov chains. Sketches for both reductions are presented in Figure 5.2. As tree structured Markov chains can be transformed into Markov chains with binary tree structure by Lemma 5.3 while preserving weights, labels and probabilities, these problems remain hard for binary tree structured Markov chains.

Proposition 5.6. *The weighted witness problem (with weights encoded in binary) and the labeled witness problem are both NP-hard for tree structured Markov chains.*

Proof. 1.) We first consider the weighted witness problem with weights encoded in binary and give a polynomial reduction from the knapsack problem. Let (n, w, v, W, V) be an instance of the knapsack problem and define $m = \sum_{1 \leq i \leq n} v(i)$. Consider the Markov chain $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$ where $S = \{s_{in}\} \cup \{q_i \mid 1 \leq i \leq n\}$ and for all $1 \leq i \leq n$:

$$P(s_{in}, q_i) = \frac{1}{n}, \quad P(q_i, \text{target}) = \frac{v(i)}{m}, \quad \text{and} \quad P(q_i, \text{exit}) = 1 - \frac{v(i)}{m}.$$

Clearly, this Markov chain is tree structured (recall that transitions to "target" and "exit" are not included in underlying graph in this chapter). Additionally, consider the weight function $wgt : S \to \mathbb{N}$ defined by $wgt(s_{in}) = 0$ and $wgt(q_i) = w(i)$ for all $1 \leq i \leq n$.

We claim that there exists a solution $N \subseteq \{1, \ldots, n\}$ of the knapsack instance if and only if there exists a subsystem $\mathcal{M}'$ of $\mathcal{M}$ such that

$$wgt(\mathcal{M}') \leq W \qquad \text{and} \qquad \Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) \geq \frac{V}{m\, n}.$$

Given a solution N of the knapsack instance, consider the subsystem $\mathcal{M}'$ of $\mathcal{M}$ induced by states $\{q_i \mid i \in N\}$. We have $wgt(\mathcal{M}') = \sum_{i \in N} wgt(q_i) = \sum_{i \in N} w(i) \leq W$ and

$$\Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) \;=\; \sum_{i \in N} \frac{v(i)}{m\, n} \;=\; \frac{1}{m\, n} \sum_{i \in N} v(i) \;\geq\; \frac{V}{m\, n}.$$

For the other direction, let $\mathcal{M}'$ be a witnessing subsystem of $\mathcal{M}$ for $\Pr_{\mathcal{M}}(\Diamond \text{target}) \geq V/(m\, n)$ satisfying $wgt(\mathcal{M}') \leq W$. Let $N \subseteq \{1, \ldots, n\}$ be such that $\{s_{in}\} \cup \{q_i \mid i \in N\}$ are the reachable states of $\mathcal{M}'$. We claim that N is a solution of the knapsack instance. From $\Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) \geq V/(m\, n)$ we can deduce $\sum_{i \in N} v(i) \geq V$ and from $wgt(\mathcal{M}') \leq W$ we get $\sum_{i \in N} w(i) \leq W$.

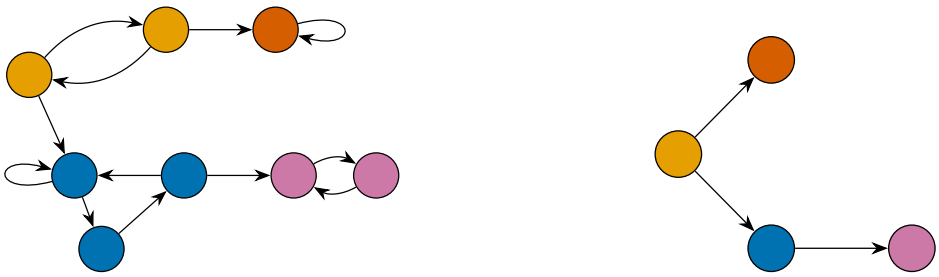
2.) We now give a polynomial reduction from the clique problem to the labeled witness problem for tree structured Markov chains. The clique problem takes as input a graph $G = (V, E)$ and a natural number $C \geq 0$ and asks whether G has a clique of size C. To solve it, we construct a Markov chain $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, s_{in}, P)$, where $S = \{s_{in}\} \cup E$. The transition probabilities are defined by $P(s_{in}, e) = 1/|E|$ and $P(e, \text{target}) = 1$ for all $e \in E$. Additionally, we label each state $e \in S \setminus \{s_{in}\}$ by the vertices of G that participate in edge e. More precisely, let $L = V$ and define the labeling function $\Lambda : S \to 2^L$ by $\Lambda(s_{in}) = \emptyset$ and $\Lambda\big(\{u, v\}\big) = \{u, v\}$.

We claim that G has a clique of size C if and only if $\mathcal{M}$ has a subsystem $\mathcal{M}'$ satisfying

$$|\,\text{labels}(\mathcal{M}')| \;\leq\; C \qquad \text{and} \qquad \Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) \;\geq\; \frac{C(C-1)}{2|E|}.$$

Recall that $C(C-1)/2$ is the number of edges in a clique of size C. For the direction from left to right, let $V' \subseteq V$ be a clique of G of size C. Consider the subsystem $\mathcal{M}'$ of $\mathcal{M}$ induced by the set of labels V'. Hence, $|\,\text{labels}(\mathcal{M}')| = |V'| = C$ and, as V' is a clique, we have $\Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) = \frac{C(C-1)}{2|E|}$.

For the other direction, let $\mathcal{M}'$ be a subsystem of $\mathcal{M}$ such that $|\,\text{labels}(\mathcal{M}')| \leq C$ and $\Pr_{\mathcal{M}', s_{in}}(\Diamond \text{target}) \geq \frac{C(C-1)}{2|E|}$. It follows that $\mathcal{M}'$ must include $C(C-1)/2$ states $e \in E$, as otherwise it would not achieve this reachability probability. On the other hand, $\mathcal{M}'$ touches at most C labels. Hence, the $C(C-1)/2$ states that $\mathcal{M}'$ includes induce as many edges in G but with only C participating vertices. So, $\text{labels}(\mathcal{M}')$ induces a set of vertices of G of size C with $C(C-1)/2$ edges in between them. But then, G has a clique of size C. □



(a) (b)

Figure 5.3: An example graph G (a) together with a tree partition induced by the coloring. The quotient graph under this partition is a tree and is shown in (b). The directed tree-partition width of G is three, because the presented tree partition is optimal and its largest block is of size three.

5.2 Directed tree- and path-partition width

This section introduces the *directed tree-partition width* and the *directed path-partition width*. They correspond to the existing notions of tree- and path-partition width for undirected graphs [See85, Ede86, Woo09]. In what follows, let $G = (V, E)$ be a fixed finite directed graph. For a given partition $\mathcal{P} = \{V_1, \dots, V_n\}$ of V we define the *quotient of* G under $\mathcal{P}$ to be the directed graph $G_{\mathcal{P}} = (\mathcal{P}, E_{\mathcal{P}})$, where $(V_i, V_j) \in E_{\mathcal{P}}$ if and only if $i \neq j$ and there exist $v \in V_i, v' \in V_j$ such that $(v, v') \in E$.

Definition 5.7 (Directed tree partition). A partition $\mathcal{P} = \{V_1, \dots, V_n\}$ of V is a *directed tree partition* of G if the quotient of G under $\mathcal{P}$ is a tree. We denote by $\mathsf{DTP}(G)$ the set of directed tree partitions of G.

We will call $\max_{S \in \mathcal{P}} |S|$ the *width* of a partition $\mathcal{P}$, henceforth denoted by $\mathrm{width}(\mathcal{P})$. The directed tree-partition width of a graph is now defined as the minimal width of all tree partitions of the graph.

Definition 5.8 (Directed tree-partition width (dtpw)). The *directed tree-partition width* of graph G is defined as

$$dtpw(G) = \min_{\mathcal{P} \in \mathsf{DTP}(G)} \mathrm{width}(\mathcal{P}).$$

Replacing *tree* by *path* in the above definitions yields the notions of *directed path partition* and *directed path-partition width* (dppw). An example of a graph and a tree partition is given in Figure 5.3.

Relation to other notions for directed graphs. While the theory of tree width and related notions for undirected graphs is very mature, the quest for analogous parameters for directed graphs is still open [GHK+16]. One option is to simply use the standard tree width of the underlying undirected graph. A notion called *directed tree width* has been proposed in [JRST01], and [Ree99] introduces a very similar parameter with the same name which differs by at most one from the directed tree width defined in [JRST01]. We will use directed tree width to refer to the notion defined in [JRST01]. Another parameter called *D-width* is studied in [Saf05].

If G_u is an undirected graph, then its tree width equals the directed tree width and D-width of the directed graph one gets by including both edges (u, v) and (v, u) whenever u and v are connected in G_u. This is not true for the directed tree-partition width. In fact, here any strongly connected component of a graph needs to be included in a single block of the partition. Hence, the directed tree-partition width is lower-bounded by the size of any SCC of the graph. The following proposition confirms that directed tree-partition width is stronger than the width parameters from [JRST01] and [Saf05], in the sense that a class with bounded dtpw is bounded with respect to the other parameters as well.

Proposition 5.9. *If a class C of finite directed graphs has bounded directed tree-partition width, then C has bounded directed tree width, bounded D-width, bounded undirected tree width and bounded undirected tree-partition width.*

Proof. Let $G = (V, E)$ be a directed graph and G_u be its underlying undirected graph. Let us denote by $\mathrm{utw}(G)$ the (undirected) tree width of G_u, by $\mathrm{dtw}(G)$ the directed tree width of G, by $\mathrm{Dw}(G)$ its D-width and by $\mathrm{utpw}(G)$ the undirected tree-partition width of G_u.

Undirected tree (partition) width. A directed tree partition of G directly yields a tree partition of G_u of the same size. Tree partitions for undirected graphs are defined analogously to Definition 5.7, see [Woo09]. It follows that $\mathrm{utpw}(G) \leq \mathrm{dtpw}(G)$. It was shown in [See85, Fact 2.] that $2 \cdot \mathrm{utpw}(G) \geq \mathrm{utw}(G) + 1$. Hence, in particular, we have $\mathrm{utw}(G) \leq 2 \cdot \mathrm{dtpw}(G) - 1$.

D-width. The D-width of G is defined using *d-decompositions* (see [Saf05]), which are pairs (T, X) where T is a tree and X is a function which labels the nodes of T by subsets of V such that

1. all vertices of G appear in at least one of the sets and
2. for every strongly connected component $\mathcal{S}$ of G the nodes t of T such that $X(t) \cap \mathcal{S} \neq \emptyset$ form a connected subtree of T.

Clearly, a directed tree partition satisfies this property as every strongly connected component needs to be contained in a single block. Hence every directed tree partition induces a d-decomposition of the same size, which implies $\mathrm{Dw}(G) \leq \mathrm{dtpw}(G)$.

Directed tree width. It is shown in [Saf05, Corollary 1.] that the directed tree width of any graph is smaller than its D-width, that is we have $\mathrm{dtw}(G) \leq \mathrm{Dw}(G)$. Hence, it follows from the discussion on D-width that $\mathrm{dtw}(G) \leq \mathrm{dtpw}(G)$ holds. □

Computing the directed tree partition width. Our next aim is to show that the problem of deciding whether a directed tree partition exists whose maximally sized block is bounded by a given number k is NP-complete. The analogous statement holds also in the undirected case (see [Ede86, Theorem 2.2]). We reduce from the *oneway bisection problem* [FY03] for directed graphs, which asks whether there exists a partition of a given graph into two equally-sized vertex sets V_0, V_1 such that all edges go from V_0 to V_1.

Proposition 5.10. *The two problems*

1. *given a directed graph G and $k \in \mathbb{N}$, decide whether* $\mathrm{dtpw(G)} \leq k$ *hold, and*
2. *given a directed graph G and $k \in \mathbb{N}$, decide whether* $\mathrm{dppw(G)} \leq k$ *holds*

are both NP-complete.

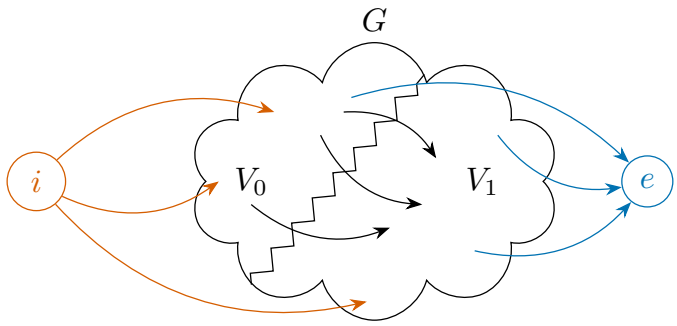


Figure 5.4: A sketch for the reduction from the oneway bisection problem to the problem of computing a directed tree or path partition of certain size. Two fresh states i and e are added to the given directed graph G, and all vertices of G get an additional incoming edge from i and an outgoing edge to e.

Proof. For membership in NP observe that one can guess a partition $\mathcal{P}$ and check whether it is a directed tree partition (resp. directed path partition) satisfying $\text{width}(\mathcal{P}) \leq k$.

To show NP-hardness we describe a polynomial reduction from the *oneway bisection problem* of directed graphs, which was shown to be NP-hard in [FY03]. It asks, given a directed graph G, whether there exists a bisection V_0, V_1 of the vertices of G (that is, a partition of the vertices into V_0 and V_1 satisfying $|V_0| = |V_1|$) such that there are no directed edges from V_1 to V_0. To reduce this problem to the question of whether the directed path-partition width is at most k, let us fix a graph $G = (V, E)$. We construct a new graph $G' = (V \cup \{i, e\}, E')$ with fresh vertices i, e and edges defined by $E' = E \cup \{(i, v), (v, e) \mid v \in V\}$. A sketch of the construction is given in Figure 5.4. We claim that

$$\text{dppw}(G') \leq 1 + \frac{|V|}{2} \quad \text{if and only if} \quad G \text{ has a oneway bisection.}$$

Suppose first that G has a oneway bisection V_0, V_1. Then $(\{i\} \cup V_0, \{e\} \cup V_1)$ is a directed path partition of G'. This follows directly from the fact that there is no directed edge from V_1 to V_0. The width of this path partition is $1 + (|V|/2)$, as $|V_0| = |V_1| = |V|/2$.

For the other direction, we first observe that any directed path partition of G' has length between one and three. This can be seen as follows. Vertex e must appear in one of the first three blocks, as any vertex of G' has a path to e of length at most three. This also implies that all vertices must be part of a block which either contains e, or precedes the block containing e.

We now show that a path partition of G' with width at most $1 + (|V|/2)$ has length two. It clearly cannot have length one, so suppose that it has length three. Then the first block, which must include i, cannot include any other vertex $v \in V$. This is because then e must be contained in the first or second block, as there exists an edge from v to e. In both cases, the third block remains empty. At the same time, no vertex $v \in V$ can be included in the third block, as it is reachable from i in a single step. Hence the second block contains all $|V|$ vertices, contradicting the fact that the width is at most $1 + (|V|/2)$.

So take a path partition of length two with width at most $1 + (|V|/2)$. Then, the two blocks have exactly $1 + (|V|/2)$ elements, and hence $|V|/2$ vertices from V respectively. This partition of V induces a oneway bisection of G as there cannot be any directed edges from the second block to the first one.

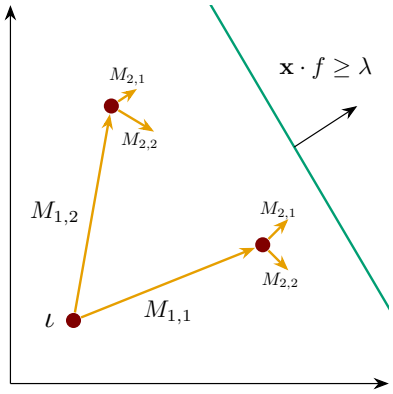


Figure 5.5: A geometric interpretation of the matrix-pair chain problem with dimension $d = 2$. Starting with ι, one of the matrices $M_{i,1}$ and $M_{i,2}$ is multiplied from the right to the current point in each round i. The goal is to generate a point after n rounds which lies in the halfspace defined by $\mathbf{x} \cdot f \geq \lambda$.

As e is reachable from all vertices, the only directed tree partitions of the graph G' in the above reductions are directed path partitions. It follows directly that deciding whether $\mathrm{dtpw}(G) \leq k$ holds is also NP-hard. □

5.3 The witness problem for Markov chains with bounded path width

We have seen that the weighted witness problem for tree structured Markov chains is solvable in polynomial time if the weights are encoded in unary (Proposition 5.4). A natural question is whether these ideas extend to Markov chains which are *similar* to trees, or, more formally, to Markov chains having low width with respect to one of the tree similarity measures discussed in the previous section.

We will now show that the witness problem for Markov chains with directed path-partition width of at most six is already NP-hard. It follows that the problem is also NP-hard for Markov chains with bounded directed tree-partition width, and, by Proposition 5.9, for Markov chains of bounded directed tree width and D-width. This shows that, unfortunately, we cannot expect an efficient algorithm parametrized by tree or path-similarity for witness problem. The proof will expose a source of combinatorial hardness in the witness problem which is quite different from what we have seen so far. To capture it we introduce the *matrix-pair chain problem*.

Definition 5.11 (d-dimensional matrix-pair chain problem). The d-dimensional matrix-pair chain problem (d-MCP) takes as input a sequence $(M_{1,1}, M_{1,2}), \ldots, (M_{n,1}, M_{n,2})$, where $M_{i,j} \in \mathbb{Q}^{d \times d}$, a starting vector $\iota \in \mathbb{Q}^{1 \times d}$, final vector $f \in \mathbb{Q}^{d \times 1}$, and $\lambda \in \mathbb{Q}$ (with all numbers encoded in binary) and asks whether there exists a tuple $(\sigma_1, \ldots, \sigma_n) \in \{1, 2\}^n$ such that

$$\iota \cdot M_{1,\sigma_1} \cdots M_{n,\sigma_n} \cdot f \geq \lambda.$$

We call n the *length* of an MCP instance. The *nonnegative* variant of the problem restricts all input numbers to be nonnegative.

The problem can also be described using the following game of n rounds (see Figure 5.5). We start with vector ι, and in the first round choose one of the matrices $M_{1,1}$ and $M_{1,2}$. The chosen matrix is multiplied to ι from the right, generating a new point. We continue generating points in this way for n rounds. If the final point $\mathbf{p}$ (which we get after applying one of the

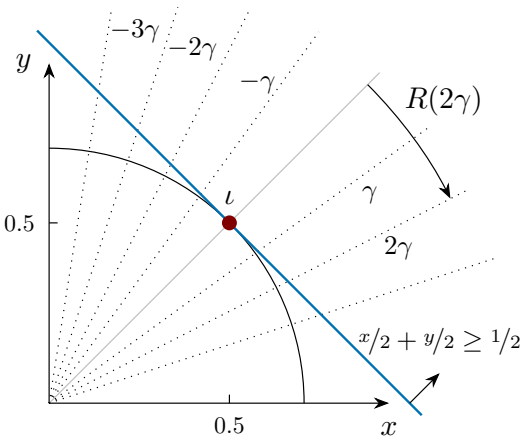
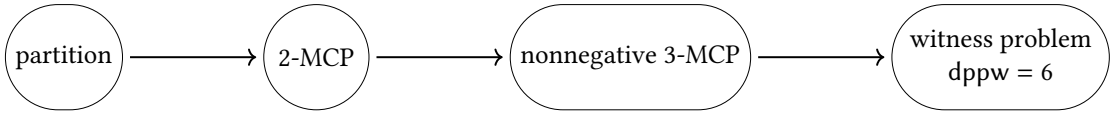


Figure 5.6: A sketch for the reduction from partition to the 2-MCP (Proposition 5.12). It associates to each number s in the partition problem a matrix pair where one matrix rotates clockwise by $\gamma \cdot s$ and the other rotates counterclockwise by $\gamma \cdot s$. Then the partition problem has a solution iff a matrix from each pair can be chosen such that the joint rotation is zero. This, in turn, is true iff the final point lies in the halfspace defined by the blue line.

matrices $M_{n,1}$ and $M_{n,2}$) satisfies $\mathbf{p} \cdot f \geq \lambda$ (i.e., lies in the halfspace defined by $\mathbf{x} \cdot f \geq \lambda$) we win. The question is whether there is a winning strategy in this game.

The d-MCP is in NP for any $d \in \mathbb{N}$, as one can guess one matrix from each pair and verify that the corresponding product is greater or equal to λ. We now show NP-hardness of the 2-MCP by a reduction from the *partition* problem, and then use this result to show NP-hardness of the nonnegative 3-MCP. Finally, we reduce the nonnegative 3-MCP to the witness problem for Markov chains of directed path-partition width at most six. The following picture summarizes the chain of polynomial reductions that we describe:



5.3.1 Hardness of the matrix-pair chain problem

NP-hardness of the 2-MCP. To show NP-hardness of the 2-MCP we reduce from the *partition* problem, which is another classical NP-complete problem [Kar72]. Given a finite set $S = \{s_1, \ldots, s_n\} \subseteq \mathbb{Z}$, whose elements are encoded in binary, it asks to decide whether there exists $W \subseteq S$ such that $\sum W = \sum(S \setminus W)$. Here we abbreviate $\sum X = \sum_{x \in X} x$. For the reduction to the 2-MCP we relate each element s_i to a pair of matrices $M_{i,1}, M_{i,2}$ where $M_{i,1}$ realizes a clockwise rotation by an angle proportional to s_i, and $M_{i,2}$ realizes the counter-clockwise rotation by the same angle. We do this in a way which guarantees that for all $\sigma_1, \ldots, \sigma_n \in \{1, 2\}^n$ we have that $W = \{s_i \mid \sigma_i = 1\}$ satisfies $\sum W = \sum(S \setminus W)$ if and only if the product $M_{1,\sigma_1} \cdots M_{n,\sigma_n}$ equals the identity matrix.

Additionally, we choose initial vector $\iota = (1/2, 1/2)$, final vector $f = (1/2, 1/2)^T$ and threshold $\lambda = 1/2$. All matrices one can generate as products of $M_{i,1}, M_{i,2}$, with $1 \leq i \leq n$, are rotation matrices and the only point on the circle around the origin with radius $1/\sqrt{2}$ that satisfies $\mathbf{x} \cdot f \geq 1/2$ is ι. Hence, the only way to generate a point which meets the threshold condition is to make sure that the product of matrices equals the rotation by zero, i.e., the identity matrix. And, by construction, this will only be possible if the partition problem is a yes-instance. Figure 5.6 illustrates the idea.

Proposition 5.12. *The two-dimensional matrix-pair chain problem (2-MCP) is NP-hard.*

Proof. We describe a polynomial reduction from the partition problem [Kar72]. Let $S = \{s_1, \dots, s_n\} \subseteq \mathbb{Z}$ be an instance of it, where the numbers are encoded in binary, and let $m = \max\{\sum S \cap \mathbb{Z}_{>0}, -\sum S \cap \mathbb{Z}_{<0}\}$ be the maximal absolute value that can be accumulated by a subset of S. Throughout the proof we will denote by $R(\theta) \in \mathbb{R}^{2\times 2}$ the rotation matrix which realizes the rotation by θ, where $\theta \in \mathbb{Q}$ represents an angle in radian. Let $R(\gamma)$ be a rational rotation matrix which rotates by an angle in radian of $\gamma < \pi/(4m)$. Such a matrix can be computed using the results of [CDR92], which shows that for any by angle φ and $\epsilon \in \mathbb{Q}_{>0}$, a rotation matrix $R(\theta)$ with rational entries and such that $|\varphi - \theta| < \epsilon$ holds can be computed in time polynomial in $\log(1/\epsilon)$. The assumption $\gamma < \pi/(4m)$ implies that the total rotation in our construction cannot exceed $\pi/4$. For an integer $a \in \mathbb{Z}$ the rotation by $a \cdot \gamma$ is given by the matrix $R(a \cdot \gamma) = \big(R(\gamma)\big)^a$.

For all $1 \le i \le n$ we first define the pair of matrices $M_{i,1}, M_{i,2} \in \mathbb{R}^{2\times 2}$ by $M_{i,1} = R(\gamma \cdot s_i)$ and $M_{i,2} = R(-\gamma \cdot s_i)$. It follows that for all $\sigma_1, \dots, \sigma_n \in \{1,2\}^n$ and $W = \{s_i \mid \sigma_i = 1\}$ we have

$$M_{1,\sigma_1} \cdots M_{n,\sigma_n} \;=\; \prod_{\substack{i \text{ s.t.}\\ \sigma_i=1}} R(\gamma \cdot s_i) \cdot \prod_{\substack{i \text{ s.t.}\\ \sigma_i=2}} R(-\gamma \cdot s_i) \;=\; R(\gamma a_1) \cdot R(-\gamma a_2) \;=\; R(\gamma(a_1 - a_2)),$$

where $a_1 = \sum_{\sigma_i=1} s_i$ and $a_2 = \sum_{\sigma_i=2} s_i$. As $R(0) = I$ (the zero rotation is the identity matrix) and γ was chosen such that $m \cdot \gamma < \pi/4$, which means that no product of n matrices $M_{i,1}, M_{i,2}$ realizes a rotation by more than 45 degrees, the following holds for all $\sigma_1, \dots, \sigma_n \in \{1,2\}^n$ and $W = \{s_i \mid \sigma_1 = 1\}$:

$$\sum_{x \in W} x \;=\; \sum_{x \in (S \setminus W)} x \qquad \text{if and only if} \qquad M_{1,\sigma_1} \cdots M_{n,\sigma_n} = I. \tag{$*$}$$

Now let $\iota = (1/2, 1/2)$, $f = (1/2, 1/2)^T$ and $\lambda = 1/2$. The only point $\mathbf{p}$ which can be reached by a rotation from ι and which satisfies $\mathbf{p} \cdot f \ge \lambda$ is ι itself (see Figure 5.6). Hence, the constructed MCP is a yes-instance iff we find $\sigma_1, \dots, \sigma_n \in \{1,2\}^n$ such that $M_{1,\sigma_1} \cdots M_{n,\sigma_n} = I$. But, by $(*)$, this holds iff the instance of the partition problem we started with is a yes-instance. □

NP-hardness of the nonnegative 3-MCP. To use rotation matrices in the above proof it is crucial that negative numbers are allowed in the MCP. We now show how the 2-MCP can be reduced in polynomial time to the *nonnegative* 3-MCP. Nonnegativity will be important for our final reduction to the witness problem. The main idea is to map each two-dimensional matrix to a nonnegative three-dimensional matrix which preserves the original dynamics when projected onto a certain two-dimensional subspace. A graphical illustration of this idea is given in Figure 5.7.

More formally, let $(M_{1,1}, M_{1,2}), \dots, (M_{n,1}, M_{n,2})$, with $M_{i,j} \in \mathbb{Q}^{2\times 2}$, together with $\iota \in \mathbb{Q}^{1\times 2}$, $f \in \mathbb{Q}^{2\times 1}$, $\lambda \in \mathbb{Q}$ be an instance of the 2-MCP. For some $\kappa \in \mathbb{Q}_{\ge 0}$, we define

$$N_{i,j} = B \begin{pmatrix} M_{i,j} & \mathbf{0} \\ \mathbf{0} & \kappa \end{pmatrix} B^{-1}, \quad \iota' = \begin{pmatrix} \iota & \kappa \end{pmatrix} B^{-1}, \quad f' = B \begin{pmatrix} f \\ \kappa \end{pmatrix} \quad \text{and} \quad \lambda' = \lambda + \kappa^{n+2} \tag{5.2}$$

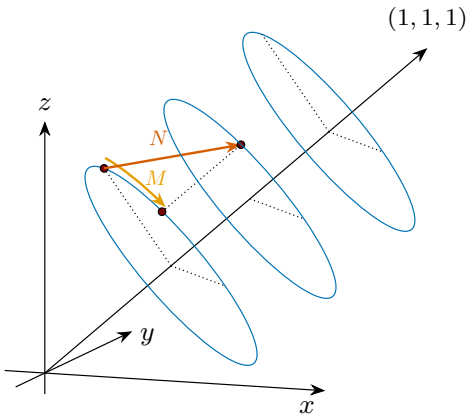


Figure 5.7: A picture for the reduction from 2-MCP to nonnegative 3-MCP. Each two-dimensional matrix M appearing of the 2-MCP instance is mapped to a three-dimensional matrix N which preserves the dynamics of M under projection and makes a step towards $(1,1,1)$. If the step is large enough, then N is nonnegative.

where we use the matrix

$$B = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \end{pmatrix} \qquad \text{with inverse} \qquad B^{-1} = 1/6 \cdot \begin{pmatrix} 3 & -3 & 0 \\ 1 & 1 & -2 \\ 2 & 2 & 2 \end{pmatrix}$$

to change the basis. The columns of B are orthogonal to each other and the third standard basis vector is mapped to $(1,1,1)$ under the change of basis. The proof of the following proposition shows that the 2-MCP instance is a yes-instance if and only if the constructed 3-MCP instance is as well. By choosing κ large enough, we furthermore can make sure that all matrices $N_{i,j}$ are nonnegative.

Proposition 5.13. *The nonnegative three-dimensional matrix-pair chain problem (nonnegative 3-MCP) is NP-hard.*

Proof. The proof goes by reduction from the 2-MCP. Let $(M_{1,1}, M_{1,2}), \ldots, (M_{n,1}, M_{n,2})$ be a sequence of rational 2×2 matrices, $\iota \in \mathbb{Q}^{1\times 2}$, $f \in \mathbb{Q}^{2\times 1}$ and $\lambda \in \mathbb{Q}_{\geq 0}$. For any $\kappa \geq 0$, consider the matrices $(N_{1,1}, N_{1,2}), \ldots, (N_{n,1}, N_{n,2})$ as defined in Equation (5.2), together with vectors ι', f' and rational λ'. Then, for any $\sigma_1, \ldots, \sigma_n \in \{1,2\}^n$ we have

$$\begin{aligned} N_{1,\sigma_1} \cdots N_{n,\sigma_n} &= B \begin{pmatrix} M_{1,\sigma_1} & \mathbf{0} \\ \mathbf{0} & \kappa \end{pmatrix} B^{-1} \cdot B \begin{pmatrix} M_{2,\sigma_2} & \mathbf{0} \\ \mathbf{0} & \kappa \end{pmatrix} B^{-1} \cdots B \begin{pmatrix} M_{n,\sigma_n} & \mathbf{0} \\ \mathbf{0} & \kappa \end{pmatrix} B^{-1} \\ &= B \begin{pmatrix} M_{1,\sigma_1} \cdots M_{n,\sigma_n} & \mathbf{0} \\ \mathbf{0} & \kappa^n \end{pmatrix} B^{-1} \end{aligned}$$

Applying initial and final weights yields:

$$\begin{aligned} \iota' \cdot B \begin{pmatrix} M_{1,\sigma_1} \cdots M_{n,\sigma_n} & \mathbf{0} \\ \mathbf{0} & \kappa^n \end{pmatrix} B^{-1} \cdot f' &= \begin{pmatrix} \iota & \kappa \end{pmatrix} \cdot B^{-1} \cdot B \begin{pmatrix} M_{1,\sigma_1} \cdots M_{n,\sigma_n} & \mathbf{0} \\ \mathbf{0} & \kappa^n \end{pmatrix} B^{-1} \cdot B \cdot \begin{pmatrix} f \\ \kappa \end{pmatrix} \\ &= \iota \cdot M_{1,\sigma_1} \cdots M_{n,\sigma_n} \cdot f + \kappa^{n+2} \end{aligned}$$

As a consequence, we have

$$\iota \cdot M_{1,\sigma_1} \cdots M_{n,\sigma_n} \cdot f \;\geq\; \lambda \quad \Longleftrightarrow \quad \iota' \cdot N_{1,\sigma_1} \cdots N_{n,\sigma_n} \cdot f' \;\geq\; \lambda + \kappa^{n+2} \;=\; \lambda'.$$

It remains to find κ such that all matrices $N_{i,j}$ and vectors ι', f' as defined in Equation (5.2) are

nonnegative. To this end, we show that $N_{i,j}$ can be written as

$$N_{i,j} = A_{i,j} + \frac{2\kappa}{6} \cdot \mathbf{1}^{3\times 3},$$

where $\mathbf{1}^{3\times 3}$ is the three times three matrix containing just ones. Let us expand the definition of $N_{i,j}$. Here we will assume that $M_{i,j}$ consists of elements $a, b, c, d \in \mathbb{Q}$.

$$\begin{aligned} N_{i,j} &= B\begin{pmatrix} M_{i,j} & \mathbf{0} \\ \mathbf{0} & \kappa \end{pmatrix} B^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 0 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & \kappa \end{pmatrix} \cdot 1/6 \begin{pmatrix} 3 & -3 & 0 \\ 1 & 1 & -2 \\ 2 & 2 & 2 \end{pmatrix} \\ &= 1/6 \begin{pmatrix} 3(a+c)+b+d & -3(a+c)+b+d & -2(b+d) \\ 3(c-a)-b+d & -3(c-a)-b+d & -2(d-b) \\ -6c-2d & 6c-2d & 4d \end{pmatrix} + 1/6 \begin{pmatrix} 2\kappa & 2\kappa & 2\kappa \\ 2\kappa & 2\kappa & 2\kappa \\ 2\kappa & 2\kappa & 2\kappa \end{pmatrix} \end{aligned}$$

Now we can define $A_{i,j}$ as the first matrix in the last sum. Then, $N_{i,j}$ is nonnegative if $2\kappa/6$ is larger than any entry in $A_{i,j}$. Observe that the entries of $A_{i,j}$ are all polynomial in the entries of $M_{i,j}$. The vectors ι' and f' have a similar structure. This implies that we can compute a κ in polynomial time such that all matrices and vectors defined in Equation (5.2) are nonnegative, which concludes the proof. □

The MCP for nearly equally valued matrices. As a last observation for the MCP, we show that the nonnegative 3-MCP remains hard even if we assume that the entries in all its matrices are very similar in terms of their value. For any function $\epsilon_n : \mathbb{N} \to \mathbb{Q}$ and rational number $C \in \mathbb{Q}$ we call an MCP instance of length n (i.e., containing n matrix pairs) (C, ϵ_n)-*equally valued* if all numbers that it contains apart from the threshold λ are in the range $[C - \epsilon_n, C]$.

