Minimal Counterexamples for Linear-Time Probabilistic Verification[☆]

Ralf Wimmer[a,*], Nils Jansen[b], Erika Ábrahám[c], Joost-Pieter Katoen[b], Bernd Becker[a]

[a] *Chair of Computer Architecture, Albert-Ludwigs-University Freiburg, Germany*
[b] *Chair for Software Modeling and Verification, RWTH Aachen University, Germany*
[c] *Theory of Hybrid Systems, RWTH Aachen University, Germany*

Abstract

Counterexamples for property violations have a number of important applications like supporting the debugging of erroneous systems and verifying large systems via counterexample-guided abstraction refinement. In this paper, we propose the usage of minimal critical subsystems of discrete-time Markov chains and Markov decision processes as counterexamples for violated ω-regular properties. Minimality can thereby be defined in terms of the number of states or transitions. This problem is known to be NP-complete for Markov decision processes. We show how to compute such subsystems using mixed integer linear programming and evaluate the practical applicability in a number of experiments. They show that our method yields substantially smaller counterexample than using existing techniques.

Keywords: Markov chain, Markov decision process, counterexample, ω-regular property, mixed integer linear programming, SAT-modulo-theories

1. Introduction

Model checking is a prominent technique to check whether a system model exhibits any undesirable behaviors, i. e., behaviors that violate the system specification. In fact, the main power of model checking is its ability to generate such violating behaviors—called *counterexamples*—whenever possible. Model checking can thus be viewed as an intelligent bug hunting technique. Even in cases when a full-fledged state-space exploration is impossible, e. g., if the system's size is too large to be effectively handled, model checking may be able to generate counterexamples provided there is refuting behavior. As Edmund Clarke argues in his talk at the celebration of 25 years of model checking [1]:

> *It is impossible to overestimate the importance of the counterexample feature. The counterexamples are invaluable in debugging complex systems. Some people use model checking just for this feature.*

[☆]This work was partly supported by the German Research Council (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS), the EU-FP7 IRSES project MEALS "Mobility between Europe and Argentina applying Logics to Systems" (Grant No. 295261), the DFG project "CEBug – Counterexample Generation for Stochastic Systems using Bounded Model Checking" (AB 461/1-1) and by the Excellence Initiative of the German federal and state government.

[*]Corresponding author:
Dr. Ralf Wimmer
Chair of Computer Architecture
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 51
79110 Freiburg im Breisgau, Germany
Phone: +49 761 203 8179
Fax: +49 761 203 8142
E-Mail: wimmer@informatik.uni-freiburg.de

Email addresses: wimmer@informatik.uni-freiburg.de (Ralf Wimmer), nils.jansen@informatik.rwth-aachen.de (Nils Jansen), abraham@informatik.rwth-aachen.de (Erika Ábrahám), katoen@informatik.rwth-aachen.de (Joost-Pieter Katoen), becker@informatik.uni-freiburg.de (Bernd Becker)

Other applications of counterexamples include automated refinement of system abstractions as used in the successful CEGAR (counterexample-guided abstraction refinement) framework [2–5].

Research on counterexample generation in model checking is abundant [6–11]