Lemma 5.14. *Let $a \in \mathbb{Q}_{>0}$ and $C \in \mathbb{Q}_{\geq 0}$ be fixed and $\epsilon_n = a^{-n}$. Then, the (C, ϵ_n)-equally valued nonnegative 3-MCP is NP-hard.*

Proof. We show that the reduction given in Proposition 5.13 can be adapted to produce (C, ϵ_n)-equally valued matrices. There, we started with a 2-MCP instance $(M_{1,1}, M_{1,2}), \ldots, (M_{n,1}, M_{n,2}), \iota, f$ and λ, and constructed a 3-MCP instance of the form

$$N_{i,j} = A_{i,j} + \frac{2\kappa}{6} \cdot \mathbf{1}^{3\times 3}, \qquad f' = f + \frac{2\kappa}{6} \cdot \mathbf{1}^{3}, \qquad \iota' = \iota + \frac{2\kappa}{6} \cdot \mathbf{1}^{3}, \qquad \lambda' = \lambda + \kappa^{n+2},$$

such that for all $\kappa \in \mathbb{Q}$ and $\sigma_1, \ldots, \sigma_n \in \{1, 2\}^n$ we have

$$\iota \cdot M_{1,\sigma_1} \cdots M_{n,\sigma_n} \cdot f \;\geq\; \lambda \quad \Longleftrightarrow \quad \iota' \cdot N_{1,\sigma_1} \cdots N_{n,\sigma_n} \cdot f' \;\geq\; \lambda'.$$

From this, we construct a (C, ϵ_n)-equally valued instance. Let $\kappa' = 2\kappa/6$ and $v_{\max}$ and $v_{\min}$ be the maximal and minimal values appearing in $A_{i,j}, f$ and ι. Define

$$N^*_{i,j} = \frac{C}{v_{\max} + \kappa'} \cdot N_{i,j}, \qquad f^* = \frac{C}{v_{\max} + \kappa'} \cdot f', \quad \text{and} \qquad \iota^* = \frac{C}{v_{\max} + \kappa'} \cdot \iota'.$$

The largest value appearing in any of the matrices $N^*_{i,j}, f^*$ and ι^* is C, and the smallest one is $(C(v_{\min} + \kappa'))/(v_{\max} + \kappa')$. Consequently, the largest difference between any two entries of these matrices is $C(v_{\max} - v_{\min})/(v_{\max} + \kappa')$. The resulting MCP is (C, ϵ_n)-equally valued if this

difference is upper bounded by ϵ_n. This is ensured if κ' satisfies:

$$\kappa' \geq \frac{C(v_{\max} - v_{\min})}{\epsilon_n} - v_{\max}.$$

As $\epsilon_n = a^{-n}$ for some constant $a > 0$, we can compute κ' (and thereby κ), satisfying this equation in polynomial time. We assume, w.l.o.g., that this κ' is larger than $v_{\max}$. If this is not true, then we can define κ' to be $v_{\max}$. This makes sure that the resulting MCP is nonnegative.

Finally, we choose $\lambda^* = \left(\frac{C}{v_{\max}+\kappa'}\right)^{n+2} \cdot \lambda'$. Then, for all $\sigma_1, \ldots, \sigma_n \in \{1,2\}^n$ we have

$$\begin{aligned} & \iota^* \cdot N^*_{1,\sigma_1} \cdots N^*_{n,\sigma_n} \cdot f^* \;\geq\; \lambda^* \\ \iff \quad & \left(\frac{C}{v_{\max}+\kappa'}\right)^{n+2} \cdot \iota' \cdot N_{1,\sigma_1} \cdots N_{n,\sigma_n} \cdot f' \;\geq\; \left(\frac{C}{v_{\max}+\kappa'}\right)^{n+2} \cdot \lambda' \\ \iff \quad & \iota' \cdot N_{1,\sigma_1} \cdots N_{n,\sigma_n} \cdot f' \;\geq\; \lambda', \end{aligned}$$

which completes the reduction. □

5.3.2 Hardness of the witness problem

This section describes a polynomial reduction from the nonnegative 3-MCP to the witness problem for Markov chains with bounded path-partition width. Let $(M_{1,1}, M_{1,2}), \ldots, (M_{n,1}, M_{n,2})$, ι, f and λ be an instance of the nonnegative 3-MCP. For technical reasons explained later, we assume that all entries of the input matrices and vectors are in the range $[1/12 - \epsilon, 1/12]$ for some ϵ that satisfies:

$$0 < 12\epsilon < 1/2 \cdot \left(1/12 - \epsilon\right)^{n+2} \tag{5.3}$$

An ϵ which satisfies the bound is $\epsilon = 1/(24^{n+3})$, which can be seen as follows. First, we insert ϵ into the right hand side and derive a lower bound for the expression (assuming $n \geq 0$).

$$\frac{1}{2}\left(\frac{1}{12} - \epsilon\right)^{n+2} = \frac{1}{2}\left(\frac{1}{12} - \frac{1}{24^{n+3}}\right)^{n+2} = \frac{1}{2 \cdot 12^{n+2}}\left(1 - \frac{1}{2^{n+3} \cdot 12^{n+2}}\right)^{n+2} > \frac{1}{2 \cdot 12^{n+2}}\left(1 - \frac{1}{12^{n+2}}\right)^{n+2}.$$

This lower bound is greater than 12ϵ for all $n \geq 0$, as the following calculation shows.

$$\frac{1}{2 \cdot 12^{n+2}} \cdot \left(1 - \frac{1}{12^{n+2}}\right)^{n+2} > 12\epsilon \iff \left(1 - \frac{1}{12^{n+2}}\right)^{n+2} > \frac{1}{2^{n+2}} \iff \left(2 - \frac{2}{12^{n+2}}\right)^{n+2} > 1.$$

The last inequality holds for all $n \geq 0$.

We are allowed to make this assumption on the 3-MCP by Lemma 5.14, which shows that for any fixed $C \geq 0$ and $a > 0$ the (C, a^{-n})-equally valued nonnegative 3-MCP is NP-hard.

Structure of the reduction. The main idea of the reduction is to relate choices of matrices in the matrix-pair chain problem to choices of subsystems in the witness problem. The structure of the reduction is shown in Figure 5.9. It consists of n main layers, where the j-th layer includes two groups of states $\{x_{j,1}, y_{j,1}, z_{j,1}\}$ and $\{x_{j,2}, y_{j,2}, z_{j,2}\}$. Transitions between layers are formed using a matrix multiplication gadget, as drawn in Figure 5.8. More precisely, a double arrow labeled by matrix M in Figure 5.9 means that transitions as defined by the matrix multiplication gadget for M are included between the two groups of states. Transitions in the initial and final layer are defined analogously. For example, the transition from state x_{n+1} to "target" is assigned

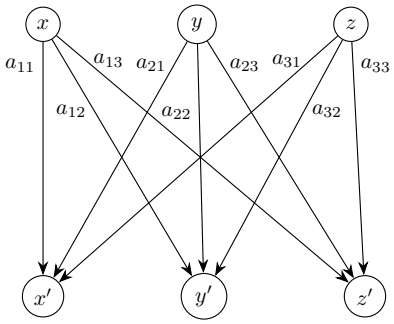


Figure 5.8: A gadget to encode matrix multiplication. Let M be a substochastic matrix with entries $(M)_{ij} = a_{ij} \in \mathbb{Q}_{\geq 0}$. If the probability of states (x', y', z') to reach some goal state is (v'_x, v'_y, v'_z), then these probabilities in states (x, y, z) are $M \cdot (v'_x, v'_y, v'_z)^T$.

probability f_x, which is the x-coordinate of vector f. All remaining probability is added to transitions to a state "exit", which are omitted in the figure.

By our assumption that all numbers appearing in matrices and vectors of the MCP instance are nonnegative and have at most value $1/12$, the construction yields a valid Markov chain, which we call $\mathcal{M}_1$. The directed path-partition width of $\mathcal{M}_1$ is six, and this does not depend on the MCP instance.

Lemma 5.15. *Let $\mathcal{M}_1$ be as defined above and assume that $n \geq 3$. We have $dppw(\mathcal{M}_1) = dtpw(\mathcal{M}_1) = 6$.*

Proof. Partitioning the states of $\mathcal{M}_1$ along the $n+1$ layers yields a directed path partition with width six. It remains to argue that there is no directed tree partition with a smaller width. Take any directed tree partition $\mathcal{P}$ of $\mathcal{M}_1$ and let $\{B_1, \dots B_m\}$ be its blocks. Pick arbitrary state from the main part of $\mathcal{M}_1$, for example $x_{i,1}$ for some $1 < i < n$, and let B_k be the block such that $x_{i,1} \in B_k$. As all successors of $x_{i,1}$ have a joint successor x_{n+1}, they either belong to B_k, or to some successor block B_{k+1} in the tree order. Similarly, all predecessors of $x_{i,1}$ belong either to B_k or to some predecessor block B_{k-1}. The same holds for all other states in the i-th layer, which means that in total 18 states are included in the sets B_{k-1}, B_k, B_{k+1}. But then, one of these three blocks needs to include at least six states, which shows that the width of $\mathcal{P}$ is at least six. □

Let $\text{left}_i = \{x_{i,1}, y_{i,1}, z_{i,1}\}$ and $\text{right}_i = \{x_{i,2}, y_{i,2}, z_{i,2}\}$. A subsystem of $\mathcal{M}_1$ is called *good* if it is induced by a set of states S' such that $\{s_{in}, x_{n+1}, y_{n+1}, z_{n+1}\} \subseteq S'$ and for all $1 \leq i \leq n$

$$\textit{either} \quad \text{left}_i \subseteq S' \text{ and } \text{right}_i \cap S' = \emptyset \qquad \textit{or} \qquad \text{right}_i \subseteq S' \text{ and } \text{left}_i \cap S' = \emptyset.$$

This means that S' "chooses" exactly one of the sets left_i and right_i for each layer $1 \leq i \leq n$. Good subsystems have size $3n+4$ (recall that target and exit are not counted in the size of a subsystem). Subsystems that are not good are called *bad*. Clearly, there is a one-to-one correspondence between good subsystems and matrix sequences in the matrix-pair chain problem. Given a sequence $\pi = \sigma_1, \dots, \sigma_n \in \{1, 2\}^n$, we define

$$S_\pi = \{s_{in}, x_{n+1}, y_{n+1}, z_{n+1}\} \cup \bigcup \left\{ \{x_{i,\sigma_i}, y_{i,\sigma_i}, z_{i,\sigma_i}\} \mid 1 \leq i \leq n \right\}.$$

We denote by $\mathcal{M}_\pi$ the induced subsystem of S_π in $\mathcal{M}_1$. The following lemma shows how the value of the product that arises in the MCP for matrix choices π corresponds to the probability of reaching "target" in $\mathcal{M}_\pi$.

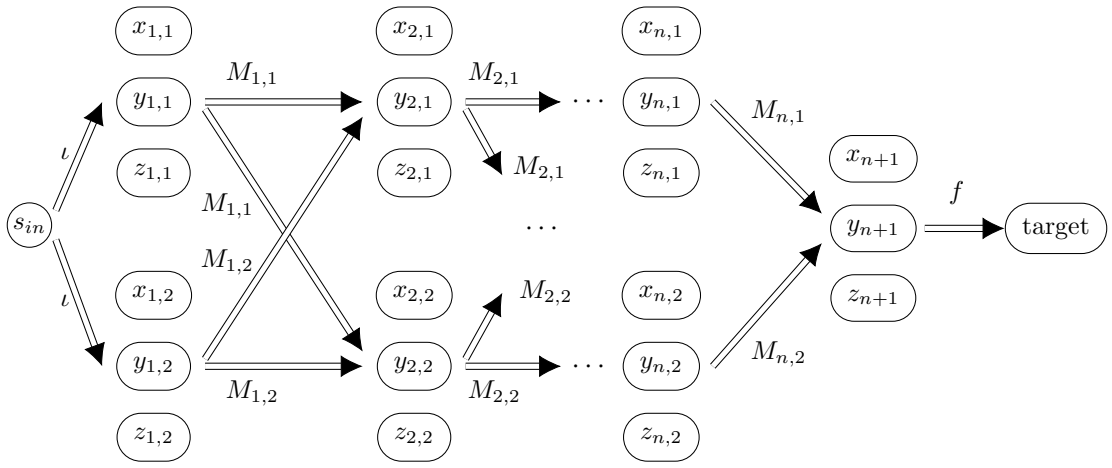


Figure 5.9: The Markov chain $\mathcal{M}_1$ constructed for nonnegative 3-MCP instance $(M_{1,1}, M_{1,2}), \ldots, (M_{n,1}, M_{n,2}), \iota$ and f. The doubled arrows represent instances of the matrix multiplication gadget as shown in Figure 5.8. A subsystem of $\mathcal{M}_1$ is called *good*, if it contains either $\{x_{j,1}, y_{j,1}, z_{j,1}\}$ or $\{x_{j,2}, y_{j,2}, z_{j,2}\}$ for each main layer j. Good subsystems correspond to choices of matrices in the MCP, and their probability to reach target corresponds to the value of the product achieved by the corresponding choice.

Lemma 5.16. *For all* $\pi = \sigma_1, \ldots, \sigma_n \in \{1, 2\}^n$ *we have* $\Pr_{\mathcal{M}_\pi}(\Diamond \text{target}) = \iota \cdot M_{1,\sigma_1} \cdots M_{n,\sigma_n} \cdot f$.

Proof. Let $v(s)$ be the probability of reaching "target" from state s in $\mathcal{M}_\pi$ and define for all $1 \leq j \leq n$: $(x_j, y_j, z_j) = (x_{j,1}, y_{j,1}, z_{j,1})$ if $\sigma_j = 1$, and else $(x_j, y_j, z_j) = (x_{j,2}, y_{j,2}, z_{j,2})$. We show by induction on i that

$$\begin{pmatrix} v(x_{n+1-i}) \\ v(y_{n+1-i}) \\ v(z_{n+1-i}) \end{pmatrix} = M_{n+1-i,\sigma_{n+1-i}} \cdots M_{n,\sigma_n} \cdot f.$$

This is enough, as $\Pr_{\mathcal{M}_\pi}(\Diamond \text{target}) = \iota \cdot (v(x_1), v(y_1), v(z_1))^T$. For $i = 0$ it is clear, as the probability vector to reach target from $(x_{n+1}, y_{n+1}, z_{n+1})$ is f. For $i = i' + 1$, we have

$$\begin{pmatrix} v(x_{n+1-i}) \\ v(y_{n+1-i}) \\ v(z_{n+1-i}) \end{pmatrix} = M_{n+1-i,\sigma_{n+1-i}} \cdot \begin{pmatrix} v(x_{n+1-i'}) \\ v(y_{n+1-i'}) \\ v(z_{n+1-i'}) \end{pmatrix}$$

by the fact that $x_{n+1-i'}, y_{n+1-i'}$ and $z_{n+1-i'}$ are the only states reachable from x_{n+1-i}, y_{n+1-i} and z_{n+1-i} in $\mathcal{M}_\pi$ by definition, and the transition probabilities between these groups of states are constructed using the matrix multiplication gadget for $M_{n+1-i,\sigma_{n+1-i}}$ (see Figure 5.8). □

It follows that the nonnegative 3-MCP reduces to deciding whether there exists a *good* subsystem whose probability to reach goal is at least λ. However, we have not ruled out yet that the 3-MCP instance is a no-instance, but there exists some *bad* subsystem of size $3n + 4$ which satisfies the threshold condition. We now show how the construction can be adapted to rule out this possibility.

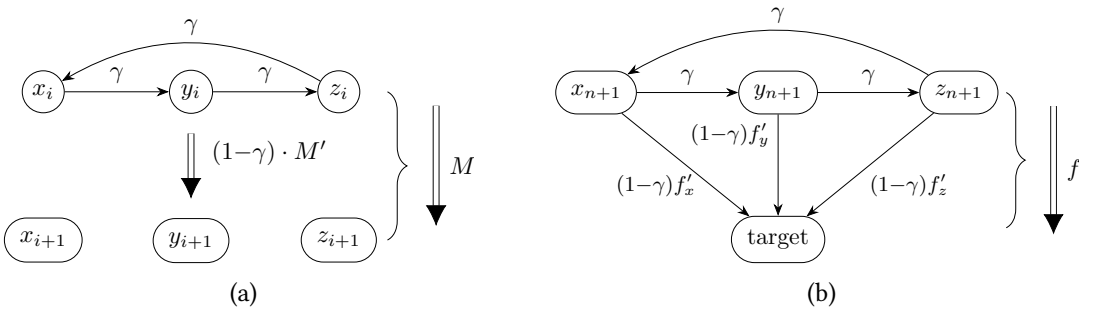


Figure 5.10: A γ-cycle is added to the upper states of the matrix multiplication gadget (see Figure 5.8) to make sure that removing any state on the cycle leads to a significant drop in probability. In (a) we see the construction used in all but the last layer, which is handled by the construction in (b). In (a), the matrix M' is chosen such that the probability of reaching $(x_{i+1}, y_{i+1}, z_{i+1})$ is $\theta \cdot M$, where θ is any initial distribution on states (x_i, y_i, z_i).

Interconnecting states. The idea is to make sure that bad subsystems have decisively less probability to reach "target", when compared with good subsystems. To this end we adapt the matrix multiplication gadget from Figure 5.8 such that removing any state leads to a large drop in probability. This is achieved by adding a cycle which connects the upper states, as shown in Figure 5.10a. The states x_i, y_i, z_i represent one of the triples $x_{i,1}, y_{i,1}, z_{i,1}$ or $x_{i,2}, y_{i,2}, z_{i,2}$, and likewise for $x_{i+1}, y_{i+1}, z_{i+1}$. The probability of staying inside the cycle is γ (whose precise value will be defined below) in each state. Transitions from states x_i, y_i, z_i to $x_{i+1}, y_{i+1}, z_{i+1}$ are given by an instance of the matrix multiplication gadget for $(1-\gamma)\cdot M'$. Our aim is to define M' such that the probabilities of moving from x_i, y_i, z_i to $x_{i+1}, y_{i+1}, z_{i+1}$ when including the γ-cycles is equal to some given matrix M.

The matrix which contains all the pairwise probabilities of reaching a state in $x_{i+1}, y_{i+1}, z_{i+1}$ from a state in x_i, y_i, z_i is given by the matrix $R \cdot M'$ defined as follows.

$$R \cdot M' \;=\; \underbrace{\frac{1-\gamma}{1-\gamma^3} \cdot \begin{pmatrix} 1 & \gamma & \gamma^2 \\ \gamma^2 & 1 & \gamma \\ \gamma & \gamma^2 & 1 \end{pmatrix}}_{R} M' \tag{5.4}$$

For example, the value $(R \cdot M')_{11}$ is equal to the probability of reaching x_{i+1} when starting in state x_i in Figure 5.10a, which can be checked by solving the corresponding equation system.

Let us assume that M is one of the matrices from the input MCP instance with entries $(M)_{lk} = a_{lk}$ for $1 \le l, k \le 3$. We want to find M' such that the gadget from Figure 5.10a realizes the matrix multiplication M. In other words, we want the probability to reach x_{i+1} from x_i to be exactly a_{11}, and similarly for the other states. Solving the equation $M = R \cdot M'$ for M' yields

$$M' \;=\; R^{-1} \cdot M \;=\; \frac{1}{1-\gamma} \begin{pmatrix} a_{11} - \gamma a_{21} & a_{12} - \gamma a_{22} & a_{13} - \gamma a_{23} \\ a_{21} - \gamma a_{31} & a_{22} - \gamma a_{32} & a_{23} - \gamma a_{33} \\ a_{31} - \gamma a_{11} & a_{32} - \gamma a_{12} & a_{33} - \gamma a_{13} \end{pmatrix} \tag{5.5}$$

To see this, we first compute the inverse of R, which is given by:

$$R^{-1} = \frac{1}{1-\gamma}\begin{pmatrix} 1 & -\gamma & 0 \\ 0 & 1 & -\gamma \\ -\gamma & 0 & 1 \end{pmatrix}$$

We choose γ to satisfy

$$12\epsilon < 1-\gamma < 1/2 \cdot \big(3(1/12-\epsilon)\big)^{n+2} \tag{5.6}$$

which is possible due to the assumption of Equation (5.3). To ensure that the construction yields a valid Markov chain, we first argue that all entries of M' are in the range $[0, 1/6]$. Here we use that the entries of M are assumed to be in the range $[1/12-\epsilon, 1/12]$. For any pair of entries a, a' of M we have

$$\frac{1}{1-\gamma}(a-\gamma a') \geq \frac{1}{1-\gamma}(1/12-\epsilon-\gamma/12) = 1/12 - \frac{\epsilon}{1-\gamma} > 0,$$

where the last inequality follows from $12\epsilon < 1-\gamma$. At the same time, we also have:

$$\frac{1}{1-\gamma}(a-\gamma a') \leq \frac{1}{1-\gamma}(1/12-\gamma(1/12-\epsilon)) = 1/12 + \frac{\gamma\epsilon}{1-\gamma} < 1/6,$$

where the last inequality follows from $\gamma < 1$ and $12\epsilon < 1-\gamma$, which is equivalent to $\epsilon/(1-\gamma) < 1/12$. The fact that $1/6$ is an upper bound on all entries of M' implies that using the gadgets from Figure 5.10 in the main reduction yields a valid Markov chain, as all states in Figure 5.9 have at most 6 outgoing edges. The derivation of values f'_x, f'_y and f'_z as used in Figure 5.10b is done in essentially the same way by setting $f' = R^{-1} \cdot f$.

We let $\mathcal{M}_2$ be the Markov chain which is obtained by using the gadgets defined in Figure 5.10 rather than Figure 5.8 to realize the matrix multiplications in the reduction shown in Figure 5.9. In particular, this means that all groups of states $x_{i,j}, y_{i,j}, z_{i,j}$, for $1 \leq i \leq n$ and $j \in \{1,2\}$, are now connected with a γ-cycle.

As the new matrix multiplication gadget correctly encodes the desired matrix multiplication by M (due to our choice of M', see Equation (5.5)), good subsystems (which are defined in the same way as for $\mathcal{M}_1$) have the same probability to reach "target" in $\mathcal{M}_1$ and $\mathcal{M}_2$. The main point of adding γ-cycles is to make sure that if one state which participates in such a cycle is excluded in a subsystem, then the probability of any participating state to reach the next layer drops significantly. This lets us derive an upper bound on the probability achievable in bad subsystems.

On the other hand, our assumption that all entries of matrices $M_{i,j}$ (with $1 \leq i \leq n$ and $j \in \{1,2\}$) and vectors ι, f have value at least $1/12-\epsilon$ implies that $\big(3(1/12-\epsilon)\big)^{n+2}$ is a lower bound on the reachability probability that is achieved by any good subsystem. These arguments are made precise in the following lemma.

Lemma 5.17. *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be subsystems of $\mathcal{M}_2$ with $3n+4$ states. If $\mathcal{N}_1$ is bad and $\mathcal{N}_2$ is good, then*

$$\Pr_{\mathcal{N}_1}(\Diamond \text{ target}) \leq \Pr_{\mathcal{N}_2}(\Diamond \text{ target}).$$

Proof. As observed above, each good subsystem achieves at least probability $\big(3(1/12-\epsilon)\big)^{n+2}$

of reaching "target" in the initial state. So it suffices to show that the probability achieved by $\mathcal{N}_1$ is less than this value.

Since $\mathcal{N}_1$ is bad, there exists a layer j of $\mathcal{N}_1$ such that both γ-cycles of this layer are interrupted (or the single one, if $j = n+1$). Hence, it suffices to show that if there exists a layer in which all γ-cycles are interrupted, then the probability to reach target is less than $\big(3(1/12-\epsilon)\big)^{n+2}$. So assume that all γ -cycles are interrupted in layer j of $\mathcal{N}_1$. The probability of reaching the next layer $j+1$ from any state in layer j is at most $(1+\gamma)(1-\gamma) \leq 2(1-\gamma)$, as one of the three states on the corresponding γ-cycle is missing and any transition to the next layer has probability lower than $1-\gamma$ (see Figure 5.10a). This implies, in particular, that $\Pr_{\mathcal{N}_1}(\Diamond\,\text{target})$ is bounded from above by $2(1-\gamma)$. Using the assumption that γ satisfies Equation (5.6) we get:

$$\Pr_{\mathcal{N}_1}(\Diamond\,\text{target}) \;\leq\; 2(1-\gamma) \;<\; (3(1/12-\epsilon))^{n+2} \qquad \square$$

Finally, observe that the directed path-partition width and tree-partition width of $\mathcal{M}_2$ is the same as of $\mathcal{M}_1$, as $\mathcal{M}_2$ includes more edges but still allows the directed path-partition which partitions states along the layers. Hence we have $\mathrm{dtpw}(\mathcal{M}_2) = \mathrm{dppw}(\mathcal{M}_1) = 6$. Together with Lemma 5.17, Lemma 5.16 and the fact that the probabilities of good subsystems in $\mathcal{M}_1$ and $\mathcal{M}_2$ coincide, this proves the following theorem.

Theorem 5.18. *The witness problem is NP-hard for Markov chains with dppw = 6 (and hence also for Markov chains with dtpw = 6).*

By combining this theorem with Proposition 5.9 it follows that the witness problem is also NP-hard for Markov chains with bounded undirected tree width, bounded undirected tree-partition width [See85], bounded directed tree width [JRST01] and bounded D-width [Saf05]. Furthermore, the unary weighted witness problem, which generalizes the witness problem, is also NP-hard for this class of Markov chains.

5.4 A dedicated algorithm for MDPs with low directed tree-partition width

The results of the previous section show that we cannot expect efficient algorithms which compute minimal witnessing subsystems even for Markov chains with low tree width. However, we can still hope for algorithms which use the information provided by a directed tree partition of the state space to solve the problem faster in practice. In this section we introduce such an algorithm.

It proceeds bottom-up along the tree order of the given directed tree partition and enumerates *partial subsystems*, which are rooted in the currently processed block. For each partial subsystem, the reachability probability achieved in the *interface* states of the block is computed. An interface state is a state which has some incoming edge from the predecessor block of the tree partition. A domination relation between partial subsystems, which compares the values achieved in the interface states, is used to prune away partial subsystems which do not need to be remembered, as they are covered by a better one.

Let $\mathcal{M} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be a fixed MDP in reachability form for the rest of this section, and $\mathcal{P} = \{B_1, \ldots, B_n\}$ be a directed tree partition of $\mathcal{M}$. We will assume that for all states $s \in S$ satisfying $\sum_{\alpha \in \text{Act}(s)} P(s, \alpha, \text{target}) > 0$ there exists a block $B \in \mathcal{P}$ such that

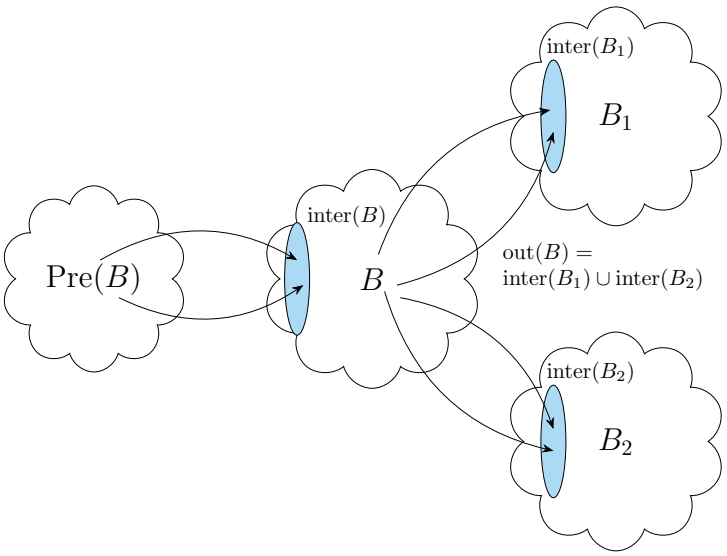


Figure 5.11: A sketch for the definitions used in this section. Additionally, Post(B) is the set of successor blocks of B, in this case $\{B_1, B_2\}$. Lemma 5.19 shows that to compute the optimal reachability probability for states in B one can first compute the optimal values for states in B_1 and B_2, then replace transitions of states in out(B) by transitions to "target" with the corresponding probability, and finally compute the values in B in this adapted MDP.

$B = \{s\}$. This condition can be ensured by a preprocessing step which adds a fresh state for each transition to target and puts it into its own block B.

We denote by $\mathrm{Post}(B) \subseteq \mathcal{P}$ the children of $B \in \mathcal{P}$ in the associated tree order, and by $\mathrm{Pre}(B) \in \mathcal{P}$ the unique parent of B. Furthermore, we let $\mathrm{inter}(B)$ be the *interface* states of B, which are states having some incoming edge from a state in $\mathrm{Pre}(B)$, or are initial. Using this notion we define $\mathrm{out}(B) = \bigcup_{B' \in \mathrm{Post}(B)} \mathrm{inter}(B')$, which represents the states outside of B which are reachable from some state in B in one step. See also Figure 5.11.

We will express the reachability probability achieved by states in B in some partial subsystem in terms of the probabilities achieved in the interface states of blocks in $\mathrm{Post}(B)$ for the same partial subsystem. To capture this formally, let f be a partial function from S to $[0,1]$ and $S' \subseteq S$ be a subset of states. We define the MDP $\mathcal{M}^f_{S'}$ using the following construction. In the subsystem $\mathcal{M}_{S'}$ of $\mathcal{M}$ induced by S' remove all outgoing edges from states $s \in \mathrm{dom}(f)$ (the domain of f) and replace them by an action with an edge to "target" with probability $f(s)$ and an edge to "exit" with probability $1-f(s)$. We write $\mathcal{M}^f$ as an abbreviation for $\mathcal{M}^f_S$.

We will use the following abbreviations which describe the value (which is either the minimal or maximal reachability probability) achieved by a states $q \in S'$ in the adapted MDP:

$$\text{min-val}^f_{S'}(q) = \mathbf{Pr}^{\min}_{\mathcal{M}^f_{S',q}}(\Diamond\,\text{target}) \qquad \text{and} \qquad \text{max-val}^f_{S'}(q) = \mathbf{Pr}^{\max}_{\mathcal{M}^f_{S',q}}(\Diamond\,\text{target}).$$

We write $\text{min-val}_{S'}$ or $\text{max-val}_{S'}$ for the respective values in the unchanged MDP $\mathcal{M}_{S'}$. The following lemma shows that to compute the values of states in $B \in \mathcal{P}$, one can first compute the values of states in $\mathrm{out}(B)$, then replace the edges of those states by an edge to "target" carrying this value, and finally compute the corresponding optimal value in the adapted MDP.

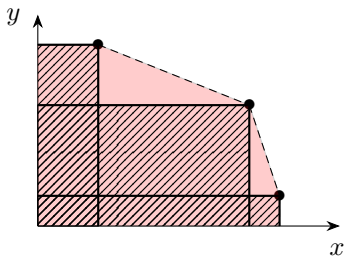


Figure 5.12: The black points represent three partial subsystems for $I = \{x, y\}$ using their value points. Partial subsystems with value points in the red area are dominated by the given three partial subsystems. If the value point of a partial subsystem lies in the dashed area, then it is *strongly* dominated by one of given partial subsystems. The weight of the partial subsystems is not considered here, but is important in general (see Definition 5.20).

Lemma 5.19. *Let $I, S' \subseteq S$ be two subsets of states and* $\text{val} \in \{\text{max-val}, \text{min-val}\}$. *Define the partial function f with domain I by: $f(q) = \text{val}_{S'}(q)$ for all $q \in I$. Then, for all $q \in S'$ we have*

$$\text{val}_{S'}(q) = \text{val}^f_{S'}(q).$$

Proof. The optimal solution of the linear program characterizing (minimal or maximal) is a vector containing the optimal reachability probabilities for each state. The linear program for MDP $\mathcal{M}^f_{S'}$ differs from the linear program for $\mathcal{M}_{S'}$ only by forcing the value for states $q \in I$ to be $f(q)$, which is defined to be the optimal value of q in the linear program for $\mathcal{M}_{S'}$. Hence, the optimal solutions of the two linear programs coincide. □

5.4.1 The domination relation

Let $\text{val} \in \{\text{max-val}, \text{min-val}\}$ be fixed for the remainder of this section, $B \in \mathcal{P}$ be a block of the directed tree partition $\mathcal{P}$ and $I = \text{inter}(B)$ be the interface states of B. We define $\text{reach}(I)$ to be the states *reachable* from I in the underlying graph of $\mathcal{M}$. A *partial subsystem* for B is a set $T \subseteq \text{reach}(I)$ and the corresponding *value point* $\text{vp}_T \in \mathbb{Q}^I$ is defined as $\text{vp}_T(q) = \text{val}_T(q)$ for all $q \in I \cap T$, and $\text{vp}_T(q) = 0$ for all $q \in I \setminus T$. The value point for T intuitively is the vector which assigns the values achieved in partial subsystem T to all states in I. We will treat partial functions with domain I as vectors in $\mathbb{Q}^I$ and use addition, multiplication by scalars and point-wise inequality checks as one would expect.

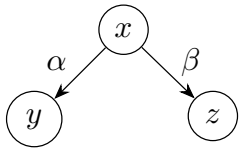
Now let us turn to the definition of a domination relation which compares different partial subsystems for B. On top of the fixed MDP $\mathcal{M}$ we will consider a weight function $wgt : S \to \mathbb{N}$. For a partial subsystem T for B we define $wgt(T) = \sum_{s \in T} wgt(s)$. First, we define the function π which collects all possible projections of a vector $\theta \in \mathbb{Q}^I$ onto a subset of the axes:

$$\pi(\theta) = \{\, \pi(\theta, D) \mid D \subseteq I \,\} \qquad \text{and} \qquad \pi(\theta, D)(x) = \begin{cases} \theta(x) & x \in D \\ 0 & \text{otherwise.} \end{cases}$$

Definition 5.20. Let $B \in \mathcal{P}$ be a block of the partition $\mathcal{P}$, $I = \text{inter}(B)$ and $\{T\} \cup \mathcal{S}$ be a set of partial subsystems for B. We say that $\mathcal{S}$ *dominates* T if there exists $\mathcal{S}' \subseteq \mathcal{S}$ such that

1. for all $T' \in \mathcal{S}'$ we have $wgt(T') \leq wgt(T)$, and
2. the value point vp_T of T is a convex combination of $\bigcup_{T' \in \mathcal{S}'} \pi(\text{vp}_{T'})$.

We say that $\mathcal{S}$ *strongly dominates* T if there exists $T' \in \mathcal{S}$ such that $\{T'\}$ dominates T.



f_1:	0	$1/2$
f_2:	$1/2$	0
f_3:	$1/4$	$1/4$

Figure 5.13: An example showing that the standard domination relation does not suffice for minimal reachability probabilities. Imagine that f_1, f_2 and f_3 represent value points for different partial subsystems for a block B with interface states $I = \{y, z\}$. Clearly, f_3 is a convex combination of f_1 and f_2, but a partial subsystem with probabilities f_3 would achieve a larger minimum value than any partial subsystem with probabilities f_1 or f_2 in states y, z.

An intuition is given in Figure 5.12. If a partial subsystem is dominated by a set of partial subsystems, then it is not relevant for the computation of minimal witnesses as it can always be replaced by one of the dominating partial subsystems without a decrease in probability. This is formalized in the following lemma. It turns out that for minimal reachability probabilities we have to use the strong domination relation. An example which highlights this difference is given in Figure 5.13.

Proposition 5.21. *Let $B \in \mathcal{P}$, $I = \mathrm{inter}(B)$ and $\mathcal{S}$ be a set of partial subsystems for B. Furthermore, let $\mathcal{M}' = \mathcal{M}_{S'}$ be a subsystem of $\mathcal{M}$ induced by the states $S' \subseteq S$ and define $S_1 = S' \setminus \mathrm{reach}(I)$ and $S_2 = S' \cap \mathrm{reach}(I)$.*

1. *If $\mathbf{Pr}^{\max}_{\mathcal{M}'}(\Diamond\,\mathrm{target}) \geq \lambda$ holds and $\mathcal{S}$ dominates S_2, then there exists $T \in \mathcal{S}$ such that $\mathcal{N} = \mathcal{M}_{S_1 \cup T}$ satisfies $\mathbf{Pr}^{\max}_{\mathcal{N}}(\Diamond\,\mathrm{target}) \geq \lambda$ and $wgt(T) \leq wgt(S_2)$.*
2. *If $\mathbf{Pr}^{\min}_{\mathcal{M}'}(\Diamond\,\mathrm{target}) \geq \lambda$ holds and $\mathcal{S}$ strongly dominates S_2, then there exists $T \in \mathcal{S}$ such that $\mathcal{N} = \mathcal{M}_{S_1 \cup T}$ satisfies $\mathbf{Pr}^{\min}_{\mathcal{N}}(\Diamond\,\mathrm{target}) \geq \lambda$ and $wgt(T) \leq wgt(S_2)$.*

Proof. 1. As $\mathcal{S}$ dominates S_2, there exists a subset $\{T_1, \ldots, T_k\} \subseteq \mathcal{S}$ such that $wgt(T_i) \leq wgt(S_2)$ for all $1 \leq i \leq k$ and $\{T_1, \ldots, T_k\}$ dominates S_2. Let $f_i = \mathrm{vp}_{T_i} \in \mathbb{Q}^I$ be the value point of partial subsystem T_i for all $1 \leq i \leq k$, and $g = \mathrm{vp}_{S_2} \in \mathbb{Q}^I$. By definition, g is a convex combination of vectors $\bigcup_{1 \leq i \leq k} \pi(f_i)$. That is, there exist $\xi_1, \ldots, \xi_m \in \mathbb{Q}_{\geq 0}$ such that

$$g = \sum_{1 \leq j \leq m} \xi_j \cdot \gamma_j, \quad \text{with} \quad \sum_{1 \leq j \leq m} \xi_j \leq 1 \quad \text{and} \quad \gamma_j \in \bigcup_{1 \leq i \leq k} \pi(f_i) \quad \text{for all } 1 \leq j \leq m.$$

Let $\gamma'_j = f_i$ if $\gamma_j \in \pi(f_i)$ (for $1 \leq j \leq m$ and $1 \leq i \leq k$), and if multiple such f_i exist then take arbitrary one. As all vectors in $\pi(f)$ are point-wise smaller or equal than f, we get

$$g \;=\; \sum_{1 \leq j \leq m} \xi_j \cdot \gamma_j \;\leq\; \sum_{1 \leq j \leq m} \xi_j \cdot \gamma'_j.$$

Let $\mathcal{N}_i = \mathcal{M}_{S_1 \cup T_i}$ be the subsystem one gets by taking S' and replacing partial subsystem S_2 by partial subsystem T_i, for all $1 \leq i \leq k$. By lemma 5.19 we have for all $1 \leq i \leq k$

$$\mathbf{Pr}^{\max}_{\mathcal{N}_i}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}^{f_i}_{S'}}(\Diamond\,\mathrm{target}) \qquad \text{and} \qquad \mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\max}_{\mathcal{M}^{g}_{S'}}(\Diamond\,\mathrm{target}).$$

We claim that one of these $\mathcal{N}_i$ satisfies $\mathbf{Pr}^{\max}_{\mathcal{N}_i}(\Diamond\,\mathrm{target}) \geq \mathbf{Pr}^{\max}_{\mathcal{M}'}(\Diamond\,\mathrm{target})$. To show this, let $\mathfrak{S}$ be a maximizing scheduler for $\mathcal{M}^{g}_{S'}$. The probability to reach "target" from s_{in} under $\mathfrak{S}$ can

be split into two parts: either "target" is reached while passing through states $\mathrm{dom}(g) = I$, or "target" is reached without seeing I. By construction of $\mathcal{M}^g_{S'}$, in which any state in I can only be seen once, we have

$$\Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond \text{target}) \;=\; \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,) \;+\; \sum_{q \in I} \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond q) \cdot g(q).$$

Here $\bar{I} = S \setminus I$. Let $\theta \in \mathbb{Q}^I$ be defined by $\theta(q) = \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond q)$ for all $q \in I$. Then the second part of the above sum can also be written as $\theta \cdot g$. Now choose $1 \le l \le k$ such that $\theta \cdot \gamma'_l$ is maximal and let $f^* = \gamma'_l$. It follows that

$$\theta \cdot f^* \;\ge\; \theta \cdot \sum_{1 \le j \le k} \xi_j \cdot \gamma'_j \;\ge\; \theta \cdot g,$$

where we use $\sum_{1 \le j \le k} \xi_j \le 1$. Finally, we have

$$\mathbf{Pr}^{\max}_{\mathcal{N}_l}(\Diamond \text{target}) \;\ge\; \Pr^{\mathfrak{S}}_{\mathcal{M}^{f^*}_{S'}}(\Diamond \text{target}) \;=\; \Pr^{\mathfrak{S}}_{\mathcal{M}^{f^*}_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,) + \theta \cdot f^* \;\ge\; \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond \text{target}).$$

The last inequality follows by $\theta \cdot f^* \ge \theta \cdot g$ and

$$\Pr^{\mathfrak{S}}_{\mathcal{M}^{f^*}_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,) \;=\; \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,),$$

which holds because $\mathcal{M}^g_{S'}$ and $\mathcal{M}^{f^*}_{S'}$ differ only in states $q \in I$. As $\mathfrak{S}$ is a maximizing scheduler for $\mathcal{M}^g_{S'}$, it follows that $\mathbf{Pr}^{\max}_{\mathcal{N}_l}(\Diamond \text{target}) \ge \mathbf{Pr}^{\max}_{\mathcal{M}'}(\Diamond \text{target})$.

2. As $\mathcal{S}$ strongly dominates S_2 there exists a partial subsystem $T \in \mathcal{S}$ such that $\{T\}$ dominates S_2. Hence we have $wgt(T) \le wgt(S_2)$ and $\mathrm{vp}_{S_2} \le \mathrm{vp}_T$. Let $f = \mathrm{vp}_T$, $g = \mathrm{vp}_{S_2}$ and $\mathcal{N} = \mathcal{M}_{S_1 \cup T}$. By Lemma 5.19 we have

$$\mathbf{Pr}^{\min}_{\mathcal{N}}(\Diamond \text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}^f_{S'}}(\Diamond \text{target}) \qquad \text{and} \qquad \mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}^g_{S'}}(\Diamond \text{target}).$$

We claim that $\mathbf{Pr}^{\min}_{\mathcal{N}}(\Diamond \text{target}) \ge \mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target})$. Let $\mathfrak{S}$ be a minimizing scheduler for $\mathcal{M}^f_{S'}$. As in case (1), we can split the probability of reaching "target" under $\mathfrak{S}$ by distinguishing whether I is reached or not. We calculate

$$\begin{aligned}\mathbf{Pr}^{\min}_{\mathcal{N}}(\Diamond \text{target}) \;&=\; \Pr^{\mathfrak{S}}_{\mathcal{M}^f_{S'}}(\Diamond \text{target}) \;=\; \Pr^{\mathfrak{S}}_{\mathcal{M}^f_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,) \;+\; \sum_{q \in I} \Pr^{\mathfrak{S}}_{\mathcal{M}^f_{S'}}(\Diamond q) \cdot f(q) \\ \ge\; \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond \text{target} \wedge \Box \bar{I}\,) \;&+\; \sum_{q \in I} \Pr^{\mathfrak{S}}_{\mathcal{M}^g_{S'}}(\Diamond q) \cdot g(q) \;\ge\; \mathbf{Pr}^{\min}_{\mathcal{M}^g_{S'}}(\Diamond \text{target}) \;=\; \mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target}).\end{aligned}$$

Here we used that $\mathcal{M}^g_{S'}$ and $\mathcal{M}^f_{S'}$ differ only in the transitions of states in I, which can only be reached once and move to "target" directly. Hence $\mathfrak{S}$ can be used as a scheduler for $\mathcal{M}^g_{S'}$ and the probabilities of reaching states $q \in I$ coincide in both MDPs under $\mathfrak{S}$, as do the probabilities of reaching "target" without seeing I. This concludes the proof. □

Computing the domination relation. To compute the domination relation, we propose Algorithm 3 which uses an incremental convex-hull algorithm as a subroutine. It takes as input a set

of partial subsystems $\mathcal{S}$ and returns a non-dominated subset of $\mathcal{S}$. First, the partial subsystems are grouped by their weight (line 3). The ConvexHull object (see line 5) allows to add points incrementally, and stores the vertices of the convex hull of points added so far in the field *vertices*. The main loop of the procedures goes through all possible weight values k, computes value points and projections of partial subsystems of weight k and adds them to the ConvexHull object $\mathcal{H}$. Then, the convex hull of the resulting set of points is computed (observe that this includes value points of partial subsystems with weight less than k from previous iterations of the loop). Only the partial subsystems of weight k whose value points are vertices of the corresponding polytope are kept. All others are convex combinations of these vertices, and hence dominated.

The convex hull of k points in dimension d can be computed in time $O(k \cdot \log k + k^{\lfloor d/2 \rfloor})$ (see [Cha93]). In our case d corresponds to the number $|I|$ of interface states, as this is the dimension of the value points. A number of dedicated and fast incremental algorithms exist to compute the convex hull in low dimensions[Gra72, BDH96, Cha96]. Therefore, tree partitions with few interface states in each block are desirable.

We now show that Algorithm 3 indeed correctly computes the domination relation.

Lemma 5.22. *Let $B \in \mathcal{P}$, $\mathcal{S}$ be a set of partial subsystems for B and $\mathcal{R}$ be the result of Algorithm 3 on input $\mathcal{S}$. Then,*

- *for any $T \in \mathcal{S} \setminus \mathcal{R}$ it holds that $\mathcal{R}$ dominates T, and*
- *no $T \in \mathcal{R}$ is dominated by $\mathcal{R} \setminus \{T\}$.*

Proof. Let m be the maximal weight of any partial subsystem in $\mathcal{S}$. For each $1 \leq k \leq m$ the set $\mathcal{H}$.vertices in line 9 contains the vertices of the convex hull of points

$$\Pi_k \;=\; \bigcup \{\, \pi(\mathrm{vp}_{S'}) \;\mid\; S' \in \mathcal{S} \;\text{ and }\; wgt(S') \leq k \,\}.$$

If for some $T \in \mathcal{S}[k]$, vp_T is not in $\mathcal{H}$.vertices at that point it is a convex combination of Π_k. Hence, T is dominated by $\bigcup_{1 \leq j \leq k} \{\, S' \in \mathcal{S}[j] \mid \mathrm{vp}_{S'} \in \mathcal{H}.\text{vertices} \,\}$, and thereby by $\mathcal{R}$.

For the second claim, suppose that some partial subsystem $T \in \mathcal{R}$ is dominated by $\mathcal{R} \setminus \{T\}$. Then, in particular T is dominated by $\{\, S' \in \mathcal{R} \setminus \{T\} \mid wgt(S') \leq wgt(T) \,\}$, and hence also by $\bigcup_{1 \leq j \leq wgt(T)} \mathcal{S}[j]$, as the former is a subset of the latter. It follows that vp_T is not a vertex of the convex hull of $\bigcup_{1 \leq j \leq wgt(T)} \{\pi(\mathrm{vp}_{S'}) \mid S' \in \mathcal{S}[j]\}$, as it is a convex combination of vectors therein. But then T cannot be in $\mathcal{R}$, as it is not added to $\mathcal{R}$ in Line 9 in the loop iteration corresponding to $k = wgt(T)$. □

5.4.2 An algorithm based on the domination relation

We are now in the position to describe Algorithm 4, which computes minimal witnessing subsystem of $\mathcal{M}$ for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}}(\Diamond \,\text{target}) \geq \lambda$ (for $\mathfrak{m} \in \{\max, \min\}$), using the structure of the tree decomposition $\mathcal{P}$ and the domination relation. It will be described for $\mathfrak{m} = \max$, but can be used in the same way for $\mathfrak{m} = \min$ by using the strong domination relation, rather than the standard one. It proceeds bottom-up along the tree partition, enumerates all partial subsystems for the current block B and prunes away those that are dominated. The enumeration is done by enumerating the subsets of states in B and combining them in all possible ways with partial subsystems which have already been computed for the successor blocks.

Algorithm 3: removeDominated

Input: Set of partial subsystems $\mathcal{S}$ for B, with $I = \text{inter}(B)$.

1 $m := \max\{\ wgt(S')\ \mid\ S' \in \mathcal{S}\ \}$
`/* Group partial subsystems by their weight. */`
2 **for** $k = 1$ **to** m **do**
3 $\quad \mathcal{S}[k] := \{\ S' \in \mathcal{S}\ \mid\ wgt(S') = k\ \}$
4 **end**
`/* Initialize an empty ConvexHull object */`
5 $\mathcal{H} :=$ ConvexHull( )
6 **for** $k = 1$ **to** m **do**
`/* Compute projections of value vectors in` $\mathcal{S}[k]$`. */`
7 $\quad \Pi := \bigcup\{\ \pi(\text{vp}_{S'})\ \mid\ S' \in \mathcal{S}[k]\ \}$
`/* Add` Π `to the incremental ConvexHull object. */`
8 $\quad \mathcal{H}$.addPoints(Π)
`/* Remember only subsystems in` $\mathcal{S}[k]$ `that are vertices of` $\mathcal{H}$`. */`
9 $\quad \mathcal{R} := \mathcal{R} \cup \{\ S' \in \mathcal{S}[k]\ \mid\ \text{vp}_{S'} \in \mathcal{H}.\text{vertices}\ \}$
10 **end**
11 **return** $\mathcal{R}$

To avoid enumerating all subsets of B, we apply a filter based on a Boolean condition. It encodes that there should be no "unnecessary" states, which are states having neither a successor or predecessor in the subset nor an incoming or outgoing edge to other blocks. This is realized by the following Boolean formula with variables in S:

$$\phi(B) = \bigwedge_{s \notin \text{inter}(B)} \left(s \to \bigvee_{s' \in \text{pre}(s) \cap B} s' \right) \wedge \bigwedge_{s \notin \text{ex}(B)} \left(s \to \bigvee_{s' \in \text{post}(s) \cap B} s' \right)$$

Here $\text{ex}(B) = \{s \in B \mid \text{post}(s) \setminus B \neq \emptyset\}$, and $\text{post}(s)$ and $\text{pre}(s)$ denote the successors and predecessors of s in the underlying graph of $\mathcal{M}$. Any partial subsystem S' for B such that $S' \cap B$ is not a model of $\phi(B)$ is dominated by another partial subsystem for B. The latter can be obtained by removing unnecessary states from S'.

Let us explain more precisely how Algorithm 4 works. The algorithm keeps a map `psubsys` from blocks $B \in \mathcal{P}$ to partial subsystems for B. This map is populated in a bottom-up traversal along the tree order of $\mathcal{P}$ (Line 1). For a given block B, the models of $\phi(B)$ (these are subsets of B) are enumerated (Line 4). The method *successorPoints* in Line 5 returns all pairs (S', f), where S' is a set which can be obtained by combining partial subsystems in $\texttt{psubsys}[B_i]$, for all $B_i \in \text{Post}(B)$, and $f \in \mathbb{Q}^{\text{out}(B)}$ is a vector including corresponding values of all states in $\text{out}(B)$. It is defined formally as follows, given that the successors of B in the tree order are $\text{Post}(B) = \{B_1, \ldots B_k\}$:

$$\text{successorPoints}(\texttt{psubsys}, B) =$$
$$\Big\{\ \Big(\ \bigcup_{1 \le j \le k} S_j,\ \text{con}\big(\text{vp}_{S_1}, \ldots, \text{vp}_{S_k}\big)\ \Big)\ \mid\ S_1 \in \texttt{psubsys}[B_1], \ldots, S_k \in \texttt{psubsys}[B_k]\ \Big\},$$

where $\text{con}(\text{vp}_{S_1}, \ldots, \text{vp}_{S_k})$ is the vector one gets by concatenating vectors vp_{S_i}, with $1 \le i \le k$. Here we use that the interfaces of blocks $B_1, \ldots, B_k$ are disjoint. These interfaces form the domains of vectors $\text{vp}_{S_1}, \ldots \text{vp}_{S_k}$. The vectors vp_{S_i} have been computed in a previous iteration

Algorithm 4: A dedicated algorithm for MDPs using a given directed tree partition.

Input: MDP $\mathcal{M}$, directed tree partition $\mathcal{P}$, rational λ
Output: Minimal witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$.

```
   /* Bottom-up traversal of the tree partition.                          */
1  for B in reverse(topologicalSort(P)) do
2      I := inter(B)
3      O := out(B)
       /* Consider only subsets of B that satisfy φ(B)                    */
4      for S_B ⊆ B such that S_B ⊨ φ(B) do
           /* Consider each combination of partial subsystems of the children of B.
              */
5          for (S', f) in successorPoints(psubsys, B) do
               /* The new partial subsystem S_new for I combines S_B and S'. */
6              S_new := S_B ∪ S'
7              vp_{S_new} := (max-val^f_{S_new})|_I
               /* Remember the corresponding partial subsystem.           */
8              psubsys[B].insert(S_new)
9          end
           /* Remove dominated points                                     */
10         psubsys[B] := removeDominated(psubsys[B])
11     end
12 end
   /* Here B_n is assumed to be the root of the tree associated with P.   */
13 return argmin { wgt(S') for S' in psubsys[B_n] such that vp_{S'}(s_in) ≥ λ }
```

of the for loop in Line 7 and are assumed to be in global memory (they are also needed to compute the domination relation).

For each such pair (S', f) one computes the value achieved in interface states of B under the partial subsystem $S_B \cup S'$ (Line 7). This corresponds to computing the maximal reachability probabilities in the MDP $\mathcal{M}^f_{S_{\text{new}}}$.

Proposition 5.23. *If Algorithm 4 returns S' on input $(\mathcal{M}, \mathcal{P}, \lambda)$, then S' is a weight-minimal witness for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$. It returns within exponential time in the size of the input.*

Proof. First, we argue that if Algorithm 4 returns S' then $\mathcal{M}_{S'}$ is a witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$. In this case we have $\text{vp}_{S'}(s_{in}) \geq \lambda$, and hence, the maximal probability achieved by $\mathcal{M}_{S'}$ in state s_{in} is indeed larger than λ. Observe that values of states in partial subsystems are computed correctly in Line 7 by Lemma 5.19.

Next, we take any witnessing subsystem $\mathcal{M}_T$ and show that $wgt(T) \geq wgt(S')$ holds. Let $B_1, \ldots, B_n$ be a reverse-topological order of the tree partition. We will construct a sequence $S_0, \ldots, S_n$ of subsets of S inductively such that

- for all $0 \leq i \leq n$ we have $wgt(T) \geq wgt(S_i)$ and S_i induces a witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\lozenge \text{target}) \geq \lambda$, and
- for all $1 \leq i \leq n$ and $j \leq i$ the partial subsystem $S_i \cap \text{reach}(\text{inter}(B_j))$ is in $\texttt{psubsys}[B_j]$ at the end of the execution of Algorithm 4.

We start by setting $S_0 = T$. To find S_{i+1} we assume that the above properties hold for all S_j with $j \leq i$. If the partial subsystem $S_i \cap \text{reach}(\text{inter}(B_{i+1}))$ is included in $\texttt{psubsys}[B_{i+1}]$,

we can set $S_{i+1} = S_i$. Otherwise, we proceed as follows. Let $\{B_{l_1}, \ldots, B_{l_m}\} = \mathrm{Post}(B_{i+1})$. By induction hypothesis, the partial subsystem $K_{l_j} = S_i \cap \mathrm{reach}(\mathrm{inter}(B_{l_j}))$ is in $\texttt{psubsys}[B_{l_j}]$ at the end of Algorithm 4 for all $B_{l_j} \in \mathrm{Post}(B_{i+1})$. Hence, the partial subsystem $K = \bigcup\{K_{l_1}, \ldots, K_{l_m}\}$ appears in $\mathit{successorPoints}(\texttt{psubsys}, B_{i+1})$ when considering block B_{i+1} in Line 5 of Algorithm 4. We make a case-distinction on whether $S_i \cap B_{i+1}$ is a model of the formula $\phi(B_{i+1})$.

Case 1: $S_i \cap B_{i+1} \models \phi(B_{i+1})$. In this case, the partial subsystem $K \cup (S_i \cap B_{i+1})$ is inserted into $\texttt{psubsys}[B_{i+1}]$ in Line 8. As it is not in $\texttt{psubsys}[B_{i+1}]$ at the end of the execution of Algorithm 4 by assumption, it must have been removed in Line 10. Hence, by Lemma 5.22, $K \cup (S_i \cap B_{i+1})$ is dominated by $\texttt{psubsys}[B_{i+1}]$. By Proposition 5.21 we can conclude that that there exists a partial subsystem $K' \in \texttt{psubsys}[B_{i+1}]$ such that $(S_i \setminus \mathrm{reach}(\mathrm{inter}(B_{i+1}))) \cup K'$ induces a witnessing subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \,\mathrm{target}) \geq \lambda$, and $wgt(K') \leq wgt(K \cup (S_i \cap B_{i+1}))$. We set $S_{i+1} = (S_i \setminus \mathrm{reach}(\mathrm{inter}(B_{i+1}))) \cup K'$. As $K' \in \texttt{psubsys}[B_{i+1}]$ holds, it follows that for all B' which are below B_{i+1} in the tree order we have $K' \cap \mathrm{reach}(\mathrm{inter}(B')) \in \texttt{psubsys}[B']$.

Case 2: $S_i \cap B_{i+1} \not\models \phi(B_{i+1})$. In this case there exists some $L \subseteq S_i \cap B_{i+1}$ such that $L \models \phi(B_{i+1})$. For any set $K \subseteq \mathrm{reach}(\mathrm{out}(B_{i+1}))$, the partial subsystem $L \cup K$ strongly dominates the partial subsystem $(S_i \cap B_{i+1}) \cup K$. Now the argument of **Case 1** can be applied by observing that if a set of partial subsystems dominate $L \cup K$, then the same set dominates $(S_i \cap B_{i+1}) \cup K$.

This shows that we can construct the sequence $S_0, \ldots, S_n$ satisfying the above properties. But then S_n induces a witnessing subsystem and satisfies $wgt(T) \geq wgt(S_n)$. As S_n is part of $\texttt{psubsys}[B_n]$ in the last line of the algorithm, we have $wgt(S_n) \geq wgt(S')$.

Finally, we argue that the algorithm takes at most exponential time to return. Let S be the states of $\mathcal{M}$, $N = |S|$ and W be the width of the given tree partition. The outermost for-loop is taken at most N times. The for-loop starting in Line 4 is taken at most 2^W times, as it ranges over subsets of B which has at most W states. The innermost for-loop in Line 5 is taken at most 2^N times, as it ranges over subsets of S. The value computation in Line 7 can be done in polynomial time in $\mathcal{M}$. The subroutine removeDominated (Algorithm 3) which is called in Line 10 requires at most exponential time in W (as the dimension of points is the number of interface states) and polynomial time in the number of input vectors (in this case $|\texttt{psubsys}[B]|$). This is because the convex hull of k points in dimension d can be computed in time $O(k \log k + k^{\lfloor d/2 \rfloor})$ [Cha93]. In our case $d = O(W)$ and $k = O(2^W \cdot |\texttt{psubsys}[B]|) = O(2^W \cdot 2^N)$. The factor of 2^W in the number of points used in the convex hull computation comes from the fact that we include all projections of any point for a partial subsystem in $\texttt{psubsys}[B]$. All in all, the algorithm requires at most exponential time in the size of $\mathcal{M}$. □

Witnesses for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \,\mathrm{target}) \geq \lambda$ can be handled by replacing the call to removeDominated in Line 10 by a method which computes the *strong domination* relation. This essentially requires computing the Pareto frontier of a set of vectors, which can be done in time $O(k(\log k)^{d-2})$ for k d-dimensional vectors [KLP75]. In general, computing the value vectors in line 7 amounts to solving a linear program, as it requires computing optimal reachability probabilities of an MDP. However, if the input is Markov chain, it suffices to solve a linear equation system. Table 5.1 gives an overview over the possible instances of the algorithm.

Additional heuristics to exclude partial subsystems. In addition to the domination relation we propose two conditions on when a partial subsystem can be excluded. Suppose that we are considering partial subsystem T for block B. If we know, by the structure of the given system, that "target" is only reachable from s_{in} through B and $\sum_{q \in \mathrm{inter}(B)} \mathrm{vp}_T(q) < \lambda$ holds, then we

Table 5.1: Different versions of Algorithm 4.

Model	value function	computing the value (Line 7)	domination relation (Line 10)
Markov chain	$\Pr(\Diamond \text{ target})$	linear equations	standard
MDP	$\mathbf{Pr}^{\max}(\Diamond \text{ target})$	linear program	standard
	$\mathbf{Pr}^{\min}(\Diamond \text{ target})$		strong

know that T cannot be part of a witnessing subsystem for $\mathbf{Pr}^{\mathrm{m}}(\Diamond \text{ target}) \geq \lambda$. This is because no matter how the partial subsystem will be completed, the value in the initial state of reaching "target" will be below λ.

For the second condition, let us assume that U is an upper bound on the weight of a minimal witnessing subsystem (which may have been computed heuristically) and M is the weight of a shortest path from the initial state to any state of I. If $M + wgt(T) > U$, then T cannot be part of any minimal witness.

5.4.3 Experimental evaluation

We have implemented Algorithm 4 in Storm using the convex hull library qhull[1]. At the moment, the implementation is limited to Markov chains and computing state-minimal witnessing subsystems, but it could be extended in a straight-forward manner to also handle weight-minimality and MDPs. All experimental data and scripts to produce them are available [Jan22b].

To evaluate it, we reconsider the *bounded retransmission protocol* (brp), which was also used for benchmarks in Section 4.2.5. It is parametrized by N (the number of "chunks" of the transmitted file) and K (the number of maximal retransmissions per chunk). We fix $K = 1$ but consider increasing values for N, yielding instances with in between 185 ($N = 10$) and 1625 ($N = 90$) states. We consider the probabilistic reachability constraint $\Pr(\Diamond \text{ target}) \geq \lambda$, for varying thresholds λ. The state "target" represents the situation that the receiver reports an uncertainty on the success of the transmission. The protocol maintains a counter which is only increased up to maximal value N, and using this fact one can compute a natural directed path partition for the model. Essentially, it partitions the state space along the possible values of the counter. The directed path partitions that we get in this way have length N+1, constant width 37 and two interface states in each block.

The experiments were run on a computer with two Intel E5-2680 processors having 8 cores each at 2.70 GHz running Linux, with a total of 378 GBs of RAM. Each instance was limited to a single core for a fairer comparison. While Gurobi is able to use multiple cores, our implementation of Algorithm 4 is not. In principle, however, Algorithm 4 offers large potential for parallelization, as it has to process a large number of independent partial subsystems. In particular, the for loop in Line 4 could be parallelized. All instances were run with a timeout of 1200 seconds and a memory limit of 30 GB.

Figure 5.14 compares the computation times of Algorithm 4 against the MILP-based approach described in Section 4.2.1. The computation times do not include the generation of the path partition. As the model at hand is a Markov chain, we can use both the min-witness program and the max-witness program defined in Section 4.2.1 (see Definitions 4.28 and 4.32). To

[1] http://www.qhull.org/

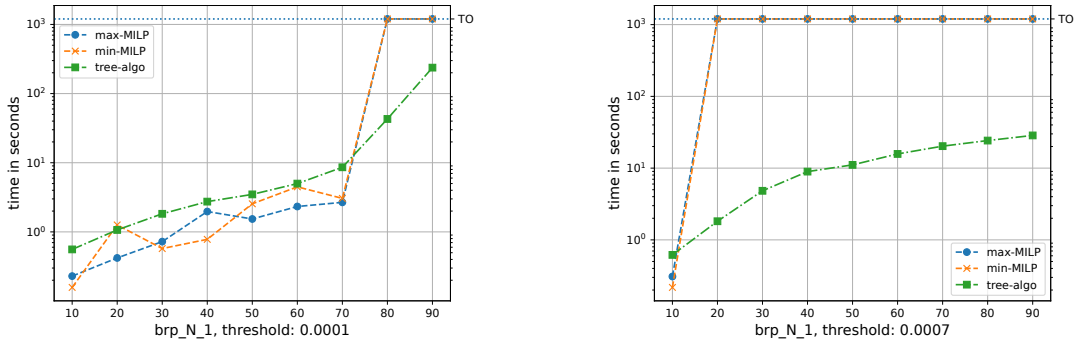


Figure 5.14: Computation times of the MILP-based approaches using the min-witness and max-witness MILPs with Switss (see Section 4.2.1) and Algorithm 4 for two different thresholds.

solve the MILPs, Switss uses the solver Gurobi [Gur22] (version 9.5). The evaluation shows that Algorithm 4 performs better for this benchmark, in particular for larger instances and thresholds. While a result is not returned before the timeout using the MILP-based approaches for the threshold 0.0007 and instances with $N \geq 20$, our implementation returns in less than 100 seconds for instances up to $N = 90$.

We also considered instances of the bounded retransmission model with maximal number of retransmissions $K = 2$, rather than $K = 1$. However, for these instances neither Algorithm 4 nor the MILP-based approaches returned an answer within the timeout, for any $N \geq 10$.

Summing up, the experiments show that Algorithm 4 may outperform MILP-based approaches for well-structured benchmarks in which favorable directed path decompositions exist and can be computed easily. Nevertheless, it does not scale to larger instances or path decompositions whose width is not very small.

Chapter 6

Explications for probabilistic timed automata

The models we have considered so far in this thesis do not contain any information about *timing aspects* of the modeled systems. For example, they do not specify for how long a system remains in any given state. To encode timing into a Markov decision process, one could make the assumption that each state is visited for a fixed amount of time, and therefore a path of length n corresponds to a time interval of n time steps. Such a model of time is inherently *discrete*, as the granularity of time steps needs to be fixed when modeling the system.

To faithfully represent *real-time systems* and the timing constraints which usually form a crucial part of their specification, this is not always appropriate. Therefore, a theory of *timed automata* was developed [AD94]. It is based on a dense-time model and extends ordinary automata by real-valued clocks. Based on this theory, *probabilistic timed automata* (PTA) were introduced [KNSS02], which describe systems that combine real-time and probabilistic aspects.

The underlying state space of such models is inherently uncountable, as the clocks are real-valued. However, it was discovered that timed automata have *finite-state bisimulation quotients* [AD94] (the classical construction is called the *region construction*), which makes many verification problems decidable. Much work has been put into making model checking technology for timed systems feasible and scalable, and very successful tools such as Uppaal [LPY97] exist. In the probabilistic world, model checking algorithms for probabilistic timed automata and related complexity questions were considered in [KNSS02, KNSW07, LS07]. Several notions of abstraction and simulation for PTA have been considered [CHK08, ST10]. The tool Prism [KNP11] is the most prominent tool for modeling and verifying PTA.

Abstraction-based methods are at the core of model checking algorithms for timed automata and a number of approaches for *(counterexample-guided) abstraction refinement* have been proposed [DKL07, HZH+10, RSM19]. All of these works address model checking of safety properties. The notions of counterexample which are usually considered are variants of *timed traces*. These are alternating sequences of states and transitions of the timed automaton, which witness the fact that a violating execution of the timed automaton exists.

In this chapter, we propose a notion of witnessing subsystems for lower-bounded prob-

abilistic reachability constraints in probabilistic timed automata. As for MDPs, the possible behavior of a subsystem is restricted when compared to the original PTA. It is witnessing if it, nevertheless, satisfies the threshold constraint on the (maximal or minimal) reachability probability. We introduce three notions of *size* for PTA subsystems, some of which take into account timing aspects. This is done by considering the logical strength of location invariants (which determine the clock valuations that are valid in a given location), or the volume of the set of clock valuations satisfying the invariants. Finally, we show that there is a correspondence between witnessing PTA subsystems and Farkas certificates for certain finite-state quotients of the PTA. Using this correspondence, we describe single-exponential algorithms for computing minimal witnessing PTA subsystems for all three notions of size.

Related work

As mentioned above, *timed traces* are utilized as counterexamples to safety properties by model checkers such as Uppaal [LPY97], and also applied in counter-example guided abstraction refinement for timed automata [DKL07, HZH+10, RSM19]. Repair mechanisms of timed automata based on analyzing timed traces are presented in [KLW19], and the extraction of *dynamic causes* from timed traces was studied in [KLS20]. Certification of *positive* model checking results has been studied for timed automata in [WvM20, WHvP20]. This work considers certificate conditions of non-reachability in standard timed automata and emptiness of timed Büchi automata. A formally verified (in the proof-assistant Isabelle/HOL) certificate checker is presented, building on previous formalizations of timed automata [Wim16, WL18]. This formalization includes probabilistic timed automata [WH18].

Counterexamples for safety properties in *hybrid automata*, which generalize timed automata, are described in [NÁCC14]. Here, counterexamples are also a form of traces, and the emphasis of the paper is to extend existing algorithms and tools (which are generally incomplete) such that they also return a counterexample when a negative answer is given. An extension of our notion of subsystem to *probabilistic rectangular automata* has been considered in [Hen21].

Outline

Section 6.1 first defines witnessing subsystems for PTA, and shows that the maximal (and minimal) reachability probabilities cannot increase when passing to a (strong) subsystem. This is done by showing that a witnessing subsystem induces a Farkas certificate in a finite-state quotient of the PTA (Theorem 6.3). Then, after introducing a *zone closure* operation for difference bounds matrices (Section 6.1.2), we discuss how to go from a Farkas certificate of a finite-state quotient to a witnessing subsystem (Section 6.1.3). Finally, three notions of minimality of witnessing subsystems, along with algorithms to compute them, are introduced in Section 6.2.

Relation to published work

The chapter is largely based on the paper [JFB20], which is joint work with Florian Funke and Christel Baier. In contrast to [JFB20], we no longer assume that all time-divergent schedulers reach $\{\text{target}, \text{exit}\}$ with probability one, and the algorithms in Section 6.1.3 are presented using label-based minimization of witnessing subsystems in the quotient MDP. Furthermore, in our definition of PTA we assume that transition labels uniquely identify the transition, which simplifies the definition of subsystems.

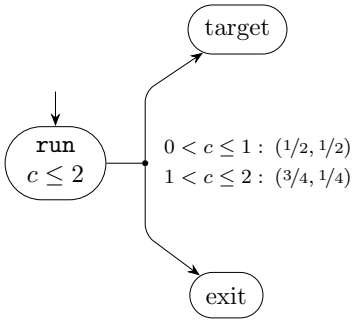


Figure 6.1: A PTA $\mathcal{T}_1$ over a single clock c, using a compact representation of transitions. The location `run` has two outgoing transitions, one with guard $0 < c \leq 1$ carrying probability $1/2$ to target and $1/2$ to exit, and another one with guard $1 < c \leq 2$, carrying probability $3/4$ to target and $1/4$ to exit. The location invariant of `run` is $c \leq 2$.

6.1 Witnessing subsystems for probabilistic timed automata

In this section we define a notion of subsystem for probabilistic timed automata (PTA), which generalizes the notion of subsystem for MDPs, as given in Definition 4.1. We consider *pointed* PTA as defined in Section 2.3, which contain distinguished absorbing locations "target" and "exit".

Before defining subsystems, let us consider two examples of pointed PTA. The PTA in Figure 6.1 represents the following simple scenario. A server runs a computation, and the probability of successfully computing a result (represented by reaching "target") depends on how long it runs. If it stops within one time unit, this probability is $1/2$, and if it runs for more than one time unit, this probability increases to $3/4$. The PTA in Figure 6.2 adds one layer of complexity to this scenario and includes an additional clock u. An update is being installed on the server, and the probability of success now depends both on the time spent on the computation, and whether the update was completed (this happens when $u = 1$) before the server stops. With probability $2/3$, the server has time to complete the update before the computation starts.

6.1.1 Subsystems for probabilistic timed automata

We start by defining subsystems for PTA.

Definition 6.1 (Subsystem). Let $\mathcal{T}$ be a pointed PTA with $\mathcal{T} = (\mathrm{Loc}, \mathsf{Cl}, \mathrm{Act}, \mathrm{inv}, T, l_{in})$. A pointed PTA $\mathcal{T}' = (\mathrm{Loc}', \mathsf{Cl}, \mathrm{Act}, \mathrm{inv}', T', l_{in})$ is a *(weak) subsystem* of $\mathcal{T}$ if $\mathrm{target}, \mathrm{exit} \in \mathrm{Loc}' \subseteq \mathrm{Loc}$ holds and for all $l \in \mathrm{Loc}' \setminus \{\mathrm{target}, \mathrm{exit}\}$ we have

1. $\mathrm{inv}'(l) \Vdash \mathrm{inv}(l)$,
2. for all $\alpha \in \mathrm{Act}$: if $T'(l, \alpha) = (g', \mu')$ and $T(l, \alpha) = (g, \mu)$, then we have
   - 2a. $g' \Vdash g$, and
   - 2b. $\mu'(C, l') \in \{\, 0,\ \mu(C, l') \,\}$ for all $(C, l') \in 2^{\mathsf{Cl}} \times \mathrm{Loc}'$ with $l' \neq \mathrm{exit}$.

We call $\mathcal{T}'$ a *strong subsystem* if (2a.) can be replaced by the stronger condition

2a*. $g' \equiv g \wedge \mathrm{inv}'(l)$,

and, additionally, for all $l \in \mathrm{Loc}'$, $v \in \mathrm{Val}(\mathsf{Cl})$ and $t \in \mathbb{R}_{>0}$ we have

3. if $v \models \mathrm{inv}'(l)$ and $v + t \models \mathrm{inv}(l)$ hold, then $v + t \models \mathrm{inv}'(l)$ holds.

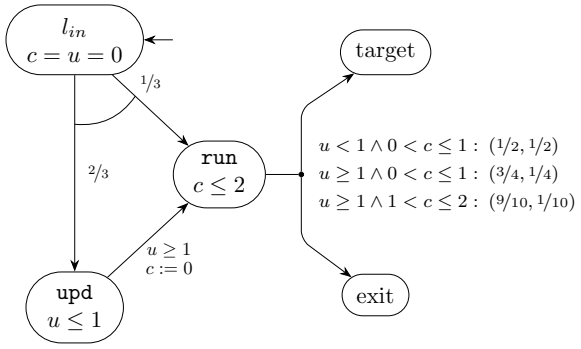


Figure 6.2: A PTA $\mathcal{T}_2$ over clocks $\{c, u\}$, using the same representation as in Figure 6.1. The single transition of location upd uses a clock reset for clock c.

Intuitively, one gets a subsystem of a PTA $\mathcal{T}$ by discarding locations, strengthening location invariants and transition guards and redirecting individual edges to the location "exit". The redirection to "exit" is implicit in the constraint (2b). It says that the probability of an edge should either coincide with that of the corresponding edge in $\mathcal{T}$, or be zero, for all edges leading to a location which is not "exit". But as μ' needs to be a probability distribution, if one chooses to set some edge probability to zero, then the same probability has to be added to an edge leading to the location "exit". Observe that by letting the guard of a transition be false (i.e., $g' = \text{false}$) in a transition in $\mathcal{T}'$, one can disable actions which were enabled in $\mathcal{T}$.

While being a subsystem is enough to witness lower bounds on $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond\,\text{target})$ (see Corollary 6.4 below), to witness lower bounds on $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond\,\text{target})$ we need the two additional constraints imposed on strong subsystems. First, guards can only shrink as much as the location invariant (2a*). In particular, this implies that transitions cannot be disabled, unless the new invariant does not overlap with the old guard g of the transition. Furthermore, the new invariant $\text{inv}'(l)$ should be closed under time successors within $\text{inv}(l)$ for all $l \in \text{Loc}'$. Together, these conditions intuitively make sure that all possibilities that a scheduler has (including the choice of time delays) in $\mathcal{T}$ are preserved in the subsystem.

Example 6.2. Consider again the PTAs $\mathcal{T}_1$ and $\mathcal{T}_2$ defined in Figures 6.1 and 6.2. Strengthening the location invariant of run in $\mathcal{T}_1$ to $c \leq 1$ yields a subsystem $\mathcal{T}_1'$. The maximal probability in $\mathcal{T}_1'$ is $1/2$, while it was $3/4$ in $\mathcal{T}_1$. The PTA $\mathcal{T}_1'$ is *not* a strong subsystem because it violates condition (3.) of Definition 6.1. To see this, observe that the valuation $v = (c \mapsto 1)$ satisfies the location invariant of $\mathcal{T}_1'$, and $v + 1$ satisfies the invariant of $\mathcal{T}_1$ but not of $\mathcal{T}_1'$.

Now consider the PTA $\mathcal{T}_2'$ which is formed by taking $\mathcal{T}_2$ and changing the invariant in location run to be $c \leq 2 \wedge u \geq 1$. This implicitly redirects the edge between l_{in} and run to "exit", as taking that edge would lead to a valuation which violates the invariant in run (see the definition of the semantics of pointed PTA in Section 2.3). One can check that the result is a strong subsystem. The minimal probability of reaching "target" in $\mathcal{T}_2'$ is $2/3 \cdot 3/4 = 1/2$. △

As for MDPs, the important property of subsystems is that the maximal and minimal reachability probabilities do not increase when passing to a subsystem. To show this, we observe that subsystems of the pointed PTA $\mathcal{T}$ induce MDP subsystems in finite-state quotients of $\mathcal{S}(\mathcal{T})$ (i.e., the semantics of $\mathcal{T}$) with respect to *probabilistic time-abstracting bisimulations* (PTABs). Details on the definition of PTAB can be found in Section 2.3.1. By the correspondence of MDP subsystems and Farkas certificates (see Theorem 4.23), it follows that we also find Farkas certificates which prove that the optimal probability in $\mathcal{T}$ is at least as high as in $\mathcal{T}'$.

Theorem 6.3 (PTA subsystems induce Farkas certificates). *Let $\mathcal{T}$ be a pointed PTA, and let $\sim$ be a PTAB on $\mathcal{T}$ which respects* target *and* exit *and has finite index. Let $\mathcal{M} = \mathcal{S}(\mathcal{T})/_\sim$ be the associated quotient MDP with states $S \cup \{\text{target}, \text{exit}\}$. Given a subsystem $\mathcal{T}'$ of $\mathcal{T}$, let $S' = \{[s] \in S \mid s \text{ is a state of } \mathcal{S}(\mathcal{T}')\}$.*

Then, there exists a Farkas certificate $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda_{\max})$ with state-supp$(\mathbf{y}) \subseteq S'$*, where $\lambda_{\max} = \mathbf{Pr}^{\max}_{\mathcal{T}'}(\Diamond \text{target})$. If $\mathcal{T}'$ is a strong subsystem, then additionally there exists a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda_{\min})$ such that* $\text{supp}(\mathbf{z}) \subseteq S'$*, where $\lambda_{\min} = \mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target})$.*

Proof. We first establish some relations between the semantics of $\mathcal{T} = (\text{Loc}, \text{Cl}, \text{Act}, \text{inv}, T, l_{in})$ and $\mathcal{T}' = (\text{Loc}', \text{Cl}, \text{Act}, \text{inv}', T', l_{in})$. For this, we denote by $S_\mathcal{T}$ the states of $\mathcal{S}(\mathcal{T})$, by $S_{\mathcal{T}'}$ the states of $\mathcal{S}(\mathcal{T}')$, by $T_{\mathcal{S}(\mathcal{T})}$ the transitions of $\mathcal{S}(\mathcal{T})$ and by $T_{\mathcal{S}(\mathcal{T}')}$ the transitions of $\mathcal{S}(\mathcal{T}')$.

(a) *$\mathcal{T}$ and $\mathcal{T}'$ have the same set of actions, and $S_{\mathcal{T}'} \subseteq S_\mathcal{T}$.*

Proof: As $\mathcal{T}$ and $\mathcal{T}'$ have the same set of actions by construction, the actions of the semantics are also the same. Then, $S_{\mathcal{T}'} \subseteq S_\mathcal{T}$ follows from $\text{Loc}' \subseteq \text{Loc}$ and point (1.) of Definition 6.1.

(b) *For any transition $(l, v) \to \mu'_{\text{sem}}$ (discrete action, or time delay) in $\mathcal{S}(\mathcal{T}')$, there exists a transition $(l, v) \to \mu_{\text{sem}}$ in $\mathcal{S}(\mathcal{T})$ such that for all $(l', C) \in \text{supp}(\mu_{\text{sem}})$ with $l' \neq \text{exit}$ we have $\mu'_{\text{sem}}(l', C) \leq \mu_{\text{sem}}(l', C)$.*

Proof: We first consider discrete transitions. Take any transition $(l, v) \xrightarrow{\alpha} \mu'_{\text{sem}} \in T_{\mathcal{S}(\mathcal{T}')}$. There must be a transition $l \xrightarrow{\alpha : g'} \mu'$ in $\mathcal{T}'$ such that $v \models g'$ and such that μ'_{sem} and μ' are related by the equalities in the definition of the semantics of PTAs. Then, by condition (2.) of Definition 6.1, there exists a transition $l \xrightarrow{\alpha : g} \mu$ of $\mathcal{T}$ such that $v \models g$ (by (2a.)). Hence, there also exists a corresponding transition $(l, v) \xrightarrow{\alpha} \mu_{\text{sem}} \in T_{\mathcal{S}(\mathcal{T})}$. From (2b) in Definition 6.1 it follows that $\mu'(C, l') \in \{\, \mu(C, l'), 0 \,\}$ for all $C \subseteq \text{Cl}$ and $l' \in \text{Loc}'$ with $l' \neq \text{exit}$. This implies that for states (l', v') of $S(\mathcal{T}')$ satisfying $l' \neq \text{exit}$ we have $\mu'_{\text{sem}}(l', v') \leq \mu_{\text{sem}}(l', v')$.

Any time delay which exists in $\mathcal{S}(\mathcal{T}')$ also exists in $\mathcal{S}(\mathcal{T})$ as the invariant of locations in $\mathcal{T}'$ implies the corresponding invariant in $\mathcal{T}$ by point (1.) of Definition 6.1.

(c) *If $\mathcal{T}'$ is a strong subsystem, then for any transition $(l, v) \to \mu_{\text{sem}}$ (discrete action, or time delay) in $\mathcal{S}(\mathcal{T})$ such that $(l, v) \in S_{\mathcal{T}'}$, there exists a transition $(l, v) \to \mu'_{\text{sem}}$ in $\mathcal{S}(\mathcal{T}')$ such that for all $(l', v') \in \text{supp}(\mu_{\text{sem}})$ with $l' \neq \text{exit}$ we have $\mu'_{\text{sem}}(l', v') \leq \mu_{\text{sem}}(l', v')$.*

Proof: We again first consider discrete actions, so take $(l, v) \xrightarrow{\alpha} \mu_{\text{sem}} \in T_{\mathcal{S}(\mathcal{T})}$. Then, there exists a corresponding transition $l \xrightarrow{\alpha : g} \mu$ in $\mathcal{T}$ satisfying $v \models g$. We use conditions (2.) and (2a*.) of Definition 6.1 to find $l \xrightarrow{\alpha : g'} \mu'$ of $\mathcal{T}'$ such that $g' \equiv g \wedge \text{inv}'(l)$. From $v \models g$ and $v \models \text{inv}'(l)$ we can derive $v \models g'$. Hence, there exists a transition $(l, v) \xrightarrow{\alpha} \mu'_{\text{sem}} \in T_{\mathcal{S}(\mathcal{T}')}$. The required relation between μ_{sem} and μ'_{sem} follows in the same way as in (b).

Now take a time delay $(l, v) \xrightarrow{t} \delta_{(l,v+t)} \in T_{\mathcal{S}(\mathcal{T})}$ such that $(l, v) \in S_{\mathcal{T}'}$. Then we have $v \models \text{inv}'(l)$ and since $(l, v{+}t) \in S_\mathcal{T}$ we have $v{+}t \models \text{inv}(l)$. By condition (3) of Definition 6.1 it follows that $v{+}t \models \text{inv}'(l)$ and hence $(l, v{+}t) \in S_{\mathcal{T}'}$. Then the transition $(l, v) \xrightarrow{t} \delta_{(l,v+t)}$ is also in $T_{\mathcal{S}(\mathcal{T}')}(l, v)$, which completes the proof.

Now let $\mathcal{M}_{S'}$ be the MDP subsystem of $\mathcal{M}$ induced by S' (see Definition 4.2). We want to establish the following chain of inequalities:

$$\mathbf{Pr}^{\max}_{\mathcal{T}'}(\Diamond \text{target}) \leq \mathbf{Pr}^{\max}_{\mathcal{M}_{S'}}(\Diamond \text{target}) \leq \mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \tag{6.1}$$

and, if $\mathcal{T}'$ is a strong subsystem:

$$\mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target}) \leq \mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target}) \leq \mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \tag{6.2}$$

In both cases, the second inequality follows from Proposition 4.4. For the first inequality we let $\mathcal{S}_{S'}$ be the timed probabilistic system (TPS) that includes exactly the states of $\mathcal{S}(\mathcal{T})$ whose equivalence class lies in S'. More precisely, let

$$\mathcal{S}_{S'} = \left(\bigcup_{[s] \in S'} [s],\ \text{Act} \cup \mathbb{R}_{>0},\ T_{S'},\ s_{in} \right),$$

where the transitions in $T_{S'}$ correspond exactly to the transitions of $\mathcal{S}(\mathcal{T})$ for the given state, with the exception that successor states that are not present in $\mathcal{S}_{S'}$ are replaced by the state "exit". As $\mathcal{S}_{S'}$ merges all states that are not in S' with exit, and elements of S' are complete equivalence classes under $\sim$, the restriction of $\sim$ to $\bigcup_{[s] \in S'} [s]$ is a PTAB on $\mathcal{S}_{S'}$. Furthermore, the corresponding quotient is $\mathcal{M}_{S'}$. Now $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S}_{S'}}(\Diamond \text{target}) = \mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}_{S'}}(\Diamond \text{target})$ follows by Lemma 2.17 for $\mathfrak{m} \in \{\min, \max\}$. It remains to show that $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}'}(\Diamond \text{target}) \leq \mathbf{Pr}^{\mathfrak{m}}_{\mathcal{S}_{S'}}(\Diamond \text{target})$ in both cases.

To show $\mathbf{Pr}^{\max}_{\mathcal{T}'}(\Diamond \text{target}) \leq \mathbf{Pr}^{\max}_{\mathcal{S}_{S'}}(\Diamond \text{target})$, it is enough to show that for every scheduler $\mathfrak{S}$ for $\mathcal{S}(\mathcal{T}')$ there exists a scheduler $\mathfrak{S}'$ for $\mathcal{S}_{S'}$ such that $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}')}(\Diamond \text{target}) \leq \Pr^{\mathfrak{S}'}_{\mathcal{S}_{S'}}(\Diamond \text{target})$. In order to prove this, take a scheduler $\mathfrak{S}$ for $\mathcal{S}(\mathcal{T}')$ and define $\mathfrak{S}'$ by mimicking $\mathfrak{S}$ on paths that exist in $\mathcal{S}(\mathcal{T}')$, and arbitrarily otherwise. This is possible by (a) and (b), as proven above, and it also directly follows by (b) that $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}')}(\Diamond \text{target}) \leq \Pr^{\mathfrak{S}'}_{\mathcal{S}_{S'}}(\Diamond \text{target})$.

To show $\mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target}) \leq \mathbf{Pr}^{\min}_{\mathcal{S}_{S'}}(\Diamond \text{target})$, we assume that $\mathcal{T}'$ is a strong subsystem. It suffices to show that for every scheduler $\mathfrak{S}'$ for $\mathcal{S}_{S'}$ there exists a scheduler $\mathfrak{S}$ for $\mathcal{S}(\mathcal{T}')$ such that $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}')}(\Diamond \text{target}) \leq \Pr^{\mathfrak{S}'}_{\mathcal{S}_{S'}}(\Diamond \text{target})$. Let $\mathfrak{S}'$ be such a scheduler for $\mathcal{S}_{S'}$ and define a scheduler $\mathfrak{S}$ for $\mathcal{S}_{\mathcal{T}'}$ by mimicking $\mathfrak{S}'$ on every path. This is possible by (a) and (c) from above, and again (c) directly implies that $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}')}(\Diamond \text{target}) \leq \Pr^{\mathfrak{S}'}_{\mathcal{S}_{S'}}(\Diamond \text{target})$. This completes the proof of equations 6.1 and 6.2.

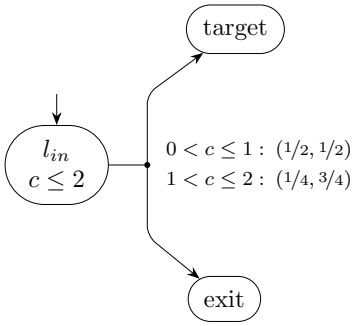
It follows that $\mathcal{M}_{S'}$ is a witnessing MDP-subsystem for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda_{\max}$, and, assuming that $\mathcal{T}'$ is a strong subsystem, also for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda_{\min}$. Then, by Theorem 4.23, there exist Farkas certificates $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda_{\max})$, respectively $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda_{\min})$, such that state-supp$(\mathbf{y}) \subseteq S'$ and supp$(\mathbf{z}) \subseteq S'$. □

As the optimal reachability probabilities in $\mathcal{T}$ coincide with the optimal reachability probabilities in the quotient of $\mathcal{S}(\mathcal{T})$ by any PTAB which respects target and exit, by Lemma 2.17, the above theorem directly yields that indeed optimal reachability probabilities cannot increase when passing to a PTA subsystem.

Corollary 6.4. *Let $\mathcal{T}$ be a pointed PTA.*

1. *If $\mathcal{T}'$ is a subsystem of $\mathcal{T}$, then $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond \text{target}) \geq \mathbf{Pr}^{\max}_{\mathcal{T}'}(\Diamond \text{target})$ holds.*
2. *If $\mathcal{T}'$ is a strong subsystem of $\mathcal{T}$, then $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond \text{target}) \geq \mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target})$ holds.*

Figure 6.3: A variation of the PTA from Figure 6.1 with different transition probabilities.



The following example shows that the assumption that $\mathcal{T}'$ is a *strong* subsystem is really necessary in point (2.) of the above corollary.

Example 6.5. Consider the PTA $\mathcal{T}$ depicted in Figure 6.3. The minimal probability of reaching target in $\mathcal{T}$ is $1/4$. Now consider the subsystem $\mathcal{T}'$ obtained by strengthening the invariant in location l_{in} to $c \leq 1$. In $\mathcal{T}'$, the minimal probability of reaching target is $1/2$, and thereby *larger* than in $\mathcal{T}$. However, $\mathcal{T}'$ is not a *strong* subsystem, because it does not satisfy the time closure condition (3) of Definition 6.1. △

6.1.2 Zone closure for difference bounds matrices

Having shown that subsystems of $\mathcal{T}$ induce Farkas certificates for quotients of $\mathcal{T}$, we now aim to show how such Farkas certificates can be translated back into subsystems of $\mathcal{T}$. Here we intuitively want Farkas certificates with small support to be mapped to small PTA subsystems. The precise meaning of this will become clear later, but in a nutshell it is the property we need to devise algorithms for the computation of small PTA based on Farkas certificates.

Location invariants can be expressed using difference bound matrices (DBMs), which are matrices in $\left((\mathbb{Z} \cup \{\infty, -\infty\}) \times \{<, \leq\}\right)^{\mathsf{Cl} \times \mathsf{Cl}}$ whose entries correspond to a bound on the difference between the values of two clocks [Dil90]. For every DBM M, there exists a unique canonical DBM M^* which includes for each pair of clocks the tightest possible constraint which does not reduce the set of satisfied clock valuations. Sets of clock valuations which are representable using DBM (or equivalently, using clock constraints) are called *zones*. For details see Section 2.3.2.

The following operation on DBMs allows to express the minimal zone which includes two zones, given as DBMs. It will be used later to define location invariants of a subsystem induced by a Farkas certificate. The maximum in the following definition is taken with respect to the standard partial order $\preceq$ on DBMs.

Definition 6.6 (Zone closure). Let L and N be DBMs over Cl. The *zone closure* $L \sqcup N$ is the DBM defined by

$$(L \sqcup N)_{ij} = \max\{\ L_{ij},\ N_{ij}\ \} \text{ for all } c_i, c_j \in \mathsf{Cl}.$$

The zone closure indeed represents the smallest zone that includes all valuations satisfying L and N, assuming that $L = L^*$ and $N = N^*$ hold.

Lemma 6.7. *Let L, N be DBMs such that $L = L^*$ and $N = N^*$. Then*

1. $\mathrm{Val}(L \sqcup N)$ *is the smallest zone in* $\mathrm{Val}(\mathsf{Cl})$ *containing* $\mathrm{Val}(L) \cup \mathrm{Val}(N)$*, and*
2. $(L \sqcup N)^* = (L \sqcup N)$.

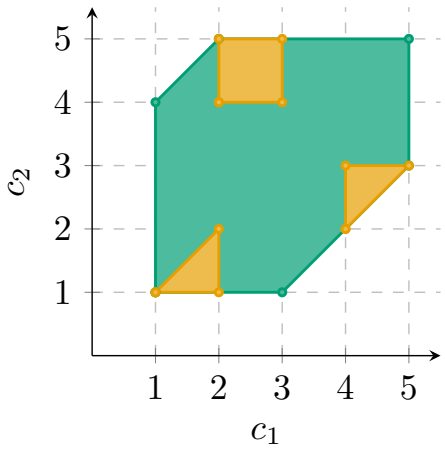


Figure 6.4: An example of the zone closure operation. The green area (including the orange zones) is the smallest set representable by a DBM which includes the three orange zones. It can be computed using the $\sqcup$ operation given DBM representations of the orange zones.

Proof. (1) Let $R = \mathrm{Val}(L) \cup \mathrm{Val}(N)$. We have $R \subseteq \mathrm{Val}(L \sqcup N)$, as if $v \in \mathrm{Val}(\mathrm{Cl})$ satisfies one of the constraints represented by L_{ij} or N_{ij}, then it also satisfies $\max\{L_{ij}, N_{ij}\}$. By point (4.) of Lemma 2.18 we have $L = M_{\mathrm{Val}(L)}$ and $N = M_{\mathrm{Val}(N)}$, and thus $L \preceq M_R$ and $N \preceq M_R$. Therefore, $L \sqcup N \preceq M_R$ holds. Now the claim follows from point (3.) of Lemma 2.18.

(2) Assume, for contradiction, that $(L \sqcup N)^* \prec (L \sqcup N)$. Then, there exist $c_i, c_j \in \mathrm{Cl}$ such that $(L \sqcup N)^*_{ij} \prec (L \sqcup N)_{ij} = \max\{L_{ij}, N_{ij}\}$. Let $(L \sqcup N)^*_{ij} = (a, \vartriangleleft_1)$ and assume, w.l.o.g., that $\max\{L_{ij}, N_{ij}\} = L_{ij} = (b, \vartriangleleft_2)$. We make the following case distinction:

(i) Assume that $a < b$ holds. Then, there is no clock valuation $v \in \mathrm{Val}(L \sqcup N) = \mathrm{Val}((L \sqcup N)^*)$ such that $v(c_i) - v(c_j) > a$. On the other hand, due to $L = L^* = M_{\mathrm{Val}(L)}$ (see Lemma 2.18, point (4.)) there exist valuations in $\mathrm{Val}(L)$ whose difference between clocks c_i and c_j is arbitrarily close to b. This yields a contradiction to $\mathrm{Val}(L) \subseteq \mathrm{Val}(L \sqcup N)$.

(ii) Assume that $a = b$, $\vartriangleleft_1 = <$ and $\vartriangleleft_2 = \leq$ hold. Again, as $L = L^* = M_{\mathrm{Val}(L)}$, there exists a valuation $v \in \mathrm{Val}(L)$ such that $v(c_i) - v(c_j) = b$. But v is not in $\mathrm{Val}(L \sqcup N)$ due to $(L \sqcup N)^*_{ij} = (b, <)$, which is again a contradiction to $\mathrm{Val}(L) \subseteq \mathrm{Val}(L \sqcup N)$. □

Given a sequence $R_1, \ldots, R_n$ of sets of clock valuations, we can express the smallest zone containing all of these sets using canonical DBMs and the zone closure operation.

Proposition 6.8. *Let $R_1, ..., R_n \subseteq \mathrm{Val}(\mathrm{Cl})$ be sets of clock valuations, $M_{R_1}, \ldots, M_{R_n}$ the corresponding canonical DBMs and $U = \bigsqcup_{i=1}^{n} M_{R_i}$. Then, $\mathrm{Val}(U)$ is the smallest zone containing all sets R_i.*

Proof. For all $1 \leq i \leq n$ we have $R_i \subseteq \mathrm{Val}(M_{R_i})$ and $M_{R_i} = M^*_{R_i}$ by Lemma 2.18. The claim follows by inductive application of Lemma 6.7. □

6.1.3 From Farkas certificates to witnessing subsystems

We now describe a construction which reverses Theorem 6.3, i.e., which computes PTA subsystems from Farkas certificates for probabilistic reachability constraints in finite-state quotients of the PTA. Of course, the constructed subsystems should *witness* the thresholds certified by the certificates on the level of the PTA. To capture this we define a notion of witnessing subsystem for PTA in analogy to the corresponding notion for MDP subsystems (see Definition 4.5). The fact that such subsystems indeed form witnesses follows from Corollary 6.4.

Definition 6.9. Let $\mathcal{T}$ be a pointed PTA and let $\lambda \in [0,1]$. A subsystem $\mathcal{T}'$ of $\mathcal{T}$ is a *witness* for $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ if it satisfies $\mathbf{Pr}^{\max}_{\mathcal{T}'}(\Diamond \text{target}) \geq \lambda$. A strong subsystem $\mathcal{T}'$ of $\mathcal{T}$ is a *witness* for $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ if it satisfies $\mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target}) \geq \lambda$.

Let $\mathcal{T}$ be a pointed PTA and $\sim$ be a PTAB on $\mathcal{T}$ which respects target and exit and has finite index. The following definition shows how to construct a subsystem of $\mathcal{T}$ induced by some subset R of equivalence classes of $\sim$. Intuitively, one first takes all locations whose states intersect some class in R, and then for each location l computes the smallest zone which includes all equivalence classes in R with location l. This then defines the location invariant for l. To obtain a *strong* subsystem, one additionally closes the invariant under time successors (using the $\uparrow$ operator, see Section 2.3), as required by Definition 6.1. Finally, edges of transitions which do not connect any pair of states in R are redirected to exit, and for weak subsystems we can additionally shrink the transition guards. The DBMs $M_{e_{|l}}$ used in the following definition are the canonical DBMs for the corresponding set of clock valuations, as defined in Section 2.3.

Definition 6.10 (Induced PTA subsystems). Let $\mathcal{T} = (\text{Loc}, \text{Cl}, \text{Act}, \text{inv}, T, l_{in})$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ with finite index which respects target and exit, and $S \cup \{\text{target}, \text{exit}\}$ be the equivalence classes of $\sim$. Given $e \in S$ and $l \in \text{Loc}$ we define $e_{|l} = \{v \in \text{Val}(\text{Cl}) \mid (l, v) \in e\}$. For $R \subseteq S$ we define the subsystems

$$\mathcal{T}^w_R = (\text{Loc}', \text{Cl}, \text{Act}, \text{inv}^w, T^w, l_{in}) \quad \text{and} \quad \mathcal{T}^s_R = (\text{Loc}', \text{Cl}, \text{Act}, \text{inv}^s, T^s, l_{in})$$

induced by R as follows. First, we set $\text{Loc}' = \{l \in \text{Loc} \mid \exists e \in R.\ e_{|l} \neq \emptyset\} \cup \{\text{target}, \text{exit}\}$. For each $l \in \text{Loc}'$ we define its location invariant using DBMs as follows:

$$\text{inv}^w(l) = M^w_l = \bigsqcup_{e \in R} M_{e_{|l}} \quad \text{and} \quad \text{inv}^s(l) = M^s_l = (\uparrow M^w_l) \sqcap M_{\text{inv}(l)}.$$

For every $l \xrightarrow{\alpha : g} \mu$ in $T(l)$ we include the transition $l \xrightarrow{\alpha : g^w} \mu'$ in $T^w(l)$, and $l \xrightarrow{\alpha : g^s} \mu'$ in $T^s(l)$, where

$$g^w = g \sqcap \bigsqcup_{\substack{e \in R \\ \exists (l,v) \in e.\ v \models g}} M_{e_{|l}} \quad \text{and} \quad g^s = g \sqcap \text{inv}^s(l),$$

and μ' is defined as follows. For $C \subseteq \text{Cl}$ and $l' \in \text{Loc}' \setminus \{\text{exit}\}$ let

$$\mu'(C, l') = \begin{cases} \mu(C, l') & \text{if } \exists e, e' \in R, (l, v) \in e.\ (l', v[C := 0]) \in e' \\ 0 & \text{otherwise} \end{cases}$$

and with the remaining probability assigned to $\mu'(\emptyset, \text{exit})$.

The PTA $\mathcal{T}^w_R$ and $\mathcal{T}^s_R$ are indeed both subsystems of $\mathcal{T}$, and $\mathcal{T}^s_R$ is additionally a strong subsystem, as the following lemma shows.

Lemma 6.11. *Let $\mathcal{T} = (\text{Loc}, \text{Cl}, \text{Act}, \text{inv}, T, l_{in})$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ with finite index which respects* target *and* exit*, and $S \cup \{\text{target}, \text{exit}\}$ be the equivalence classes of $\sim$. Then, for any $R \subseteq S$, $\mathcal{T}^w_R$ is a subsystem of $\mathcal{T}$ and $\mathcal{T}^s_R$ is a strong subsystem of $\mathcal{T}$.*

Proof. We show that $\mathcal{T}^w_R$ satisfies the conditions (1) and (2) from Definition 6.1 and $\mathcal{T}^s_R$ additionally satisfies (2*) and (3). The requirement $\text{Loc}' \subseteq \text{Loc}$ is trivially satisfied.

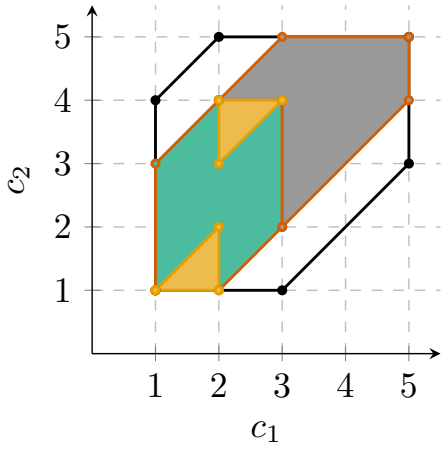


Figure 6.5: A graphical sketch which shows how $\mathrm{inv}^w(l)$ and $\mathrm{inv}^s(l)$ are constructed in Definition 6.10. Let the orange zones be equivalence classes included in R for location l, and $\mathrm{inv}(l)$ be the zone inside the outer black line. Then the green area is $\mathrm{inv}^w(l)$ (the minimal zone including the orange ones), and the gray area is the time closure of $\mathrm{inv}^w(l)$ inside $\mathrm{inv}(l)$. The green and gray areas together form $\mathrm{inv}^s(l)$.

Condition (1) requires that for all $l \in \mathrm{Loc}'$ we have $\mathrm{inv}'(l) \Vdash \mathrm{inv}(l)$. We first show this for $\mathrm{inv}^w(l) = M_l^w = \bigsqcup_{e \in R} M_{e|l}$. From Proposition 6.8 it follows that $\mathrm{Val}(\mathrm{inv}^w(l))$ is the smallest zone that contains all states in $\bigcup_{e \in R} e_{|l}$. Since this set is included in $\mathrm{Val}(\mathrm{inv}(l))$, we have $\mathrm{Val}(\mathrm{inv}^w(l)) \subseteq \mathrm{Val}(\mathrm{inv}(l))$ and hence by definition $\mathrm{inv}^w(l) \Vdash \mathrm{inv}(l)$. For $\mathrm{inv}^s(l) = M_l^s = (\uparrow M_l^w) \sqcap M_{\mathrm{inv}(l)}$, the property $\mathrm{inv}^s(l) \Vdash \mathrm{inv}(l)$ is trivial. Conditions (2) for $\mathcal{T}^w$ and (2*) and (3) for T_R^s follow immediately from the construction. □

The following proposition states that Farkas certificates for the quotient of $\mathcal{T}$ under the PTAB $\sim$ can be used to find witnesses for probabilistic reachability constraints. Hence, it provides a converse of Theorem 6.3. The constructed witnesses are the PTA subsystems induced by the support of the Farkas certificates.

Proposition 6.12. *Let $\mathcal{T} = (\mathrm{Loc}, \mathrm{Cl}, \mathrm{Act}, \mathrm{inv}, T, l_{in})$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ with finite index which respects* target *and* exit*, and $S \cup \{\mathrm{target}, \mathrm{exit}\}$ be the equivalence classes of $\sim$. Furthermore, let $\mathcal{M} = \mathcal{S}(\mathcal{T})/_\sim$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$, and fix $\lambda \in [0,1]$ and $R \subseteq S$.*

If there exists a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with $\mathrm{supp}(\mathbf{z}) \subseteq R$, then $\mathcal{T}_R^s$ is a witness for $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$. Likewise, if there exists a Farkas certificate $\mathbf{y} \in \mathcal{F}^{\max}_{\mathcal{M},\geq}(\lambda)$ with $\text{state-supp}(\mathbf{y}) \subseteq R$, then $\mathcal{T}_R^w$ is a witness for $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$.

Proof. Consider the MDP subsystem $\mathcal{M}_R$ of $\mathcal{M}$ induced by R, as defined in Definition 4.2 and suppose there exists a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ with $\mathrm{supp}(\mathbf{z}) \subseteq R$. Then, $\mathcal{M}_R$ satisfies $\mathbf{Pr}^{\min}_{\mathcal{M}_R}(\Diamond\,\mathrm{target}) \geq \lambda$, by Theorem 4.23. We now intend to show that $\mathcal{T}_R^s$ is a witness for $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$ by establishing the chain of inequalities

$$\mathbf{Pr}^{\min}_{\mathcal{T}_R^s}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\min}_{\mathcal{S}(\mathcal{T}_R^s)}(\Diamond\,\mathrm{target}) \geq \mathbf{Pr}^{\min}_{\mathcal{S}_R}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}_R}(\Diamond\,\mathrm{target}) \geq \lambda, \tag{6.3}$$

where $\mathcal{S}_R$ is the TPS that includes exactly the states of $\mathcal{S}(\mathcal{T})$ whose equivalence class lies in R (compare also the proof of Theorem 6.3). More precisely, let $\mathcal{S}_R = \left(\bigcup_{[s] \in R}[s], \mathrm{Act} \cup \mathbb{R}_{>0}, T_R, s_{in}\right)$, where the transitions in T_R correspond exactly to the transitions of $\mathcal{S}(\mathcal{T})$ for the given state, with the exception that successor states that are not present in $\mathcal{S}_R$ are replaced by the state "exit". Then the quotient of $\mathcal{S}_R$ under the restriction of $\sim$ is $\mathcal{M}_R$, and as a consequence we have $\mathbf{Pr}^{\min}_{\mathcal{S}_R}(\Diamond\,\mathrm{target}) = \mathbf{Pr}^{\min}_{\mathcal{M}_R}(\Diamond\,\mathrm{target})$ by Lemma 2.17. As the first equality in (6.3) follows from the

definition and the final inequality in (6.3) was established above, it remains to show

$$\mathbf{Pr}^{\min}_{\mathcal{S}(\mathcal{T}^s_R)}(\Diamond \text{target}) \geq \mathbf{Pr}^{\min}_{\mathcal{S}_R}(\Diamond \text{target}). \tag{6.4}$$

Observe that $\mathcal{S}_R$ does not necessarily coincide with $\mathcal{S}(\mathcal{T}^s_R)$ (the semantics of $\mathcal{T}^s_R$). This is because the set of equivalence classes in R corresponding to a location l may not form a zone. However, if an equivalence classes is present in $\mathcal{S}(\mathcal{T}^s_R)$ but not in $\mathcal{S}_R$, then the corresponding incoming transitions in $\mathcal{S}_R$ are redirected to exit. Hence, intuitively, the minimal probability of reaching target cannot be larger in $\mathcal{S}_R$ than in $\mathcal{S}(\mathcal{T}^s_R)$. We now show this formally.

First, we show that every state of $\mathcal{S}_R$ is a state of $\mathcal{S}(\mathcal{T}^s_R)$. Take a state (l, v) of $\mathcal{S}_R$. It follows that $[(l, v)] \in R$ and hence l is a location of $\mathcal{T}^s_R$. Moreover, since $\mathrm{inv}^w(l) = \bigsqcup_{s \in R} M_{s_{|l}}$, we have $v \models \mathrm{inv}^w(l)$ and therefore also $v \models \mathrm{inv}^s(l) = \uparrow(\mathrm{inv}^w(l)) \sqcap M_{\mathrm{inv}(l)}$. It follows that (l, v) is a state of $\mathcal{S}(\mathcal{T}^s_R)$.

In the following we will use the notation T_X to denote the transition function of X, which is either a PTA or a timed probabilistic system. We now show that every transition in $T_{\mathcal{S}(\mathcal{T}^s_R)}$ is matched by a transition in $T_{\mathcal{S}_R}$ which may differ only in the fact that some edges are redirected to exit in $\mathcal{S}_R$. Let $s = (l, v)$ be a state of $\mathcal{S}_R$ and $s \xrightarrow{\alpha} \mu'_{\mathrm{sem}}$ be a transition in $T_{\mathcal{S}(\mathcal{T}^s_R)}$. Then there is some transition $l \xrightarrow{\alpha : g^s} \mu' \in T_{\mathcal{T}^s_R}$. By definition of $\mathcal{T}^s_R$, there exists $l \xrightarrow{\alpha : g} \mu \in T_{\mathcal{T}}$ such that $\mu'(C, l') = \mu(C, l')$ whenever $[(l', v[C := 0])] \in R$. This induces a transition $s \xrightarrow{\alpha} \mu_{\mathrm{sem}} \in T_{\mathcal{S}(\mathcal{T})}$ and accordingly a transition $s \xrightarrow{\alpha} \overline{\mu}_{\mathrm{sem}} \in T_{\mathcal{S}_R}$ with $\overline{\mu}_{\mathrm{sem}}(t) = \mu_{\mathrm{sem}}(t) = \mu'_{\mathrm{sem}}(t)$ for all states t of $\mathcal{S}_R$. In summary, every transition of $\mathcal{S}(\mathcal{T}^s_R)$ for states in $\mathcal{S}_R$ is mirrored by a transition in $\mathcal{S}_R$ with the same distribution on states in $\mathcal{S}_R$ and remaining probability redirected to exit. Analogous reasoning shows, vice versa, that every path in $\mathcal{S}_R$ is also a path in $\mathcal{S}(\mathcal{T}^s_R)$.

In order to prove (6.4) we need to argue that for every scheduler $\mathfrak{S}$ on $\mathcal{S}(\mathcal{T}^s_R)$ there exists a scheduler $\mathfrak{S}'$ on $\mathcal{S}_R$ with $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}^s_R)}(\Diamond \text{target}) \geq \Pr^{\mathfrak{S}'}_{\mathcal{S}_R}(\Diamond \text{target})$. We can define $\mathfrak{S}'(\pi) = \mathfrak{S}(\pi)$ for every finite path π in $\mathcal{S}_R$, and with the notation above this means that a transition $s \xrightarrow{\alpha} \mu'_{\mathrm{sem}}$ will always be matched by a transition $s \xrightarrow{\alpha} \overline{\mu}_{\mathrm{sem}}$ in $\mathcal{S}_R$ under $\mathfrak{S}'$. Since $\overline{\mu}_{\mathrm{sem}}$ coincides with μ'_{sem} on the states of $\mathcal{S}_R$ and redirects the remaining probability to exit, the desired inequality $\Pr^{\mathfrak{S}}_{\mathcal{S}(\mathcal{T}^s_R)}(\Diamond \text{target}) \geq \Pr^{\mathfrak{S}'}_{\mathcal{S}_R}(\Diamond \text{target})$ follows.

The proof for the corresponding statement about $\mathcal{T}^w_R$ is analogous. □

6.2 Minimal witnessing PTA subsystems

We will now introduce three notions of minimality for subsystems of PTAs. One of them simply counts the number of locations in the subsystem and thereby resembles the notion of state-minimality in MDPs. The other two take the timing aspect of PTA into account. The idea is to say that a subsystem is smaller if it puts stronger restrictions on the clock valuations that are allowed in a location.

6.2.1 Notions of minimality for PTA subsystems

As before, let $\mathsf{Cl} = \{c_0, c_1, \ldots, c_n\}$ be a set of clocks, where c_0 is the special zero clock which must be interpreted by zero in all valuations. For a set of valuations $R \subseteq \mathrm{Val}(\mathsf{Cl})$ we denote by $\mathrm{vol}(R)$ the Lebesgue volume of R considered as a subset of $\mathbb{R}^{\mathsf{Cl} \setminus \{c_0\}}$. The *volume* of a PTA $\mathcal{T}$

over clocks Cl is defined as

$$\mathrm{vol}(\mathcal{T}) = \sum_{l \in \mathrm{Loc}(\mathcal{T})} \mathrm{vol}\left(\mathrm{Val}(\mathrm{inv}(l))\right) \in \mathbb{R}_{\geq 0} \cup \{\infty\}.$$

Definition 6.13. Let $\mathcal{T}$ be a PTA. We define the three preorders $\leq_{\mathrm{loc}}$, $\leq_{\mathrm{inv}}$ and $\leq_{\mathrm{vol}}$ on subsystems $\mathcal{T}_1$ and $\mathcal{T}_2$ of $\mathcal{T}$ as follows:

1. $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ iff $|\mathrm{Loc}(\mathcal{T}_1)| \leq |\mathrm{Loc}(\mathcal{T}_2)|$ holds,
2. $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$ iff $\mathrm{Loc}(\mathcal{T}_1) \subseteq \mathrm{Loc}(\mathcal{T}_2)$ and for all $l \in \mathrm{Loc}(\mathcal{T}_1)$: $\mathrm{inv}_{\mathcal{T}_1}(l) \Vdash \mathrm{inv}_{\mathcal{T}_2}(l)$, and
3. $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$ iff $\mathrm{vol}(\mathcal{T}_1) \leq \mathrm{vol}(\mathcal{T}_2)$ holds.

We say that a witness $\mathcal{T}'$ of $\mathcal{T}$ as defined in Definition 6.9 is *loc-minimal* (respectively, *inv-minimal* or *vol-minimal*) if $\mathcal{T}'$ is a $\leq_{\mathrm{loc}}$-minimal element (respectively, $\leq_{\mathrm{inv}}$-minimal or $\leq_{\mathrm{vol}}$-minimal element) among all witnesses of $\mathcal{T}$ for the same property.

While the invariant order compares the logical strength of location invariants location wise, the volume order compares the sum of volumes of location invariants. Its main benefit is that it yields a total relation, while we have to expect many incomparable minimal witnesses under the invariant order.

Lemma 6.14. *Let $\mathcal{T}$ be a pointed PTA, $\leq_{\mathrm{inv}}$, $\leq_{\mathrm{loc}}$ and $\leq_{\mathrm{vol}}$ be the preorders as defined above and $\mathcal{T}_1, \mathcal{T}_2$ be arbitrary subsystems of $\mathcal{T}$. We have*

- *if $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$ holds, then so does $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ and $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$, and*
- *$\leq_{\mathrm{vol}}$ and $\leq_{\mathrm{loc}}$ are incomparable in general.*

Proof. Let $\mathcal{T}_1, \mathcal{T}_2$ be PTAs satisfying $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$. Then $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ follows directly from $\mathrm{Loc}(\mathcal{T}_1) \subseteq \mathrm{Loc}(\mathcal{T}_2)$ and $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$ follows from $\mathrm{inv}_{\mathcal{T}_1}(l) \Vdash \mathrm{inv}_{\mathcal{T}_2}(l)$ for all $l \in \mathrm{Loc}(\mathcal{T}_1)$.

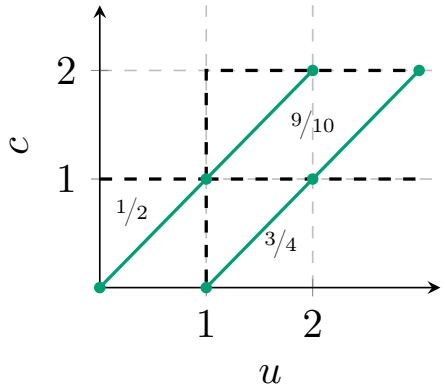
By considering two PTAs with a single location and different invariants, it becomes clear that $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ does not imply $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$ nor $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$. To see that $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$ does not imply $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ or $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$ in general, it suffices to arrange $\mathcal{T}_1$ to have one location more than $\mathcal{T}_2$ but less volume in total. □

The above lemma does not imply that inv-minimal witnesses are loc-minimal or vol-minimal. This is because an inv-minimal witness might be $\leq_{\mathrm{inv}}$-incomparable to witnesses with smaller volume.

Example 6.15. Consider again the PTAs $\mathcal{T}_1$ and $\mathcal{T}_2$ shown in Figures 6.1 and 6.2. While $\mathcal{T}_1$ does not have any strong subsystems (apart from $\mathcal{T}_1$ itself), the subsystem one gets by strengthening the location invariant in `run` to $c < 1$ is a witness for the property $\mathbf{Pr}^{\max}_{\mathcal{T}_1}(\Diamond\,\mathrm{target}) \geq 1/2$. It is inv-minimal, as the location invariant in `run` cannot be strengthened further while still intersecting the guard $0 < c \leq 1$ of the first transition. It is also vol-minimal for the same reason. Observe that the subsystem one gets by setting the invariant in `run` to be $c \leq 1$ is also vol-minimal, as the two do not differ with respect to their volume.

Now consider the PTA $\mathcal{T}_2$. The subsystem induced by locations $\{l_{in}, \mathtt{run}\}$ (i.e., excluding `upd`), is a loc-minimal witness for both properties $\mathbf{Pr}^{\max}_{\mathcal{T}_2}(\Diamond\,\mathrm{target}) \geq 3/10$ and $\mathbf{Pr}^{\min}_{\mathcal{T}_2}(\Diamond\,\mathrm{target}) \geq 1/6$. The difference between the two is that for minimal probabilities one has to consider the transition

Figure 6.6: The plot shows the reachable clock valuations in location run of PTA $\mathcal{T}_2$, as defined in Figure 6.2. Additionally, the probabilities of taking the only enabled transition is shown for the different regions of clock valuations.



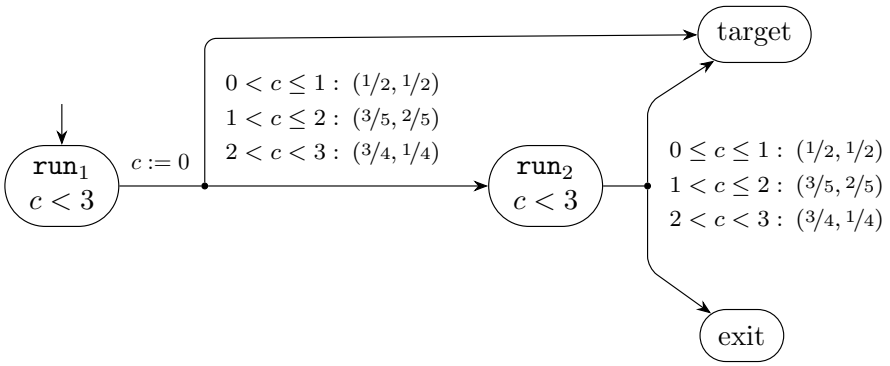


Figure 6.7: A PTA $\mathcal{T}_3$ with two locations, used in Example 6.16.

in run with lowest probability of reaching target (this is 1/2), while for maximal probabilities, the transition with highest probability counts (this is 9/10).

Figure 6.6 shows the clock valuations which are reachable in location run in $\mathcal{T}_2$. Additionally, the probabilities of taking the transition to "target" are shown for the different clock regions. In $\mathcal{T}_2$, the valuation $(c \mapsto 0, u \mapsto 0)$ in location run is reached with probability 1/3, while $(c \mapsto 0, u \mapsto 1)$ is reached with probability 2/3. An inv-minimal witness for $\mathbf{Pr}^{\min}_{\mathcal{T}_2}(\Diamond \text{target}) \geq 1/2$ is obtained by strengthening the invariant in run to exclude the left line in Figure 6.6. This is achieved by the clock constraint $c \leq 2 \ \wedge \ u - c = 1$. The same subsystem is a witness for $\mathbf{Pr}^{\max}_{\mathcal{T}_2}(\Diamond \text{target}) \geq 9/10$. By further strengthening the invariant to $c < 1 \ \wedge \ u - c = 1$ one gets an inv-minimal witness for $\mathbf{Pr}^{\max}_{\mathcal{T}_2}(\Diamond \text{target}) \geq 1/2$. This subsystem is not strong, however, as it does not satisfy the time closure condition (point (3) of Definition 6.1). △

Example 6.16. Consider the PTA $\mathcal{T}_3$ as shown in Figure 6.7. It models the following scenario. A server runs a computation, and whether it completes successfully depends on how long the server runs. If the computation is not successful, the server may restart once. This is modeled by moving to location run_2. Both locations run_1 and run_2 have volume three (the volume of $\{c \mid 0 \leq c \leq 3\} \subseteq \mathbb{R}$ is three), and hence $\text{vol}(\mathcal{T}_3) = 6$. A vol-minimal witness for the property $\mathbf{Pr}^{\max}_{\mathcal{T}_3}(\Diamond \text{target}) \geq 3/4$ is the subsystem obtained by strengthening the invariants in both locations to $c < 1$. This subsystem has volume two. Now consider a second subsystem which excludes run_2 completely, but does not strengthen the invariant in run_2 at all. This is an inv-minimal witness for $\mathbf{Pr}^{\max}_{\mathcal{T}_3}(\Diamond \text{target}) \geq 3/4$, as strengthening the invariant in run_2 would lead to a subsystem with maximal probability below 3/4. However, the volume of this second

subsystem is three. This shows that inv-incomparable subsystems may have different volume. To answer the question "what is the least number of time-units that is required to achieve a maximal probability of 3/4", vol-minimality is the appropriate notion of size in this example. △

6.2.2 Computing minimal witnesses

In general, computing minimal witnesses for PTA is at least as hard as deciding the corresponding threshold property, as the latter is equivalent to deciding whether any witness exists at all. It is known that deciding $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond \text{target}) = 1$ is EXPTIME-hard [LS07, Theorem 3.1] for PTAs. In [AD94, Theorem 4.17] it is shown that standard non-probabilistic reachability is PSPACE-hard, and, as a direct consequence of their proof, this holds also for time-bounded reachability. This problem can be reduced to the problem of deciding $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond \text{target}) < 1$. Hence, deciding $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond \text{target}) = 1$ is also PSPACE-hard. It follows that we cannot expect algorithms which require less than exponential time even to compute any witness for such threshold properties.

In the following, let $\mathcal{T}$ be a pointed PTA, the relation $\sim$ be a PTAB on $\mathcal{T}$ which respects target and exit and has finite index. Furthermore, let $\mathcal{M} = \mathcal{S}(\mathcal{T})/_{\sim} = (S \cup \{\text{target}, \text{exit}\}, \text{Act}, s_{in}, P)$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$. Additional assumptions on $\mathcal{T}$ and $\sim$ will be placed in the following sections, depending on which notion of size we consider.

Computing loc-minimal witnesses

For loc-minimality we will assume that whenever $(l_1, v_1) \sim (l_2, v_2)$, then $l_1 = l_2$. Or, in other words, the bisimulation $\sim$ distinguishes locations. As before, we will use $[(l, v)]$ to denote the equivalence class of the state $(l, v) \in \text{Loc} \times \text{Val}(\text{Cl})$.

The general plan to compute loc-minimal witnesses is to reduce the problem to the labeled witness problem for MDPs. To this end we define the labeling function $\Lambda_{\text{loc}} : S \to 2^{\text{Loc}}$ of $\mathcal{M}$ (the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$) as follows:

$$\Lambda_{\text{loc}}\big(\ [(l, v)]\ \big) \quad = \quad \{\ l\ \}.$$

Here we use Loc as the finite set of labels and simply label each equivalence class of $\sim$ by the location it belongs to. This is well defined by our assumption that $\sim$ distinguishes locations.

Proposition 6.17. *Let $\mathcal{T}$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ which respects* target *and* exit *and distinguishes locations and $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$. Then, for all $\mathfrak{m} \in \{\min, \max\}$ and $\lambda \in [0, 1]$:*

There exists a witnessing subsystem for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ with at most k locations (excluding target *and* exit*) if and only if there exists a witnessing subsystem $\mathcal{M}'$ for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{M}}(\Diamond \text{target}) \geq \lambda$ such that $|\Lambda_{\text{loc}}(\mathcal{M}')| \leq k$.*

Proof. We show only the case of $\mathfrak{m} = \min$, as the other case is analogous.

"$\Longrightarrow$": Let $\mathcal{T}'$ be a strong subsystem of $\mathcal{T}$ with at most k locations and assume that $\mathbf{Pr}^{\min}_{\mathcal{T}'}(\Diamond \text{target}) \geq \lambda$ holds. Let $S' = \{[s] \in S \mid s \text{ is a state in } \mathcal{S}(\mathcal{T}')\}$. Then, by Theorem 6.3, there exists a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ satisfying $\text{supp}(\mathbf{z}) \subseteq S'$. It follows by Theorem 4.23 that the subsystem $\mathcal{M}_{S'}$ induced by S' satisfies $\mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target}) \geq \lambda$. But as $\mathcal{T}'$ has at most k locations, we get $|\Lambda_{\text{loc}}(\mathcal{M}_{S'})| \leq k$.

"$\Longleftarrow$": Let $S' \subseteq S$ and assume that $\mathcal{M}_{S'}$ satisfies $\mathbf{Pr}^{\min}_{\mathcal{M}_{S'}}(\Diamond \text{target}) \geq \lambda$ and $|\Lambda_{\text{loc}}(\mathcal{M}_{S'})| \leq k$. By Theorem 4.23 we find a Farkas certificate $\mathbf{z} \in \mathcal{F}^{\min}_{\mathcal{M},\geq}(\lambda)$ such that $\text{supp}(\mathbf{z}) \subseteq S'$. It

follows from Proposition 6.12 that the PTA $\mathcal{T}^s_{\mathrm{supp}(\mathbf{z})}$ is a witness for $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$. As $|\Lambda_{\mathrm{loc}}(\mathcal{M}_{S'})| \leq k$, S' includes equivalence classes from at most k locations, which implies that $\mathcal{T}^s_{\mathrm{supp}(\mathbf{z})}$ has at most k locations. □

As the MDP $\mathcal{M}$ will in general be exponentially larger than $\mathcal{T}$, one might think that computing loc-minimal witnesses using this method leads to a double exponential blow up, as the witnesses problem for MDPs is NP-complete. However, the fact that the locations are very few in general and the integer variables used in the MILP computing minimal witnesses for MDP (see Definition 4.35) correspond to the number of labels, we get a single exponential upper bound.

Proposition 6.18. *Let $\mathcal{T}$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ which respects* target *and* exit *and distinguishes locations and let $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$.*

Then for all $\mathfrak{m} \in \{\min, \max\}$ and $\lambda \in [0,1]$ a loc-minimal witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$ can be computed in time $O(2^{|\,\mathrm{Loc}\,|} \cdot \mathrm{poly}(|\mathcal{M}|))$.

Proof. By Proposition 6.17 we can equivalently compute a label-minimal witnessing subsystem for $\mathcal{M}$ and the labeling function Λ_{loc}. The labeled witness problem for MDP $\mathcal{N}$ with finite set of labels L can be solved in time $O(2^{|L|} \cdot \mathrm{poly}(|\mathcal{N}|))$. An algorithm which achieves this enumerates all subsets $L' \subseteq L$, computes the subsystem $\mathcal{N}_{L'}$ which excludes exactly the states labeled by some $l \notin L'$ and computes the probability (either minimal or maximal) achieved by $\mathcal{N}_{L'}$. Then, from the subsystems $\mathcal{N}_{L'}$ which satisfy the probabilistic reachability constraint one can pick any one with minimal $|L'|$. □

The reduction to the labeled witnessing problem for MDPs enables using all the techniques that have been developed to compute witnessing subsystem for MDPs. In particular, one can use the quotient-sum heuristic (Section 4.2.3) to compute small witnesses, rather than minimal ones. Furthermore, we want to emphasize that any PTAB which distinguishes locations can be used for the above reduction. Hence, the quotient MDPs may be considerably smaller than the quotient one gets by using the region equivalence.

Computing inv-minimal witnesses

When considering inv- and vol-minimality, we will assume that $\mathrm{Val}(\mathrm{inv}(l))$ is bounded for every location $l \in \mathrm{Loc}$ or, equivalently, that a finite upper bound K on the value of all clocks exists. This guarantees that the set of witnesses that we have to consider is finite, and, for vol-minimality, that their volume is finite. An important application that justifies this assumption is *time-bounded* reachability, where target needs to be reached before an absolute time-bound K.

While for loc-minimality we assumed that $\sim$ distinguishes locations, now we additionally assume that if $(l_1, v_1) \sim (l_2, v_2)$ holds, then there is no clock constraint γ such that $v_1 \models \gamma$ and $v_2 \not\models \gamma$. So, $\sim$-equivalent valuations must be indistinguishable by clock constraints. The coarsest PTAB that achieves this is the region equivalence, and hence we will say that a PTAB *distinguishes regions* if it satisfies this property. As before, let $\mathcal{T}$ be a pointed PTA and $\mathcal{M} = (S \cup \{\mathrm{target}, \mathrm{exit}\}, \mathrm{Act}, s_{in}, P)$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$.

To encode invariant strength, we will use $B = 4K{+}1$ distinct labels for each location $l \in \mathrm{Loc}$ and ordered pair of clocks $c_i, c_j \in \mathrm{Cl}$. Here K is the global upper bound on any clock value

which we have assumed to exist. In total, this gives $B \cdot |\mathrm{Loc}| \cdot |\mathsf{Cl}|^2$ labels. We will denote the labels by $\xi^l_{ij}(k)$, for $k \in \{-2K, \dots, 2K\}$, and the set of labels by

$$L \;=\; \big\{\, \xi^l_{ij}(k) \;\mid\; c_i, c_j \in \mathsf{Cl},\; l \in \mathrm{Loc},\; k \in \{-2K, \dots, 2K\} \big\}.$$

For a given equivalence class $[(l, v)] \in S$, let $M_{[(l,v)]}$ be the canonical DBM for the corresponding set of valuations (see Section 2.3.2). We define the labeling function $\Lambda^{\min}_{\mathrm{inv}} : S \to 2^L$ as follows:

$$\Lambda^{\min}_{\mathrm{inv}}\big([(l,v)]\big) \;=\; \bigcup_{\substack{c_i, c_j \in \mathsf{Cl} \\ c_j \neq c_0}} \Big(\big\{\, \xi^l_{ij}(2a) \;\mid\; (a, \leq) \leq (M_{[(l,v)]})_{ij},\; -K \leq a \leq K \,\big\} \cup \big\{\, \xi^l_{ij}(2a{-}1) \;\mid\; (a, <) \leq (M_{[(l,v)]})_{ij},\; -K \leq a \leq K \,\big\} \Big).$$

The labeling function $\Lambda^{\max}_{\mathrm{inv}} : S \to 2^L$ is defined as above but with the large union ranging over all $c_i, c_j \in \mathsf{Cl}$, including the case $c_j = c_0$. In a DBM M, the entry $M_{i0} = (a, \triangleleft)$ represents the constraint $v(c_i) - v(c_0) \triangleleft a$. This is equivalent to $v(c_i) \triangleleft a$, as c_0 is interpreted by zero in all valuations. Hence, these entries correspond to absolute upper bounds on the clocks. In the min-case reducing the upper bound on a clock c_i should not lead to a better score in the labeling function as the invariants of strong subsystems anyway have to be closed under time successors within the invariant of $\mathcal{T}$ (see Definition 6.1). This is the reason for excluding $c_j = 0$ in the definition of $\Lambda^{\min}_{\mathrm{inv}}$.

The set of labels $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}([l, v])$ corresponds tightly to the canonical DBM $M_{[(l,v)]}$ for states (l, v) and $\mathfrak{m} \in \{\min, \max\}$. Take arbitrary $l \in \mathrm{Loc}$, $c_i, c_j \in \mathsf{Cl}$ such that $c_j \neq c_0$ and let $v \in \mathrm{Val}(\mathsf{Cl})$ and $e = [(l, v)]$. For all $v \in \mathrm{Val}(\mathsf{Cl})$, the following holds by construction for all $-2K \leq b \leq 2K$:

$$\text{if} \quad \xi^l_{ij}(b) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(e), \;\; \text{then} \;\; \xi^l_{ij}(b') \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(e) \;\; \text{for all} \;\; b' \in \{-2K, \dots, b\}. \tag{6.5}$$

Intuitively, we have $\xi^l_{ij}(2b{-}1) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(e)$ if the upper bound on $v(c_i) - v(c_j)$ that the canonical DBM for e defines is at least b.

Example 6.19. Let $e \in S$ be an equivalence class of $\sim$, $c_i, c_j \in \mathsf{Cl}$ with $c_j \neq c_0$ and assume that $(M_e)_{ij} = (0, \leq)$ (here M_e represents the canonical DBM for $\{v \in \mathrm{Val}(\mathsf{Cl}) \mid (v, l) \in e\}$). This means that $v(c_i) - v(c_j) \leq 0$ holds for all valuations $(v, l) \in e$ (we assume here that all states of e belong to the same location $l \in \mathrm{Loc}$). Then, $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}(e)$ will include all the labels

$$\{\, \xi^l_{ij}(-2K),\; \xi^l_{ij}(-2K+1),\; \dots,\; \xi^l_{ij}(0) \,\}.$$

By the observation above, the "last" element $\xi^l_{ij}(b)$ included in $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}(e)$ (i.e., such that b is largest) determines exactly which $\xi^l_{ij}(b')$ are included in the labeling for the pair of clocks c_i, c_j. If we had $(M_e)_{ij} = (0, <)$, then this last element would be $\xi^l_{ij}(-1)$. For $(M_e)_{ij} = (5, \leq)$, it would be $\xi^l_{ij}(10)$ and for $(M_e)_{ij} = (-3, <)$, it would be $\xi^l_{ij}(-7)$.

In this way, the last element $\xi^l_{ij}(b)$ precisely determines the entry of $(M_e)_{ij}$. Observe that there are exactly $4K + 2$ possible values of $(M_e)_{ij}$, given that K is an absolute upper bound for any clock (and hence also for difference of any two clocks). △

We have seen above how the "last" element $\xi^l_{ij}(b)$ included in the set $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}([(l, v)])$ corresponds to the entry $(M_{[l,v]})_{ij}$ of the canonical DBM. Using this idea, one can read of all location invariants of the strong subsystem $\mathcal{T}^s_{S'}$ of $\mathcal{T}$ from the set of labels $\Lambda^{\min}_{\mathrm{inv}}(\mathcal{M}_{S'})$ for some subsystem

$\mathcal{M}_{S'}$ of $\mathcal{M}$ (and similarly for maximal probabilities). The following lemmas depend crucially on the fact that K is an upper bound on all clock evaluations, and hence also on all differences between two clocks. This implies that the DBMs representing the location invariants of $\mathcal{T}$ have no numerical entry larger than K or less than $-K$.

Lemma 6.20. *Let $S' \subseteq S$, $\mathfrak{m} \in \{\min, \max\}$ and $L' = \Lambda^{\mathfrak{m}}_{\text{inv}}(\mathcal{M}_{S'})$. For all $l \in \text{Loc}$ and $c_i, c_j \in \text{Cl}$, let b^l_{ij} be the maximal number $b \in \{-2K, \dots, 2K\}$ such that $\xi^l_{ij}(b) \in L'$.*

Then, the location invariant of PTA subsystem $\mathcal{T}^s_{S'}$ in l is equivalent to the DBM M defined for all $c_i, c_j \in \text{Cl}$ with $c_j \neq c_0$ by

$$M_{ij} = \begin{cases} (\lceil b^l_{ij}/2 \rceil, \leq) & \text{if } b^l_{ij} \text{ is even,} \\ (\lceil b^l_{ij}/2 \rceil, <) & \text{otherwise,} \end{cases}$$

and $M_{i0} = \big(M_{\text{inv}_{\mathcal{T}}}(l)\big)_{i0}$. The location invariant of PTA subsystem $\mathcal{T}^w_{S'}$ in l is equivalent to M as defined above but with the definition ranging over all $c_i, c_j \in \text{Cl}$ (including $c_j = c_0$).

Proof. The location invariant in location l of PTA subsystem $\mathcal{T}^w_{S'}$ induced by S' is defined to be $M^w_l = \bigsqcup_{e \in S'} M_{e|_l}$, where $e|_l = \{v \in \text{Val}(\text{Cl}) \mid (l, v) \in e\}$ (see Definition 6.10). For a given pair of clocks c_i, c_j, let $(a, \triangleleft)$ be the maximal element in $\big\{\big(M_{e|_l}\big)_{ij} \mid e \in S'\big\}$ with respect to the order $\preceq$. By definition of the zone closure operation, we have $\big(M^w_l\big)_{ij} = (a, \triangleleft)$. At the same time, the largest $b \in \{-2K, \dots, 2K\}$ such that $\xi^l_{ij}(b) \in \Lambda^{\max}_{\text{inv}}(\mathcal{M}_{S'})$ is $b = 2a$ if $\triangleleft = \leq$, or $b = 2a-1$ if $\triangleleft = <$. This shows the claim for $\mathcal{T}^w_{S'}$.

For $\mathcal{T}^s_{S'}$, the invariant is defined by $M^s_l = (\uparrow M^w_l) \sqcap M_{\text{inv}_{\mathcal{T}}(l)}$. The $\uparrow$ operation on DBMs is realized by replacing the upper bound for all clocks c_i (which correspond to the index $i0$ in the DBM) by ∞. After intersecting with $M_{\text{inv}_{\mathcal{T}}(l)}$ the resulting upper bound is the one from $M_{\text{inv}(l)}$ (which is in the range $\{-K, \dots, K\}$). Hence, all entries of M^s_l correspond to the entry of M^w_l apart from $(M^s_l)_{i0}$ for all $c_i \in \text{Cl}$, which is equal to $(M_{\text{inv}_{\mathcal{T}}(l)})_{i0}$. This concludes the proof. □

With the above lemma in place, we can show that label-inclusion of subsystems of $\mathcal{M}$ corresponds to the invariant order on PTA subsystems of $\mathcal{T}$.

Lemma 6.21. *For all $S_1, S_2 \subseteq S$ we have*

- $\Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_1}) \subset \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_2})$ *if and only if* $\mathcal{T}^s_{S_1} <_{\text{inv}} \mathcal{T}^s_{S_2}$, *and*
- $\Lambda^{\max}_{\text{inv}}(\mathcal{M}_{S_1}) \subset \Lambda^{\max}_{\text{inv}}(\mathcal{M}_{S_2})$ *if and only if* $\mathcal{T}^w_{S_1} <_{\text{inv}} \mathcal{T}^w_{S_2}$.

Proof. We show the statement in the first bullet point, the other one follows analogously.

"$\Longrightarrow$": Let $l \in \text{Loc}$ and M_1, M_2 be the DBMs representing $\text{inv}_{\mathcal{T}^s_{S_1}}(l)$ and $\text{inv}_{\mathcal{T}^s_{S_2}}(l)$ respectively. By Lemma 6.20 and $\Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_1}) \subset \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_2})$ we immediately get $M_1 \preceq M_2$. Furthermore, if $\Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_2}) \setminus \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_1}) \neq \emptyset$, then we have $(M_1)_{ij} \prec (M_2)_{ij}$. Here we have used the monotonicity property shown in Equation (6.5).

"$\Longleftarrow$": We show for all $l \in \text{Loc}$ and $c_i, c_j \in \text{Cl}$ that

$$\big\{\xi^l_{ij}(b) \in \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_1}) \mid b \in \{-2K, \dots, 2K\}\big\} \subseteq \big\{\xi^l_{ij}(b) \in \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_2}) \mid b \in \{-2K, \dots, 2K\}\big\}.$$

Let $l \in \text{Loc}$ and M_1, M_2 be the DBMs representing $\text{inv}_{\mathcal{T}^s_{S_1}}(l)$ and $\text{inv}_{\mathcal{T}^s_{S_2}}(l)$ respectively. By assumption, we have $(M_1)_{ij} \preceq (M_2)_{ij}$ for all $c_i, c_j \in \text{Cl}$. Hence the largest b such that $\xi^l_{ij}(b) \in \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_1})$ is less or equal to the largest b such that $\xi^l_{ij}(b) \in \Lambda^{\min}_{\text{inv}}(\mathcal{M}_{S_2})$ by Lemma 6.20. The

subset relation above follows by the monotonicity as shown in Equation (6.5). Similarly to the other case, if $(M_1)_{ij} \prec (M_2)_{ij}$ holds for some c_i, c_j and $l \in \mathrm{Loc}$, then the subset relation is strict. □

The label-based witness problem for $\mathcal{M}$ with respect to labeling function $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}$ (with $\mathfrak{m} \in \{\min, \max\}$) searches for a witnessing subsystem of $\mathcal{M}$ such that $|\Lambda^{\mathfrak{m}}_{\mathrm{inv}}(\mathcal{M})|$ is minimal. By the above lemma, such a subsystem induces an inv-minimal witnessing PTA subsystem of $\mathcal{T}$ for the corresponding threshold property. This gives us the following proposition.

Proposition 6.22. *Let $\mathcal{T}$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ with finite index which respects* target *and* exit *and distinguishes regions. Let $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$. Then, for all $\lambda \in [0,1]$ and $S' \subseteq S$:*

- *if $\mathcal{M}_{S'}$ is a label-minimal witness for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\mathrm{target}) \geq \lambda$ with respect to $\Lambda^{\min}_{\mathrm{inv}}$, then $\mathcal{T}^s_{S'}$ is an inv-minimal witness for $\mathbf{Pr}^{\min}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$.*
- *if $\mathcal{M}_{S'}$ is a label-minimal witness for $\mathbf{Pr}^{\max}_{\mathcal{M}}(\Diamond\,\mathrm{target}) \geq \lambda$ with respect to $\Lambda^{\max}_{\mathrm{inv}}$, then $\mathcal{T}^w_{S'}$ is an inv-minimal witness for $\mathbf{Pr}^{\max}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$.*

Proof. We show the first claim, the second one follows analogously. If $\mathcal{M}_{S'}$ is a witnessing subsystem for $\mathbf{Pr}^{\min}_{\mathcal{M}}(\Diamond\,\mathrm{target}) \geq \lambda$, then $\mathcal{T}^s_{S'}$ is a strong subsystem satisfying $\mathbf{Pr}^{\min}_{\mathcal{T}^s_{S'}}(\Diamond\,\mathrm{target}) \geq \lambda$ by Theorem 4.23 and Proposition 6.12. So suppose, for contradiction, that $\mathcal{T}^s_{S'}$ is not inv-minimal. Then, there exists another witness $\mathcal{T}'$ for the same property satisfying $\mathcal{T}' <_{\mathrm{inv}} \mathcal{T}^s_{S'}$. As $\sim$ distinguishes regions, and hence no clock invariant can distinguish $\sim$-equivalent states, we may assume that $\mathcal{T}'$ is induced by a set of equivalence classes $R \subseteq S$. That is, we have $\mathcal{T}' = \mathcal{T}^s_R$. By Lemma 6.21, we have $\Lambda^{\min}_{\mathrm{inv}}(\mathcal{M}_R) \subset \Lambda^{\min}_{\mathrm{inv}}(\mathcal{M}_{S'})$. But this contradicts label-minimality of $\mathcal{M}_{S'}$. □

In the MILP used to compute label-minimal subsystems for MDPs in Definition 4.35, the number of integer variables corresponds to the number of labels in the labeling function. With $B = 4K + 1$, the number of labels ξ^l_{ij} that we have introduced is $B \cdot |\,\mathrm{Loc}\,| \cdot |\mathrm{Cl}|^2$. However, due to the monotonicity of labels as given by Equation (6.5) there are only B possible label configurations for every location and pair of clock constraints. Hence, the number of label-configurations one has to enumerate to find an optimal one is bounded by $(B+1)^{|\,\mathrm{Loc}\,|\cdot|\mathrm{Cl}|^2}$ (rather than $2^{B\cdot|\,\mathrm{Loc}\,|\cdot|\mathrm{Cl}|^2}$). The "+1" accounts for the case that no label is included for some location and clock pair. Observe that the number of possible values of an entry in a DBM is $4K + 2 = B + 1$, under the assumption that K is an upper bound on all clocks.

In a similar way as for Proposition 6.18 it follows that an inv-minimal witness can be computed in single exponential time (if K, encoded in binary, is assumed to be part of input).

Proposition 6.23. *Let $\mathcal{T}$ be a pointed PTA, $\sim$ be a PTAB on $\mathcal{T}$ which respects* target *and* exit*, has finite index and distinguishes regions, and let $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$. Furthermore, let K be an upper bound on the possible value of any clock in any location and let $B = 4K + 1$.*

Then, for all $\mathfrak{m} \in \{\min, \max\}$ and $\lambda \in [0,1]$, an inv-minimal witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond\,\mathrm{target}) \geq \lambda$ can be computed in time $O\big(2^{(\log(B+1)\cdot|\,\mathrm{Loc}\,|\cdot|\mathrm{Cl}|^2)} \cdot \mathrm{poly}(|\mathcal{M}|)\big)$, if one exists.

Proof. It suffices to enumerate all relevant label-subsets $L' \subseteq L$, where L is the entire set of labels, and compute the optimal probability of the MDP one gets by excluding all states that are labeled by some label in $L \setminus L'$. Computing this value is doable in polynomial time in $|\mathcal{M}|$. The number of relevant label subsets is bounded by $O\big((B+1)^{|\,\mathrm{Loc}\,|\cdot|\mathrm{Cl}|^2}\big) = O\big(2^{\log(B+1)\cdot|\,\mathrm{Loc}\,|\cdot|\mathrm{Cl}|^2}\big)$, due to the monotonicity of labels given in Equation (6.5). □

Computing vol-minimal witnesses

As for inv-minimality, we will assume that $\sim$ is a PTAB for $\mathcal{T}$ with finite index which distinguishes regions. We let $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$ and assume that K is an upper bound on all clocks. The following lemma shows that the set of inv-minimal witnesses includes a vol-minimal witness.

Lemma 6.24. *Let $\mathcal{T}$ be a pointed PTA. For all $\mathfrak{m} \in \{\min, \max\}$ and $\lambda \in [0,1]$, there is at least one witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ which is both inv- and vol-minimal.*

Proof. Assume first that there exists a vol-minimal witness with finite volume and suppose that the sets of vol- and inv-minimal witnesses are disjoint. Then, for each vol-minimal witness $\mathcal{T}_1$ there must exist another witness $\mathcal{T}_2$ such that $\mathcal{T}_2 <_{\text{inv}} \mathcal{T}_1$, as otherwise $\mathcal{T}_1$ would be inv-minimal. By definition of $\leq_{\text{inv}}$ it follows that $\text{vol}(\mathcal{T}_2) \leq \text{vol}(\mathcal{T}_1)$ and as $\mathcal{T}_1$ is vol-minimal, we get $\text{vol}(\mathcal{T}_2) = \text{vol}(\mathcal{T}_1)$. Iterating this argument yields an infinitely descending chain of finite-volume subsystems that are all strictly smaller in the $\leq_{\text{inv}}$ order. But this cannot exist, as the relation $<_{\text{inv}}$ is well-founded.

Now suppose that a vol-minimal witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ has infinite volume. Then, trivially, any witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ is vol-minimal since they all have infinite volume. In particular, every inv-minimal witness is also vol-minimal. □

Hence, to find a vol-minimal witness it suffices to compute *all* inv-minimal witnesses and to compare their volumes. We will again use the labeling functions $\Lambda^{\min}_{\text{inv}}$ and $\Lambda^{\max}_{\text{inv}}$ of $\mathcal{M}$ into the set of labels

$$L \;=\; \left\{\, \xi^{l}_{ij}(k) \;\middle|\; c_i, c_j \in \mathsf{Cl},\ l \in \text{Loc},\ k \in \{-2K, \dots, 2K\} \right\},$$

as defined in the previous section. Lemma 6.21 shows that inv-minimal witnesses correspond to label-minimal witnesses of $\mathcal{M}$ *with respect to the subset order*. With respect to this order, a subsystem $\mathcal{M}_{S_1}$ is smaller than $\mathcal{M}_{S_2}$ if $\Lambda^{\mathfrak{m}}_{\text{inv}}(\mathcal{M}_{S_1}) \subset \Lambda^{\mathfrak{m}}_{\text{inv}}(\mathcal{M}_{S_2})$ holds. This should be compared to the standard label ordering, which is defined using $|\Lambda^{\mathfrak{m}}_{\text{inv}}(\mathcal{M}_{S_1})| \leq |\Lambda^{\mathfrak{m}}_{\text{inv}}(\mathcal{M}_{S_2})|$.

As in Proposition 6.23, let $B = 4k + 1$. We have seen that there exist $O\big(2^{\log(B+1)\cdot|\text{Loc}|\cdot|\mathsf{Cl}|^2}\big)$ relevant subsets of L due to the monotonicity of labels (see Equation (6.5) and Proposition 6.23). To compute a vol-minimal witness, it is hence enough to enumerate this number of subsystems of $\mathcal{M}$ and compare their volumes. This leads to a single exponential algorithm, as stated in the following proposition. Here we let $\text{vol}(|\mathsf{Cl}|^2, \log(K))$ be the time required to compute the volume of a DBM over clocks Cl and with all integer entries in $\{-K, \dots, K\}$.

Proposition 6.25. *Let $\mathcal{T}$ be a PTA, $\sim$ be a PTAB on $\mathcal{T}$ which respects* target *and* exit*, has finite index and distinguishes regions, and let $\mathcal{M}$ be the quotient of $\mathcal{S}(\mathcal{T})$ by $\sim$. Furthermore, let K be an upper bound on the possible value of any clock in any location and let $B = 4K + 1$.*

Then for all $\mathfrak{m} \in \{\min, \max\}$ and $\lambda \in [0,1]$, a vol-minimal witness for $\mathbf{Pr}^{\mathfrak{m}}_{\mathcal{T}}(\Diamond \text{target}) \geq \lambda$ can be computed in time $O\big(2^{\log(B+1)\cdot|\text{Loc}|\cdot|\mathsf{Cl}|^2} \cdot \text{vol}(|\mathsf{Cl}|^2, \log(K)) \cdot \text{poly}(|\mathcal{M}|)\big)$, if one exists.

Proof. By Lemma 6.21, inv-minimal subsystems of $\mathcal{T}$ correspond to label-minimal subsystems of $\mathcal{M}$ with respect to the subset ordering of the labeling function $\Lambda^{\mathfrak{m}}_{\text{inv}}$. Under these labeling functions, there are $2^{\log(B+1)\cdot|\text{Loc}|\cdot|\mathsf{Cl}|^2}$ relevant subsets of labels to consider (see Proposition 6.23). An algorithm which achieves the claimed running time enumerates all of these subsets of labels, checks whether the induced MDP subsystem is a witness for the threshold property and, if

so, computes the volume of the induced PTA subsystem. After completing the enumeration, the PTA subsystem which achieved a minimal volume among the ones that were stored is returned. □

The algorithm that was sketched in the above proof relies on a complete enumeration of relevant label subsets. To circumvent this issue, one can formulate the problem as a *multi-objective* mixed-integer linear program. This is a mixed-integer linear program subject to a set of optimization objectives, rather than a single one. As a consequence we deal with a partial order of feasible solutions, and the goal is to compute the set of minimal (sometimes called non-dominated) solutions with respect to this partial order.

Recall that the objective function used to compute label-minimal witnessing subsystems for MDPs is of the form $\min \sum_{p \in L} \sigma(p)$, where $\sigma(p)$ is an integer variable representing the choice of whether label p should be included or not (see Definition 4.35). For $L_1, L_2 \subseteq L$, let $\mathcal{M}_1$ be the MDP subsystem of $\mathcal{M}$ which excludes exactly the states which are labeled by some label in $L \setminus L_1$, and $\mathcal{M}_2$ be defined analogously. Now for fixed $l \in \mathrm{Loc}$, $c_i, c_j \in \mathrm{Cl}$ we have

$$\{\xi^l_{ij}(b) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(\mathcal{M}_1) \mid b \in \{-2K, \dots, 2K\}\} \quad \subseteq \quad \{\xi^l_{ij}(b) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(\mathcal{M}_2) \mid b \in \{-2K, \dots, 2K\}\}$$

if and only if

$$|\{\xi^l_{ij}(b) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(\mathcal{M}_1) \mid b \in \{-2K, \dots, 2K\}\}| \quad \leq \quad |\{\xi^l_{ij}(b) \in \Lambda^{\mathfrak{m}}_{\mathrm{inv}}(\mathcal{M}_2) \mid b \in \{-2K, \dots, 2K\}\}|,$$

by the monotonicity of labels described in Equation (6.5).

Consider the *multi-objective* mixed-integer linear program one gets by replacing the objective function in the MILP for the label-minimal witness problem for $\mathcal{M}$ under labeling $\Lambda^{\mathfrak{m}}_{\mathrm{inv}}$ by the $|\mathrm{Loc}| \cdot |\mathrm{Cl}|^2$ objective functions

$$\min \sum_{-2K \leq b \leq 2K} \xi^l_{ij}(b), \qquad \text{for all } l \in \mathrm{Loc},\ c_i, c_j \in \mathrm{Cl}, \tag{6.6}$$

where we interpret $\xi^l_{ij}(b)$ as a binary integer variable corresponding to the label with the same name.

The solutions of the resulting program correspond to the inv-minimal witnesses of $\mathcal{T}$ by the above discussion and Lemma 6.21. Hence, to compute a vol-minimal PTA subsystem one can first solve this multi-objective MILP to receive the set of inv-minimal subsystems. Out of these, one can then pick one with minimal volume. With this approach one can avoid the exhaustive enumeration of label-subsets to compute the set of inv-minimal witnesses, by exploiting techniques for solving multi-objective MILPs as presented in [ÖK10, PO19].

The complexity of deciding the volume order

A necessary ingredient in the described algorithms for vol-minimal PTA subsystems are algorithms which compute the volume of a polytope. This problem generally requires exponential time in the number of dimensions [GK94]. However, as the location invariants of PTA have a very restricted form involving only linear inequalities with at most two clocks, one might hope that computing their volume is easier. We now show that this is not the case.

We recall that #P is the counting complexity class which includes the functions that can be expressed as the number of accepting runs of a polynomial time, nondeterministic Turing

machine (NTM) for a given input. Hardness for #P is typically defined using polynomial-time Turing reductions. A problem P_1 is reducible to P_2 under such reductions if one can solve P_1 using a polynomially time-bounded Turing machine with an oracle for P_2. The analogous decision class is PP, where $L \in \mathrm{PP}$ if there is a polynomial time NTM such that $x \in L$ if and only if the majority of runs of the NTM on x is accepting (see [AB09, Chapter 9] for more information on these complexity classes). Via a reduction from specific results on polytope volume computation, the following proposition shows that computing the volume of a DBM is #P-hard.

Proposition 6.26. *Computing* $\mathrm{vol}(\mathrm{Val}(M))$ *for a DBM* M *is* #P*-hard.*

Proof. The problem of counting the number of linear extensions of a partially ordered set is known to be #P-complete [BW91]. It turns out that this problem is polynomially interreducible with the problem of computing the volume of a so called *order-polytope* [GK94, Theorem 5.1.4]. Let $\sqsubset$ be a partial order on the set $\{1, \dots, n\}$. The order polytope $\mathcal{P}^{\sqsubset}$ is defined as:

$$\mathcal{P}^{\sqsubset} \;=\; \{\, x \in [0,1]^n \;\mid\; \text{if } i \sqsubset j \text{ then } x(i) \le x(j) \text{ for all } i, j \in \{1, \dots, n\} \,\}.$$

Such polytopes can be defined using DBMs over clocks $\mathsf{Cl} = \{c_0, \dots, c_n\}$ as follows:

$$M^{\sqsubset}_{ij} = \begin{cases} (1, \le) & \text{if } i \ge 1, j = 0, \\ (0, \le) & \text{if } i = 0, j \ge 0, \\ (0, \le) & \text{if } i \sqsubset j, \\ (1, \le) & \text{otherwise.} \end{cases}$$

The first two cases represent the constraint $0 \le v(c_i) \le 1$ for all clocks $c_i \in \mathsf{Cl}$, by defining appropriate upper and lower bounds on the difference to the zero clock c_0. The third case formalizes that $v(c_i) - v(c_j) \le 0$ should hold whenever $(i, j) \in I$. Given that $0 \le v(c_i) \le 1$ holds, the fourth condition does not impose any further restriction on the polytope. Then, $\mathcal{P}^{\sqsubset}$ equals $\mathrm{Val}(M^{\sqsubset})$ considered as a subset of $\mathbb{R}^{\mathsf{Cl} \setminus \{c_0\}} \cong \mathbb{R}^n$, and hence $\mathrm{vol}(\mathcal{P}^{\sqsubset}) = \mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$.

It follows that the volume computation problem for these special DBMs is #P-complete, and in general #P-hard. □

This observation can be used to show that comparing the volumes of two PTA subsystems (i.e., deciding the $\le_{\mathrm{vol}}$ order) is hard.

Theorem 6.27. *Given two subsystems* $\mathcal{T}_1, \mathcal{T}_2$ *in a PTA* $\mathcal{T}$*, deciding whether* $\mathcal{T}_1 \le_{\mathrm{vol}} \mathcal{T}_2$ *holds is PP-hard under polynomial-time Turing reductions.*

Proof. As in Proposition 6.26, we consider the DBMs $M^{\sqsubset}$ (henceforth called order-DBMs) over clocks $\mathsf{Cl} = \{c_0, \dots, c_n\}$, defined using the partial order $\sqsubset$ over $\{1, \dots, n\}$. The problem of computing $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$ for order-DBMs is #P-complete by the proof of Proposition 6.26. It follows that the threshold problem $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset})) \ge k$, for a given order-DBM $M^{\sqsubset}$ and $k \in \mathbb{Q}$, is PP-hard under polynomial-time Turing reductions. This is because using an oracle for this problem, one can compute $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$ bitwise using a binary search. Only polynomially many calls to the oracle are needed here as the length of the binary representation of $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$ is polynomial (this holds for order polytopes in general).

It remains to show that computing the volume-threshold problem for order-DBMs can be reduced to deciding whether $\mathcal{T}_1 \le_{\mathrm{vol}} \mathcal{T}_2$ holds given a PTA $\mathcal{T}$ and two subsystems $\mathcal{T}_1, \mathcal{T}_2$ of $\mathcal{T}$.

We first make the observation that the value $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$ corresponds to the number of linear extensions of $\sqsubset$ divided by $n!$. Now the proof idea is to construct (the invariant condition of) $\mathcal{T}_1$ by scaling $M^{\sqsubset}$, such that the resulting volume becomes $n!^n \cdot \mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$. On the other hand, $\mathcal{T}_2$ is constructed to have a volume $n!^n \cdot k$. The reason for scaling both volumes by $n!^n$ is that it is not obvious how to define a DBM whose volume is exactly k.

Consider the PTA $\mathcal{T}$ having two locations l_1, l_2 with invariants M_1 and M_2, respectively, defined as follows. The first invariant M_1 inherits all its entries from $M^{\sqsubset}$, apart from the upper bounds (i.e., entries $(M_1)_{i0}$ for all $1 \leq i \leq n$) which are set to $n!$. As $n! = O(2^{n \log n})$, we can express $n!$ in $\mathrm{poly}(n)$ bits. We have $\mathrm{vol}(\mathrm{Val}(M_1)) = n!^n \cdot \mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$, as the polytope that M_1 represents is essentially the order polytope $\mathcal{P}^{\sqsubset}$ but scaled to the n-dimensional cube with side length $n!$.

As $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset}))$ is a multiple of $1/n!$ we can assume that so is k (or we round up to the nearest rational with this property). We let M_2 be the DBM that describes a row of $k \cdot n!^n$ (which is an integer) 1-cubes in n dimensions. This is achieved by letting all variables have upper bound 1 apart from a single variable with upper bound $k \cdot n!^n$. As $k \cdot n!^n = O(k \cdot (2^{n \log n})^n) = O(k \cdot (2^{n^2 \cdot \log n}))$, this number is expressible with $O(\mathrm{poly}(n) + \log(k))$ many bits. We have $\mathrm{vol}(\mathrm{Val}(M_2)) = k \cdot n!^n$.

Now let $\mathcal{T}_1$ be the subsystem that includes only location l_1, and $\mathcal{T}_2$ be the subsystem that includes only location l_2. Then, we have $\mathrm{vol}(\mathrm{Val}(M^{\sqsubset})) \geq k$ if and only if $\mathcal{T}_2 \leq_{\mathrm{vol}} \mathcal{T}_1$. This completes the reduction of the threshold problem for the volume of order-DBMs to deciding $\leq_{\mathrm{vol}}$. □

It follows that if there existed a polynomial time algorithm to decide $\mathcal{T}_1 \leq_{\mathrm{vol}} \mathcal{T}_2$, then the polynomial hierarchy would collapse by Toda's Theorem [Tod91]. This should be contrasted with the relations $\leq_{\mathrm{loc}}$ and $\leq_{\mathrm{inv}}$. To decide $\mathcal{T}_1 \leq_{\mathrm{loc}} \mathcal{T}_2$ one just counts the locations of the two subsystems, and for $\mathcal{T}_1 \leq_{\mathrm{inv}} \mathcal{T}_2$ one checks the inclusion of locations and inspects the entries of the canonical DBMs associated to the invariants. Hence, both of these checks can be done in polynomial time in $\mathcal{T}_1$ and $\mathcal{T}_2$.

Chapter 7

Conclusion

This thesis has introduced a number of new techniques to explicate and certify properties in the context of probabilistic model checking. We have focused on constraints on the optimal reachability probabilities in Markov decision processes and probabilistic timed automata. Apart from describing new kinds of explications and algorithms to compute them, a goal of this thesis was to determine the precise complexity of the corresponding computational problems.

Farkas certificates can be used to certify model checking algorithms for probabilistic reachability constraints in MDPs. They are vectors satisfying certain systems of linear inequalities, derived from the classical linear-programming-based characterizations of optimal reachability probabilities in MDPs and Farkas lemma. We use Farkas lemma to transform the question of *unsatisfiability* of one system of linear inequalities to the question of *satisfiability* of another. Hence, solutions of the latter may serve as certificates for the unsatisfiability of the former. An important observation of the thesis is that this duality can be used to provide certificate conditions in terms of satisfiability of linear inequalities for all kinds of probabilistic reachability constraints. Validating Farkas certificates amounts to checking whether a candidate certificate is a solution of the corresponding inequalities and can hence be done in linear time.

Witnessing subsystems were introduced in [JÁK+11] as a means to explicate properties of the form $\Pr(\Diamond\,\text{target}) \geq \lambda$ in Markov chains, and later also for lower bounds on the maximal reachability probability in Markov decision processes [WJÁ+12]. We showed that the support (i.e., the set of indices with non-zero value) of a Farkas certificate induces a witnessing subsystem for the same property. This observation leads to novel algorithms for computing minimal witnessing subsystems, both exactly and heuristically. The quotient-sum heuristic aims to compute Farkas certificates with small support by solving a sequence of linear programs. In an experimental analysis we showed that this heuristic is competitive with known approaches in terms of computation time and the size of computed witnesses. It generally returns a good solution already after a few (usually two to three) iterations. Hence, the overhead of using the heuristic is not huge when compared with model checking, which requires solving a single linear program of the same size. All algorithms which are based on computing Farkas certificates are certifying, by construction. The Farkas certificate which is returned along with a witnessing subsystem (actually, the certificate *induces* the witness) provides an easily-verifiable proof that

the subsystem is an actual witness for the considered property.

Regarding the complexity of computing minimal witnessing subsystems, we have shown that the corresponding decision problem is NP-complete already for acyclic Markov chains. To find possibly tractable subclasses, we studied Markov chains with bounded tree width. Here, a negative result was proved: even for the class of Markov chains with bounded path width, the problem of computing minimal witnessing subsystems remains NP-hard. The proof exposes a new type of combinatorial hardness, which was not utilized in other NP-hardness proofs. On the other hand, we show that one can still hope for algorithms which exploit tree structure of an MDP to compute witnesses faster in practice. Such an algorithm was proposed and experiments show that it outperforms other approaches for systems with favorable structure.

Finally, we considered explications on the level of probabilistic timed automata (PTA), a well established model for probabilistic real-time systems. We proposed two notions of witnessing subsystems for lower-bounded reachability constraints for PTA, one for maximal and one for minimal reachability probabilities. Intuitively, one gets a subsystem by removing locations and strengthening location invariants and transition guards. For minimal probabilities, additional care has to be taken to ensure that the minimal probability cannot increase in a subsystem. Small subsystems are more informative, and we consider three different notions of size for subsystems of PTA. While the first simply counts the number of locations included in the subsystems, the other two are designed to take timing aspects into account.

Future work

Richer properties in Markov chains. Our focus in this thesis was on reachability probabilities. This is a natural choice, as they are an important building block for many probabilistic model checking problems. A standard approach to compute the probability of an ω-regular property in a Markov chain $\mathcal{M}$ is to compute a deterministic Rabin automaton (DRA) $\mathcal{A}$ for the property, construct the product of $\mathcal{M}$ and $\mathcal{A}$ (which is again a Markov chain), and compute the probability of reaching a bottom strongly connected component (BSCC) B in the product which satisfies the Rabin property. This means that for one of the Rabin pairs (L, K) we have $B \cap L = \emptyset$ and $B \cap K \neq \emptyset$. In this way, many problems considering ω-regular properties in Markov chains can be reduced to the case of reachability in a product Markov chain (which may be much larger than the original system). In particular, Farkas certificates for the reachability property in the product serve as certificates for the ω-regular property in the original system.

Witnessing subsystems with respect to ω-regular properties were considered in [WJÁ+14, Jan15]. The notion of a witnessing subsystem is essentially the same as for reachability[1]. To compute them for a Markov chain, one can consider the product of $\mathcal{M}$ and a DRA $\mathcal{A}$ for the property, as sketched above. Now, one labels the states in the product by their first component (i.e., the corresponding state of $\mathcal{M}$) and collapses the accepting BSCCs, which form the target set. The collapsed BSCCs are labeled by all states of the Markov chain included in the BSCC. Now we have a labeled Markov chain in reachability form, and the minimal witnessing subsystems of $\mathcal{M}$ w.r.t. the ω-regular property correspond to the label-minimal witnessing subsystems for a reachability property in the product. This reduction was described in [WJÁ+14]. With this idea, the algorithms and heuristics presented in this thesis can be used to compute minimal

[1] It is more convenient in this setting to use substochastic matrices and make edges "disappear" in a subsystem, rather than redirecting them to a dedicated state "exit". This is because there is no such canonical rejecting state if we consider arbitrary ω-regular properties.

witnesses in Markov chains for threshold constraints on the probability of satisfying ω-regular properties.

If the property is described using an LTL formula, then the above approach is not entirely satisfactory. This is because it requires double-exponential space, as any transformation of LTL formulas into DRA produces double-exponentially larger automata in the worst case. However, it is clear that minimal witnessing subsystems can be computed in exponential time by enumerating all subsystems of the Markov chain and checking whether they satisfy the probability bound on the LTL formula. This check can be done in exponential time [CY95]. Of course, such an exhaustive enumeration of subsystems is not feasible in practice. A promising direction to overcome this complexity gap is to apply methods for LTL model checking of Markov chains which use *unambiguous* Büchi automata [BKK+16]. With these techniques, the probability induced by an LTL formula can be computed using an (only) exponentially larger system of linear equations. It would be interesting to see whether and how our techniques can be extended to compute small witnessing subsystems and Farkas certificates on the basis of this system of linear equations.

Richer properties in Markov decision processes. For Markov decision processes, the situation is a bit different. The product construction works in an analogous way and again reduces the ω-regular case to a Rabin condition. A set of accepting states with respect to the maximal probabilities of satisfying the Rabin condition can also be computed (see [BK08, Theorem 10.125]). However, by collapsing these states we may lose information which is required to compute minimal witnessing subsystems. This is because a scheduler may only need to visit *some* of those states to achieve a desired probability. Observe that the difference to Markov chains is that in a Markov chain, partly including a BSCC in a subsystem is never useful. This is because for the subsystem this means that a rejecting state will be visited with probability one inside the former BSCC. However, in MDPs, partly including maximal end components may well be useful in a subsystem. A MILP-formulation which computes minimal witnesses for lower-bounded maximal probability constraints w.r.t. Rabin conditions in MDPs is presented in [WJÁ+14]. To overcome the issues sketched above, this MILP includes carefully crafted constraints to ensure that the schedulers induced by its solutions indeed realize accepting end components for the Rabin property. These constraints are costly, however, as they require $K \cdot N$ binary variables, where N is the number of states in the product and K is the number of Rabin pairs.

A similar issue arose when we considered invariance properties in Section 4.4. Here, one could not simply collapse proper end components in the case of maximal probabilities, because one would have potentially lost subsystems which may have been minimal witnesses. We solved this issue by using linear equation systems whose solutions induce proper end components (see also Lemma 3.8). For Rabin conditions, one would have to additionally guarantee that the induced proper end components satisfy the Rabin condition. Considering how our techniques can be extended to handle richer conditions such as Rabin or Streett is definitely worth exploring, with the goal of enabling better algorithms and heuristics for the computation of minimal witnessing subsystems with respect to ω-regular properties in MDPs.

That being said, the above issues concern only the question of computing small and minimal witnessing subsystems. Farkas certificates for ω-regular properties in MDPs can be defined via the reduction to reachability properties.

It may also be worth to reconsider the notion of witness for ω-regular properties. The

fact that the states in the product construction correspond to memory locations of the schedulers in the original system could be taken into account to possibly provide more informative explications.

Probabilistic computation tree logic. Another pathway for further research is to consider richer subclasses of PCTL properties beyond probabilistic reachability constraints. The standard approach to PCTL model checking recursively computes accepting state sets for all subformulas. Having computed these sets, one can compute the accepting state set of the main formula using methods for reachability probabilities. However, removing states to form a subsystem may change the acceptance status of subformulas in all states in ways that are difficult to predict. Witnessing subsystems for a safety fragment of PCTL were considered in [CV10]. Their algorithm for computing witnesses is based on removing states iteratively, until removing any further state would lead to the violation of the property. The approach is completely unguided, however, in the sense that states are removed in an arbitrary order. As probabilistic reachability constraints form the basis of PCTL, we believe that our work can be a good starting point to design exact algorithms and heuristics for the computation of minimal witnessing subsystems for such properties.

Other properties. And then, there are all the other kinds of properties. By considering the expected total reward, we have only scratched the surface on the quantitative and weighted properties which are important for MDPs. One can, and should, consider explications and certificates for (threshold constraints on) the expected mean-payoff, cost-utility ratios, energy properties, conditional expectations, to name a few. All of them will require new methods, but we hope that some of the ideas presented in this thesis can serve as starting points and inspiration.

Richer systems. It is also interesting to consider witnesses and certificates in classes of systems which go beyond Markov decision processes. A natural first candidate to address are stochastic two player reachability games (also called simple stochastic games). They resemble standard two player graph games, with the exception that the players now choose probability distributions over states (similarly as in MDPs). The objective of one player is to maximize the probability of reaching the target, while the other player tries to minimize it. The threshold problem of these games is in NP $\cap$ co-NP and not known to be solvable in polynomial time. In particular, (non-stochastic) parity games reduce to it [CF11]. Hence, we cannot hope for certificates which are solutions of small systems of linear inequalities (as this would imply a polynomial time algorithm for the problem). However, the fact that we have identified Farkas certificates for two special cases – namely, if all states belong to the minimizing or the maximizing player, respectively – makes us believe that our work is an interesting starting point to look for certificates in the general case. The notion of subsystems which is used for MDPs also makes sense here (although plausible alternatives certainly exist). If a state is removed, which means that we assign to it value zero, then the value of all other states does not increase[1]. Hence, a subsystem which satisfies a lower bound on the value in any given state, witnesses that this lower bound holds in the original system.

Another direction to explore are probabilistic hybrid automata, which generalize probabilistic timed automata. Here, one quickly runs into the issue that the model checking problem is

[1]By value we mean the reachability probability under optimal strategies for both players.

undecidable already with minor extensions beyond the setting of timed automata [HKPV98]. Still, decidable classes do exist, and first steps to generalize our results for probabilistic timed automata have been made [Hen21]. Also, incomplete model checking procedures for large classes of hybrid automata exist, and whenever such a procedure yields a conclusive answer, it is useful to provide an explication for it. The work on counterexamples for hybrid automata presented in [NÁCC14] goes in this direction. In this setting, the question of what constitutes a useful explication is still largely unexplored, and will be an important one to pursue.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[ÁBD+14] Erika Ábrahám, Bernd Becker, Christian Dehnert, Nils Jansen, Joost-Pieter Katoen, and Ralf Wimmer. Counterexample Generation for Discrete-Time Markov Models: An Introductory Survey. In Marco Bernardo, Ferruccio Damiani, Reiner Hähnle, Einar Broch Johnsen, and Ina Schaefer, editors, *Formal Methods for Executable Software Models: 14th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, (SFM)*, Lecture Notes in Computer Science, pages 65–121. Springer International Publishing, Cham, 2014. doi:10.1007/978-3-319-07317-0_3.

[ACD93] Rajeev Alur, Costas Courcoubetis, and David L. Dill. Model-Checking in Dense Real-Time. *Information and Computation*, 104(1):2–34, May 1993. doi:10.1006/inco.1993.1024.

[ACDE07] David L. Applegate, William Cook, Sanjeeb Dash, and Daniel G. Espinoza. Exact solutions to linear programming problems. *Operations Research Letters*, 35(6):693–699, Nov. 2007. doi:10.1016/j.orl.2006.12.010.

[ACG+20] Ali Asadi, Krishnendu Chatterjee, Amir Kafshdar Goharshady, Kiarash Mohammadi, and Andreas Pavlogiannis. Faster Algorithms for Quantitative Analysis of MCs and MDPs with Small Treewidth. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis - 18th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 253–270, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-59152-6_14.

[AČJ+21] Roman Andriushchenko, Milan Češka, Sebastian Junges, Joost-Pieter Katoen, and Šimon Stupinský. PAYNT: A Tool for Inductive Synthesis of Probabilistic Programs. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification - 33rd International Conference (CAV)*, Lecture Notes in Computer Science, pages 856–869, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-81685-8_40.

[AČJK21] Roman Andriushchenko, Milan Češka, Sebastian Junges, and Joost-Pieter Katoen. Inductive Synthesis for Probabilistic Programs Reaches New Horizons. In Jan Friso

Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 191–209, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-72016-2_11.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, Apr. 1994. doi:10.1016/0304-3975(94)90010-8.

[ADvR09] Miguel E. Andrés, Pedro D'Argenio, and Peter van Rossum. Significant Diagnostic Counterexamples in Probabilistic Model Checking. In Hana Chockler and Alan J. Hu, editors, *Hardware and Software: Verification and Testing*, Lecture Notes in Computer Science, pages 129–148, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-01702-5_15.

[AF92] David Avis and Komei Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(3):295–313, Sept. 1992. doi:10.1007/BF02293050.

[AF96] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1):21–46, Mar. 1996. doi:10.1016/0166-218X(95)00026-N.

[AFG+09] Husain Aljazzar, Manuel Fischer, Lars Grunske, Matthias Kuntz, Florian Leitner-Fischer, and Stefan Leue. Safety Analysis of an Airbag System Using Probabilistic FMEA and Probabilistic Counterexamples. In *QEST 2009, Sixth International Conference on the Quantitative Evaluation of Systems*, pages 299–308, Sept. 2009. doi:10.1109/QEST.2009.8.

[AH90] James Aspnes and Maurice Herlihy. Fast randomized consensus using shared memory. *Journal of Algorithms*, 11(3):441–461, Sept. 1990. doi:10.1016/0196-6774(90)90021-6.

[AH99] Rajeev Alur and Thomas A. Henzinger. Reactive Modules. *Formal Methods in System Design*, 15(1):7–48, July 1999. doi:10.1023/A:1008739929481.

[AHL05] Husain Aljazzar, Holger Hermanns, and Stefan Leue. Counterexamples for Timed Probabilistic Reachability. In Paul Pettersson and Wang Yi, editors, *Formal Modeling and Analysis of Timed Systems, Third International Conference (FORMATS)*, Lecture Notes in Computer Science, pages 177–195, Berlin, Heidelberg, 2005. Springer. doi:10.1007/11603009_15.

[ÁJW+10] Erika Ábrahám, Nils Jansen, Ralf Wimmer, Joost-Pieter Katoen, and Bernd Becker. DTMC Model Checking by SCC Reduction. In *QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems*, pages 37–46, 2010. doi:10.1109/QEST.2010.13.

[AK98] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1):237–260, Dec. 1998. doi:10.1016/S0304-3975(97)00115-1.

[AL06] Husain Aljazzar and Stefan Leue. Extended Directed Search for Probabilistic Timed Reachability. In Eugene Asarin and Patricia Bouyer, editors, *Formal Modeling and Analysis of Timed Systems, 4th International Conference (FORMATS)*, Lecture Notes in Computer Science, pages 33–51, Berlin, Heidelberg, 2006. Springer. doi:10.1007/11867340_4.

[AL08] Husain Aljazzar and Stefan Leue. Debugging of Dependability Models Using Interactive Visualization of Counterexamples. In *QEST 2008, Fifth International Conference on Quantitative Evaluation of Systems*, pages 189–198, Sept. 2008. doi:10.1109/QEST.2008.40.

[AL09] Husain Aljazzar and Stefan Leue. Generation of Counterexamples for Model Checking of Markov Decision Processes. In *QEST 2009, Sixth International Conference on the Quantitative Evaluation of Systems*, pages 197–206, Sept. 2009. doi:10.1109/QEST.2009.10.

[AL10] Husain Aljazzar and Stefan Leue. Directed Explicit State-Space Search in the Generation of Counterexamples for Stochastic Model Checking. *IEEE Transactions on Software Engineering*, 36(1):37–60, Jan. 2010. doi:10.1109/TSE.2009.57.

[ALLS11] Husain Aljazzar, Florian Leitner-Fischer, Stefan Leue, and Dimitar Simeonov. DiPro - A Tool for Probabilistic Counterexample Generation. In Alex Groce and Madanlal Musuvathi, editors, *Model Checking Software*, Lecture Notes in Computer Science, pages 183–187, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-22306-8_13.

[AMP09] Alessandro Armando, Jacopo Mantovani, and Lorenzo Platania. Bounded model checking of software using SMT solvers instead of SAT solvers. *International Journal on Software Tools for Technology Transfer*, 11(1):69–83, Feb. 2009. doi:10.1007/s10009-008-0091-0.

[And01] Erling D. Andersen. Certificates of Primal or Dual Infeasibility in Linear Programming. *Computational Optimization and Applications*, 20(2):171–183, Nov. 2001. doi:10.1023/A:1011259103627.

[BBF+16] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications*, 65(3):545–566, Dec. 2016. doi:10.1007/s10589-016-9847-8.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. doi:10.1007/978-3-662-07964-5.

[BCC+15] Tomáš Brázdil, Krishnendu Chatterjee, Martin Chmelík, Andreas Fellner, and Jan Křetínský. Counterexample Explanation by Learning Small Strategies in Markov Decision Processes. In Daniel Kroening and Corina S. Păsăreanu, editors, *Computer Aided Verification - 27th International Conference (CAV)*, Lecture Notes in

Computer Science, pages 158–177, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-21690-4_10.

[BCCZ99] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. Symbolic Model Checking without BDDs. In W. Rance Cleaveland, editor, *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 193–207, Berlin, Heidelberg, 1999. Springer. doi:10.1007/3-540-49059-0_14.

[BCM+92] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation*, 98(2):142–170, 1992. doi:10.1016/0890-5401(92)90017-A.

[BdeA95] Andrea Bianco and Luca de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. S. Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science, 15th Conference (FSTTCS)*, Lecture Notes in Computer Science, pages 499–513, Berlin, Heidelberg, 1995. Springer. doi:10.1007/3-540-60692-0_70.

[BDH96] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, Dec. 1996. doi:10.1145/235815.235821.

[Ben08] Mordechai Ben-Ari. *Principles of the Spin Model Checker*. Springer-Verlag, London, 2008. doi:10.1007/978-1-84628-770-1.

[BHHK03] Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, June 2003. doi:10.1109/TSE.2003.1205180.

[BHJM07] Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. The software model checker Blast. *International Journal on Software Tools for Technology Transfer*, 9(5):505–525, Oct. 2007. doi:10.1007/s10009-007-0044-z.

[BHZ08] Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1):3–21, June 2008. doi:10.1016/j.scico.2007.08.001.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[BKK+16] Christel Baier, Stefan Kiefer, Joachim Klein, Sascha Klüppelholz, David Müller, and James Worrell. Markov Chains and Unambiguous Büchi Automata. In Swarat Chaudhuri and Azadeh Farzan, editors, *Computer Aided Verification - 28th International Conference, (CAV)*, Lecture Notes in Computer Science, pages 23–42, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-41528-4_2.

[BKL+17] Christel Baier, Joachim Klein, Linda Leuschner, David Parker, and Sascha Wunderlich. Ensuring the Reliability of Your Model Checker: Interval Iteration for Markov Decision Processes. In Rupak Majumdar and Viktor Kunčak, editors, *Computer Aided Verification - 29th International Conference, (CAV)*, Lecture Notes in Computer Science, pages 160–180, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-63387-9_8.

[BLR11] Thomas Ball, Vladimir Levin, and Sriram K. Rajamani. A decade of software model checking with SLAM. *Communications of the ACM*, 54(7):68–76, July 2011. doi:10.1145/1965724.1965743.

[BLTW15] Pierre Bonami, Andrea Lodi, Andrea Tramontani, and Sven Wiese. On mathematical programming with indicator constraints. *Mathematical Programming*, 151(1):191–223, June 2015. doi:10.1007/s10107-015-0891-4.

[BMS+17] Anna Bernasconi, Claudio Menghi, Paola Spoletini, Lenore D. Zuck, and Carlo Ghezzi. From Model Checking to a Temporal Proof for Partial Models. In Alessandro Cimatti and Marjan Sirjani, editors, *Software Engineering and Formal Methods - 15th International Conference (SEFM)*, Lecture Notes in Computer Science, pages 54–69, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-66197-1_4.

[Bod97] Hans L. Bodlaender. Treewidth: Algorithmic techniques and results. In Igor Prívara and Peter Ružička, editors, *Mathematical Foundations of Computer Science, 22nd International Symposium, (MFCS)*, Lecture Notes in Computer Science, pages 19–36, Berlin, Heidelberg, 1997. Springer. doi:10.1007/BFb0029946.

[BT91] Dimitri P. Bertsekas and John N. Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16(3):580–595, Aug. 1991. doi:10.1287/moor.16.3.580.

[BV14] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014. doi:10.1017/CBO9780511804441.

[BW91] Graham Brightwell and Peter Winkler. Counting linear extensions. *Order*, 8(3):225–242, Sept. 1991. doi:10.1007/BF00383444.

[BWB+11] Bettina Braitling, Ralf Wimmer, Bernd Becker, Nils Jansen, and Erika Ábrahám. Counterexample Generation for Markov Chains Using SMT-Based Bounded Model Checking. In Roberto Bruni and Juergen Dingel, editors, *Formal Techniques for Distributed Systems*, Lecture Notes in Computer Science, pages 75–89, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-21461-5_5.

[BY04] Johan Bengtsson and Wang Yi. Timed Automata: Semantics, Algorithms and Tools. In Jörg Desel, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, Lecture Notes in Computer Science, pages 87–124. Springer, Berlin, Heidelberg, 2004. doi:10.1007/978-3-540-27755-2_3.

[CB06] Frank Ciesinski and Christel Baier. LiQuor: A tool for Qualitative and Quantitative Linear Time analysis of Reactive Systems. In *QEST 2006, Third International Conference on the Quantitative Evaluation of Systems*, pages 131–132. IEEE Computer Society, 2006. doi:10.1109/QEST.2006.25.

[CBGK08] Frank Ciesinski, Christel Baier, Marcus Größer, and Joachim Klein. Reduction Techniques for Model Checking Markov Decision Processes. In *QEST 2008, Fifth International Conference on Quantitative Evaluation of Systems*, pages 45–54, Sept. 2008. doi:10.1109/QEST.2008.45.

[CDR92] John Canny, Bruce Donald, and Eugene K. Ressler. A rational rotation method for robust geometric algorithms. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, (SCG), pages 251–260, New York, NY, USA, July 1992. Association for Computing Machinery. doi:10.1145/142675.142726.

[CES09] Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis. Model checking: Algorithmic verification and debugging. *Communincations of the ACM*, 52(11):74–84, 2009. doi:10.1145/1592761.1592781.

[CF11] Krishnendu Chatterjee and Nathanaël Fijalkow. A reduction from parity games to simple stochastic games. *Second International Symposium on Games, Automata, Logics, and Formal Verification (GandALF)*, 54:74–86, June 2011, 1106.1232. doi:10.4204/EPTCS.54.6.

[CGJ+03] Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM*, 50(5):752–794, Sept. 2003. doi:10.1145/876638.876643.

[CGL94] Edmund M. Clarke, Orna Grumberg, and David E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, Sept. 1994. doi:10.1145/186025.186051.

[Cha93] Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(4):377–409, Dec. 1993. doi:10.1007/BF02573985.

[Cha96] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, Apr. 1996. doi:10.1007/BF02712873.

[ČHJK19] Milan Češka, Christian Hensel, Sebastian Junges, and Joost-Pieter Katoen. Counterexample-Driven Synthesis for Probabilistic Program Sketches. In Maurice H. ter Beek, Annabelle McIver, and José N. Oliveira, editors, *Formal Methods – The Next 30 Years*, Lecture Notes in Computer Science, pages 101–120, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-30942-8_8.

[CHJM05] Krishnendu Chatterjee, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. Counterexample-guided Planning. In *Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*, pages 104–111. AUAI Press, 2005.

[CHK08] Taolue Chen, Tingting Han, and Joost-Pieter Katoen. Time-Abstracting Bisimulation for Probabilistic Timed Automata. In *2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 177–184, June 2008. doi:10.1109/TASE.2008.29.

[CHVB18] Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors. *Handbook of Model Checking*. Springer, 2018. doi:10.1007/978-3-319-10575-8.

[CIP15] Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Andreas Pavlogiannis. Faster Algorithms for Quantitative Verification in Constant Treewidth Graphs. In Daniel Kroening and Corina S. Păsăreanu, editors, *Computer Aided Verification - 27th International Conference (CAV)*, Lecture Notes in Computer Science, pages 140–157, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-21690-4_9.

[CJLV02] Edmund M. Clarke, Somesh Jha, Yuan Lu, and Helmut Veith. Tree-like counterexamples in model checking. In *17th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 19–29, July 2002. doi:10.1109/LICS.2002.1029814.

[CŁ13] Krishnendu Chatterjee and Jakub Łącki. Faster Algorithms for Markov Decision Processes with Low Treewidth. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference (CAV)*, Lecture Notes in Computer Science, pages 543–558, Berlin, Heidelberg, 2013. Springer. doi:10.1007/978-3-642-39799-8_36.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.

[CNŽ17] Krishnendu Chatterjee, Petr Novotný, and Đorđe Žikelić. Stochastic invariants for probabilistic termination. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, (POPL), pages 145–160, New York, NY, USA, Jan. 2017. Association for Computing Machinery. doi:10.1145/3009837.3009873.

[Cou90] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, Mar. 1990. doi:10.1016/0890-5401(90)90043-H.

[CSS03] Michael A. Colón, Sriram Sankaranarayanan, and Henny B. Sipma. Linear Invariant Generation Using Non-linear Constraint Solving. In Warren A. Hunt and Fabio Somenzi, editors, *Computer Aided Verification, 15th International Conference (CAV)*, Lecture Notes in Computer Science, pages 420–432, Berlin, Heidelberg, 2003. Springer. doi:10.1007/978-3-540-45069-6_39.

[CV03] Edmund Clarke and Helmut Veith. Counterexamples Revisited: Principles, Algorithms, Applications. In Nachum Dershowitz, editor, *Verification: Theory and Practice. Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, Lecture Notes in Computer Science, pages 208–224. Springer, Berlin, Heidelberg, 2003. doi:10.1007/978-3-540-39910-0_9.

[CV10] Rohit Chadha and Mahesh Viswanathan. A counterexample-guided abstraction-refinement framework for Markov decision processes. *ACM Transactions on Computational Logic*, 12(1):1:1–1:49, Nov. 2010. doi:10.1145/1838552.1838553.

[CY90] Costas Courcoubetis and Mihalis Yannakakis. Markov decision processes and regular events (extended abstract). In Michael S. Paterson, editor, *Automata, Languages and Programming, 17th International Colloquium (ICALP)*, Lecture Notes in Computer Science, pages 336–349, Berlin, Heidelberg, 1990. Springer. doi:10.1007/BFb0032043.

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, July 1995. doi:10.1145/210332.210339.

[deA97] Luca de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford university, 1997.

[deA99] Luca de Alfaro. Computing Minimum and Maximum Reachability Times in Probabilistic Systems. In Jos C. M. Baeten and Sjouke Mauw, editors, *Concurrency Theory, 10th International Conference (CONCUR)*, Lecture Notes in Computer Science, pages 66–81, Berlin, Heidelberg, 1999. Springer. doi:10.1007/3-540-48320-9_7.

[deAKN+00] Luca de Alfaro, Marta Kwiatkowska, Gethin Norman, David Parker, and Roberto Segala. Symbolic Model Checking of Probabilistic Processes Using MTBDDs and the Kronecker Representation. In Susanne Graf and Michael Schwartzbach, editors, *Tools and Algorithms for Construction and Analysis of Systems, 6th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 395–410, Berlin, Heidelberg, 2000. Springer. doi:10.1007/3-540-46419-0_27.

[DHK08] Berteun Damman, Tingting Han, and Joost-Pieter Katoen. Regular Expressions for PCTL Counterexamples. In *QEST 2008, Fifth International Conference on Quantitative Evaluation of Systems*, pages 179–188, Sept. 2008. doi:10.1109/QEST.2008.11.

[Dil90] David L. Dill. Timing assumptions and verification of finite-state concurrent systems. In Joseph Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, Lecture Notes in Computer Science, pages 197–212, Berlin, Heidelberg, 1990. Springer. doi:10.1007/3-540-52148-8_17.

[DJ14] N. Dinh and V. Jeyakumar. Farkas' lemma: Three decades of generalizations for mathematical optimization. *TOP*, 22(1):1–22, Apr. 2014. doi:10.1007/s11750-014-0319-y.

[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, *Computer Aided Verification - 29th International Conference (CAV)*, Lecture Notes in Computer Science, pages 592–600, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-63390-9_31.

[DJW+14] Christian Dehnert, Nils Jansen, Ralf Wimmer, Erika Ábrahám, and Joost-Pieter Katoen. Fast Debugging of PRISM Models. In Franck Cassez and Jean-François

Raskin, editors, *Automated Technology for Verification and Analysis - 12th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 146–162, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-11936-6_11.

[DKL07] Henning Dierks, Sebastian Kupferschmid, and Kim G. Larsen. Automatic Abstraction Refinement for Timed Automata. In Jean-François Raskin and P. S. Thiagarajan, editors, *Formal Modeling and Analysis of Timed Systems, 5th International Conference (FORMATS)*, Lecture Notes in Computer Science, pages 114–129, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-75454-1_10.

[DO96] Guoli Ding and Bogdan Oporowski. On tree-partitions of graphs. *Discrete Mathematics*, 149(1):45–58, Feb. 1996. doi:10.1016/0012-365X(94)00337-I.

[EC82] E. Allen Emerson and Edmund M. Clarke. Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons. *Science of Computer Programming*, 2(3):241–266, 1982. doi:10.1016/0167-6423(83)90017-5.

[Ede86] Anders Edenbrandt. Quotient tree partitioning of undirected graphs. *BIT Numerical Mathematics*, 26(2):148–155, June 1986. doi:10.1007/BF01933740.

[Far02] Julius Farkas. Theorie der einfachen Ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1902(124):1–27, Jan. 1902. doi:10.1515/crll.1902.124.1.

[FJB20] Florian Funke, Simon Jantsch, and Christel Baier. Farkas Certificates and Minimal Witnesses for Probabilistic Reachability Constraints. In Armin Biere and David Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 324–345, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-45190-5_18.

[FKNP11] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, and David Parker. Automated Verification Techniques for Probabilistic Systems. In Marco Bernardo and Valérie Issarny, editors, *Formal Methods for Eternal Networked Software Systems: 11th International School on Formal Methods for the Design of Computer, Communication and Software Systems (SFM)*, Lecture Notes in Computer Science, pages 53–113. Springer, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-21455-4_3.

[FM06] Zhaohui Fu and Sharad Malik. On Solving the Partial MAX-SAT Problem. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing, 9th International Conference (SAT)*, volume 4121 of *Lecture Notes in Computer Science*, pages 252–265. Springer, 2006. doi:10.1007/11814948_25.

[FMY97] Masahiro Fujita, Patrick C. McGeer, and Jerry Chih-Yuan Yang. Multi-Terminal Binary Decision Diagrams: An Efficient Data Structure for Matrix Representation. *Formal Methods in System Design*, 10(2):149–169, Apr. 1997. doi:10.1023/A:1008647823331.

[FY03] Uriel Feige and Orly Yahalom. On the complexity of finding balanced oneway cuts. *Information Processing Letters*, 87(1):1–5, July 2003. doi:10.1016/S0020-0190(03)00251-5.

[GHK+16] Robert Ganian, Petr Hliněný, Joachim Kneis, Daniel Meister, Jan Obdržálek, Peter Rossmanith, and Somnath Sikdar. Are there any good digraph width measures? *Journal of Combinatorial Theory, Series B*, 116:250–286, Jan. 2016. doi:10.1016/j.jctb.2015.09.001.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[GK94] Peter Gritzmann and Victor Klee. On the Complexity of Some Basic Problems in Computational Convexity. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, NATO ASI Series, pages 373–466. Springer Netherlands, Dordrecht, 1994. doi:10.1007/978-94-011-0924-6_17.

[GKS05] Patrice Godefroid, Nils Klarlund, and Koushik Sen. DART: Directed automated random testing. In Vivek Sarkar and Mary W. Hall, editors, *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 213–223. ACM, 2005. doi:10.1145/1065010.1065036.

[GL81] Peter Gács and Laszlo Lovász. Khachiyan's algorithm for linear programming. In H. König, B. Korte, and K. Ritter, editors, *Mathematical Programming at Oberwolfach*, Mathematical Programming Studies, pages 61–68. Springer, Berlin, Heidelberg, 1981. doi:10.1007/BFb0120921.

[GMT02] Jens Gustedt, Ole A. Mæhle, and Jan Arne Telle. The Treewidth of Java Programs. In David M. Mount and Clifford Stein, editors, *Algorithm Engineering and Experiments*, Lecture Notes in Computer Science, pages 86–97, Berlin, Heidelberg, 2002. Springer. doi:10.1007/3-540-45643-0_7.

[Gra72] Ronald L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, June 1972. doi:10.1016/0020-0190(72)90045-2.

[GRT18] Alberto Griggio, Marco Roveri, and Stefano Tonetta. Certifying Proofs for LTL Model Checking. In *Proceedings of the 18th Conference on Formal Methods in Computer Aided Design (FMCAD)*, pages 1–9, Oct. 2018. doi:10.23919/FMCAD.2018.8603022.

[Gur22] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2022. URL `https://www.gurobi.com`.

[Han91] Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems*. PhD thesis, University Uppsala, Sweden, 1991.

[HC11] Monika Henzinger and Krishnendu Chatterjee. Faster and Dynamic Algorithms For Maximal End-Component Decomposition And Related Graph Problems In Probabilistic Verification. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, San Francisco, Jan. 2011. doi:10.1137/1.9781611973082.101.

[Hen96] Thomas A. Henzinger. The theory of hybrid automata. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 278–292, July 1996. doi:10.1109/LICS.1996.561342.

[Hen21] Tom René Hennig. Witnessing Subsystems and Farkas Certificates for Probabilistic Rectangular Automata. Master's thesis, Technische Universität Dresden, 2021.

[HK07a] Tingting Han and Joost-Pieter Katoen. Counterexamples in Probabilistic Model Checking. In Orna Grumberg and Michael Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 72–86, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-71209-1_8.

[HK07b] Tingting Han and Joost-Pieter Katoen. Providing Evidence of Likely Being on Time: Counterexample Generation for CTMC Model Checking. In Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura, editors, *Automated Technology for Verification and Analysis, 5th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 331–346, Berlin, Heidelberg, 2007. Springer. doi:10.1007/978-3-540-75596-8_24.

[HK20] Arnd Hartmanns and Benjamin Lucien Kaminski. Optimistic Value Iteration. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification - 32nd International Conference (CAV)*, Lecture Notes in Computer Science, pages 488–511, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-53291-8_26.

[HKD09] Tingting Han, Joost-Pieter Katoen, and Berteun Damman. Counterexample Generation in Probabilistic Model Checking. *IEEE Transactions on Software Engineering*, 35(2):241–257, Mar. 2009. doi:10.1109/TSE.2009.5.

[HKPV98] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's Decidable about Hybrid Automata? *Journal of Computer and System Sciences*, 57(1):94–124, Aug. 1998. doi:10.1006/jcss.1998.1581.

[HLS+14] Ernst Moritz Hahn, Yi Li, Sven Schewe, Andrea Turrini, and Lijun Zhang. iscasMc: A Web-Based Probabilistic Model Checker. In Cliff Jones, Pekka Pihlajasaari, and Jun Sun, editors, *FM 2014: Formal Methods - 19th International Symposium*, Lecture Notes in Computer Science, pages 312–317, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-06410-9_22.

[HM14] Serge Haddad and Benjamin Monmege. Reachability in MDPs: Refining Convergence of Value Iteration. In Joël Ouaknine, Igor Potapov, and James Worrell, editors, *Reachability Problems - 8th International Workshop (RP)*, Lecture Notes in Computer Science, pages 125–137, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-11439-2_10.

[HSV93] Leen Helmink, M. P. A. Sellink, and Frits W. Vaandrager. Proof-Checking a Data Link Protocol. In Henk Barendregt and Tobias Nipkow, editors, *Types for Proofs and Programs, International Workshop (TYPES)*, volume 806 of *Lecture Notes in Computer Science*, pages 127–165. Springer, 1993. doi:10.1007/3-540-58085-9_75.

[HWZ08] Holger Hermanns, Björn Wachter, and Lijun Zhang. Probabilistic CEGAR. In Aarti Gupta and Sharad Malik, editors, *Computer Aided Verification, 20th International Conference (CAV)*, Lecture Notes in Computer Science, pages 162–175, Berlin, Heidelberg, 2008. Springer. doi:10.1007/978-3-540-70545-1_16.

[HZH+10] Fei He, He Zhu, William N.N. Hung, Xiaoyu Song, and Ming Gu. Compositional Abstraction Refinement for Timed Systems. In *4th IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 168–176, Aug. 2010. doi:10.1109/TASE.2010.27.

[IR90] Alon Itai and Michael Rodeh. Symmetry breaking in distributed networks. *Information and Computation*, 88(1):60–87, 1990. doi:10.1016/0890-5401(90)90004-2.

[JÁK+11] Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, and Bernd Becker. Hierarchical Counterexamples for Discrete-Time Markov Chains. In Tevfik Bultan and Pao-Ann Hsiung, editors, *Automated Technology for Verification and Analysis, 9th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 443–452, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-24372-1_33.

[Jan15] Nils Jansen. *Counterexamples in Probabilistic Verification.* PhD thesis, RWTH Aachen University, Germany, 2015.

[Jan22a] Simon Jantsch. Certificates and witnesses for probabilistic model checking – examples, Feb 2022. doi:10.6084/m9.figshare.19209429.

[Jan22b] Simon Jantsch. Certificates and witnesses for probabilistic model checking – supplementary material, Feb 2022. doi:10.6084/m9.figshare.19209303.

[JÁV+12] Nils Jansen, Erika Ábrahám, Matthias Volk, Ralf Wimmer, Joost-Pieter Katoen, and Bernd Becker. The COMICS Tool – Computing Minimal Counterexamples for DTMCs. In Supratik Chakraborty and Madhavan Mukund, editors, *Automated Technology for Verification and Analysis - 10th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 349–353, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-33386-6_27.

[JÁZ+13] Nils Jansen, Erika Ábrahám, Barna Zajzon, Ralf Wimmer, Johann Schuster, Joost-Pieter Katoen, and Bernd Becker. Symbolic Counterexample Generation for Discrete-Time Markov Chains. In Corina S. Păsăreanu and Gwen Salaün, editors, *Formal Aspects of Component Software, 9th International Symposium (FACS)*, Lecture Notes in Computer Science, pages 134–151, Berlin, Heidelberg, 2013. Springer. doi:10.1007/978-3-642-35861-6_9.

[JFB20] Simon Jantsch, Florian Funke, and Christel Baier. Minimal Witnesses for Probabilistic Timed Automata. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis - 18th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 501–517, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-59152-6_28.

[JHFB20] Simon Jantsch, Hans Harder, Florian Funke, and Christel Baier. SWITSS: Computing Small Witnessing Subsystems. In *Proceedings of the 20th Conference on Formal Methods in Computer-Aided Design (FMCAD)*, volume 1, pages 236–244. TU Wien Academic Press, 2020. doi:10.34727/2020/isbn.978-3-85448-042-6_31.

[JPB21] Simon Jantsch, Jakob Piribauer, and Christel Baier. Witnessing Subsystems for Probabilistic Systems with Low Tree Width. In *12th International Symposium on Games, Automata, Logics, and Formal Verification (GandALF)*, volume 346 of *Electronic Proceedings in Theoretical Computer Science*, pages 35–51. Open Publishing Association, Sept. 2021, 2109.08326. doi:10.4204/EPTCS.346.3.

[JRST01] Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed Tree-Width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, May 2001. doi:10.1006/jctb.2000.2031.

[JWÁ+14] Nils Jansen, Ralf Wimmer, Erika Ábrahám, Barna Zajzon, Joost-Pieter Katoen, Bernd Becker, and Johann Schuster. Symbolic counterexample generation for large discrete-time Markov chains. *Science of Computer Programming*, 91:90–114, Oct. 2014. doi:10.1016/j.scico.2014.02.001.

[KÁJW15] Joost-Pieter Katoen, Erika Ábrahám, Nils Jansen, and Ralf Wimmer. High-level Counterexamples for Probabilistic Automata. *Logical Methods in Computer Science*, 11(1), Mar. 2015. doi:10.2168/LMCS-11(1:15)2015.

[Kal83] Lodewijk C. M. Kallenberg. *Linear Programming and Finite Markovian Control Problems*. Mathematical Centre, Amsterdam, 1983.

[Kal94] Lodewijk C. M. Kallenberg. Survey of linear programming for standard and nonstandard Markovian control problems. Part I: Theory. *Zeitschrift für Operations Research*, 40(1):1–42, Mar. 1994. doi:10.1007/BF01414028.

[Kal16] Lodewijk C. M. Kallenberg. Markov Decision Processes. *Lecture Notes*. 2016, University of Leiden.

[Kar72] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, Boston, MA, 1972. doi:10.1007/978-1-4684-2001-2_9.

[Kha79] Leonid Genrikhovich Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk*, 244(5), 1979.

[Kin76] James C. King. Symbolic execution and program testing. *Communications of the ACM*, 19(7):385–394, July 1976. doi:10.1145/360248.360252.

[KLL11] Matthias Kuntz, Florian Leitner-Fischer, and Stefan Leue. From Probabilistic Counterexamples via Causality to Fault Trees. In Francesco Flammini, Sandro Bologna, and Valeria Vittorini, editors, *Computer Safety, Reliability, and Security - 30th International Conference (SAFECOMP)*, Lecture Notes in Computer Science, pages 71–84, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-24270-0_6.

[KLP75] H. T. Kung, Fabrizio Luccio, and Franco P. Preparata. On Finding the Maxima of a Set of Vectors. *Journal of the ACM*, 22(4):469–476, Oct. 1975. doi:10.1145/321906.321910.

[KLS20] Martin Kölbl, Stefan Leue, and Robert Schmid. Dynamic Causes for the Violation of Timed Reachability Properties. In Nathalie Bertrand and Nils Jansen, editors, *Formal Modeling and Analysis of Timed Systems - 18th International Conference (FORMATS)*, Lecture Notes in Computer Science, pages 127–143, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-57628-8_8.

[KLW19] Martin Kölbl, Stefan Leue, and Thomas Wies. Clock Bound Repair for Timed Systems. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference (CAV)*, Lecture Notes in Computer Science, pages 79–96, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-25540-4_5.

[KM17] Jan Křetínský and Tobias Meggendorfer. Efficient Strategy Iteration for Mean Payoff in Markov Decision Processes. In Deepak D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis - 15th International Symposium (ATVA)*, Lecture Notes in Computer Science, pages 380–399, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-68167-2_25.

[KM20] Jan Křetínský and Tobias Meggendorfer. Of Cores: A Partial-Exploration Framework for Markov Decision Processes. *Logical Methods in Computer Science*, 16(4), 2020. doi:10.23638/LMCS-16(4:3)2020.

[KMMM10] Joost-Pieter Katoen, Annabelle McIver, Larissa Meinicke, and Carroll C. Morgan. Linear-Invariant Generation for Probabilistic Programs: Automated Support for Proof-Based Methods. In Radhia Cousot and Matthieu Martel, editors, *Static Analysis - 17th International Symposium (SAS)*, volume 6337 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2010. doi:10.1007/978-3-642-15769-1_24.

[KMMS06] Dieter Kratsch, Ross M. McConnell, Kurt Mehlhorn, and Jeremy P. Spinrad. Certifying Algorithms for Recognizing Interval Graphs and Permutation Graphs. *SIAM Journal on Computing*, 36(2):326–353, Jan. 2006. doi:10.1137/S0097539703437855.

[KNP11] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference (CAV)*, Lecture Notes in Computer Science, pages 585–591, Berlin, Heidelberg, 2011. Springer. doi:10.1007/978-3-642-22110-1_47.

[KNP12] Marta Kwiatkowsa, Gethin Norman, and David Parker. The PRISM Benchmark Suite. In *QEST 2012, Ninth International Conference on Quantitative Evaluation of Systems*, pages 203–204, Sept. 2012. doi:10.1109/QEST.2012.14.

[KNSS02] Marta Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282(1):101–150, June 2002. doi:10.1016/S0304-3975(01)00046-9.

[KNSW07] Marta Kwiatkowska, Gethin Norman, Jeremy Sproston, and Fuzhi Wang. Symbolic model checking for probabilistic timed automata. *Information and Computation*, 205(7):1027–1077, July 2007. doi:10.1016/j.ic.2007.01.004.

[KNVG22] Arut Prakash Kaleeswaran, Arne Nordmann, Thomas Vogel, and Lars Grunske. A systematic literature review on counterexample explanation. *Information and Software Technology*, 145:106800, May 2022. doi:10.1016/j.infsof.2021.106800.

[KS76] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1976.

[KT14] Daniel Kroening and Michael Tautschnig. CBMC – C Bounded Model Checker. In Erika Ábrahám and Klaus Havelund, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 389–391, Berlin, Heidelberg, 2014. Springer. doi:10.1007/978-3-642-54862-8_26.

[KV04] Orna Kupferman and Moshe Y. Vardi. From Complementation to Certification. In Kurt Jensen and Andreas Podelski, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, (TACAS)*, Lecture Notes in Computer Science, pages 591–606, Berlin, Heidelberg, 2004. Springer. doi:10.1007/978-3-540-24730-2_43.

[KZH+11] Joost-Pieter Katoen, Ivan S. Zapreev, Ernst Moritz Hahn, Holger Hermanns, and David N. Jansen. The ins and outs of the probabilistic model checker MRMC. *Performance Evaluation*, 68(2):90–104, Feb. 2011. doi:10.1016/j.peva.2010.04.001.

[LMT07] Ruggero Lanotte, Andrea Maggiolo-Schettini, and Angelo Troina. Parametric probabilistic transition systems for system design and analysis. *Formal Aspects of Computing*, 19(1):93–109, Mar. 2007. doi:10.1007/s00165-006-0015-2.

[LP19] Ratan Lal and Pavithra Prabhakar. Counterexample Guided Abstraction Refinement for Polyhedral Probabilistic Hybrid Systems. *ACM Transactions on Embedded Computing Systems*, 18(5s):98:1–98:23, Oct. 2019. doi:10.1145/3358217.

[LPY97] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a Nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1-2):134–152, 1997. doi:10.1007/s100090050010.

[LS07] François Laroussinie and Jeremy Sproston. State explosion in almost-sure probabilistic reachability. *Information Processing Letters*, 102(6):236–241, June 2007. doi:10.1016/j.ipl.2007.01.003.

[MMNS11] R. M. McConnell, K. Mehlhorn, S. Näher, and P. Schweitzer. Certifying algorithms. *Computer Science Review*, 5(2):119–161, May 2011. doi:10.1016/j.cosrev.2010.09.009.

[MN98] Kurt Mehlhorn and Stefan Näher. From algorithms to working programs: On the use of program checking in LEDA. In Luboš Brim, Jozef Gruska, and Jiří Zlatuška, editors, *Mathematical Foundations of Computer Science, 23rd International*

Symposium (MFCS), Lecture Notes in Computer Science, pages 84–93, Berlin, Heidelberg, 1998. Springer. doi:10.1007/BFb0055759.

[MN99] Kurt Mehlhorn and Stefan Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[MNS+99] Kurt Mehlhorn, Stefan Näher, Michael Seel, Raimund Seidel, Thomas Schilz, Stefan Schirra, and Christian Uhrig. Checking geometric programs or verification of geometric structures. *Computational Geometry*, 12(1):85–103, Feb. 1999. doi:10.1016/S0925-7721(98)00036-4.

[NÁCC14] Johanna Nellen, Erika Ábrahám, Xin Chen, and Pieter Collins. Counterexample Generation for Hybrid Automata. In Cyrille Artho and Peter Csaba Ölveczky, editors, *Formal Techniques for Safety-Critical Systems - Second International Workshop (FTSCS)*, Communications in Computer and Information Science, pages 88–106, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-05416-2_7.

[Nam01] Kedar S. Namjoshi. Certifying Model Checkers. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Computer Aided Verification, 13th International Conference (CAV)*, Lecture Notes in Computer Science, pages 2–13, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-44585-4_2.

[NPW02] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002. doi:10.1007/3-540-45949-9.

[ÖK10] Özgür Özpeynirci and Murat Köksalan. An Exact Algorithm for Finding Extreme Supported Nondominated Points of Multiobjective Mixed Integer Programs. *Management Science*, 56(12):2302–2315, 2010. doi:10.1287/mnsc.1100.1248.

[PO19] William Pettersson and Melih Ozlen. Multi-objective mixed integer programming: An objective space algorithm. *AIP Conference Proceedings*, 2070(1):020039, Feb. 2019. doi:10.1063/1.5090006.

[PPZ01] Doron Peled, Amir Pnueli, and Lenore Zuck. From Falsification to Verification. In Ramesh Hariharan, V. Vinay, and Madhavan Mukund, editors, *Foundations of Software Technology and Theoretical Computer Science, 21st Conference (FSTTCS)*, Lecture Notes in Computer Science, pages 292–304, Berlin, Heidelberg, 2001. Springer. doi:10.1007/3-540-45294-X_25.

[Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. doi:10.1002/9780470316887.

[QJD+15] Tim Quatmann, Nils Jansen, Christian Dehnert, Ralf Wimmer, Erika Ábrahám, Joost-Pieter Katoen, and Bernd Becker. Counterexamples for Expected Rewards. In Nikolaj Bjørner and Frank de Boer, editors, *FM 2015: Formal Methods*, Lecture Notes in Computer Science, pages 435–452, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-19249-9_27.

[QK18] Tim Quatmann and Joost-Pieter Katoen. Sound Value Iteration. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference (CAV)*, Lecture Notes in Computer Science, pages 643–661, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-96145-3_37.

[QS82] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *International Symposium on Programming, 5th Colloquium*, volume 137 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 1982. doi:10.1007/3-540-11494-7_22.

[Ree99] Bruce A. Reed. Introducing Directed Tree Width. *Electronic Notes in Discrete Mathematics*, 3:222–229, May 1999. doi:10.1016/S1571-0653(05)80061-7.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998. doi:10.1145/290163.290168.

[RS86] Neil Robertson and Paul D. Seymour. Graph Minors. II. Algorithmic Aspects of Tree-Width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

[RSM19] Victor Roussanaly, Ocan Sankur, and Nicolas Markey. Abstraction Refinement Algorithms for Timed Automata. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference (CAV)*, Lecture Notes in Computer Science, pages 22–40, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-25540-4_2.

[Saf05] Mohammad Ali Safari. D-Width: A More Natural Measure for Directed Tree Width. In Joanna Jędrzejowicz and Andrzej Szepietowski, editors, *Mathematical Foundations of Computer Science, 30th International Symposium (MFCS)*, Lecture Notes in Computer Science, pages 745–756, Berlin, Heidelberg, 2005. Springer. doi:10.1007/11549345_64.

[Sch99] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, 1999.

[See85] Detlef Seese. Tree-partite graphs and the complexity of algorithms. In Lothar Budach, editor, *Fundamentals of Computation Theory (FCT)*, Lecture Notes in Computer Science, pages 412–421, Berlin, Heidelberg, 1985. Springer. doi:10.1007/BFb0028825.

[SSM04] Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Constraint-Based Linear-Relations Analysis. In Roberto Giacobazzi, editor, *Static Analysis, 11th International Symposium (SAS)*, Lecture Notes in Computer Science, pages 53–68, Berlin, Heidelberg, 2004. Springer. doi:10.1007/978-3-540-27864-1_7.

[ST10] Jeremy Sproston and Angelo Troina. Simulation and Bisimulation for Probabilistic Timed Automata. In *Formal Modeling and Analysis of Timed Systems - 8th International Conference (FORMATS)*, pages 213–227. Springer, Berlin, Heidelberg, Sept. 2010. doi:10.1007/978-3-642-15297-9_17.

[SVV09] Matthias Schmalz, Daniele Varacca, and Hagen Völzer. Counterexamples in Probabilistic LTL Model Checking for Markov Chains. In Mario Bravetti and Gianluigi Zavattaro, editors, *Concurrency Theory, 20th International Conference (CONCUR)*, Lecture Notes in Computer Science, pages 587–602, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-04081-8_39.

[Tho98] Mikkel Thorup. All Structured Programs Have Small Tree Width and Good Register Allocation. *Information and Computation*, 142(2):159–181, May 1998. doi:10.1006/inco.1997.2697.

[Tod91] Seinosuke Toda. PP is as Hard as the Polynomial-Time Hierarchy. *SIAM Journal on Computing*, 20(5):865–877, Oct. 1991. doi:10.1137/0220053.

[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, Oct. 1985. doi:10.1109/SFCS.1985.12.

[VW86] Moshe Y. Vardi and Pierre Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *Proceedings, 1th Annual IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 1986.

[WBB09] Ralf Wimmer, Bettina Braitling, and Bernd Becker. Counterexample Generation for Discrete-Time Markov Chains Using Bounded Model Checking. In Neil D. Jones and Markus Müller-Olm, editors, *Verification, Model Checking, and Abstract Interpretation, 10th International Conference (VMCAI)*, Lecture Notes in Computer Science, pages 366–380, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-540-93900-9_29.

[WH18] Simon Wimmer and Johannes Hölzl. MDP + TA = PTA: Probabilistic Timed Automata, Formalized (Short Paper). In Jeremy Avigad and Assia Mahboubi, editors, *Interactive Theorem Proving - 9th International Conference (ITP)*, Lecture Notes in Computer Science, pages 597–603, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-94821-8_35.

[WHvP20] Simon Wimmer, Frédéric Herbreteau, and Jaco van de Pol. Certifying Emptiness of Timed Büchi Automata. In Nathalie Bertrand and Nils Jansen, editors, *Formal Modeling and Analysis of Timed Systems - 18th International Conference (FORMATS)*, Lecture Notes in Computer Science, pages 58–75, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-57628-8_4.

[Wim16] Simon Wimmer. Formalized Timed Automata. In Jasmin Christian Blanchette and Stephan Merz, editors, *Interactive Theorem Proving - 7th International Conference (ITP)*, Lecture Notes in Computer Science, pages 425–440, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-43144-4_26.

[WJÁ+12] Ralf Wimmer, Nils Jansen, Erika Ábrahám, Bernd Becker, and Joost-Pieter Katoen. Minimal Critical Subsystems for Discrete-Time Markov Models. In Cormac Flanagan and Barbara König, editors, *Tools and Algorithms for the Construction and*

Analysis of Systems - 18th International Conference (TACAS), Lecture Notes in Computer Science, pages 299–314, Berlin, Heidelberg, 2012. Springer. doi:10.1007/978-3-642-28756-5_21.

[WJÁ+14] Ralf Wimmer, Nils Jansen, Erika Ábrahám, Joost-Pieter Katoen, and Bernd Becker. Minimal counterexamples for linear-time probabilistic verification. *Theoretical Computer Science*, 549:61–100, Sept. 2014. doi:10.1016/j.tcs.2014.06.020.

[WJV+13] Ralf Wimmer, Nils Jansen, Andreas Vorpahl, Erika Ábrahám, Joost-Pieter Katoen, and Bernd Becker. High-Level Counterexamples for Probabilistic Automata. In Kaustubh Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors, *QEST 2013, 10th International Conference on Quantitative Evaluation of Systems*, Lecture Notes in Computer Science, pages 39–54, Berlin, Heidelberg, 2013. Springer. doi:10.1007/978-3-642-40196-1_4.

[WL18] Simon Wimmer and Peter Lammich. Verified Model Checking of Timed Automata. In Dirk Beyer and Marieke Huisman, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 61–78, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-89960-2_4.

[Woo09] David R. Wood. On tree-partition-width. *European Journal of Combinatorics*, 30(5):1245–1253, July 2009. doi:10.1016/j.ejc.2008.11.010.

[WvM20] Simon Wimmer and Joshua von Mutius. Verified Certification of Reachability Checking for Timed Automata. In Armin Biere and David Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 26th International Conference (TACAS)*, Lecture Notes in Computer Science, pages 425–443, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-45190-5_24.

[Ye11] Yinyu Ye. The Simplex and Policy-Iteration Methods Are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate. *Mathematics of Operations Research*, 36(4):593–603, Nov. 2011. doi:10.1287/moor.1110.0516.

List of Figures

List of Tables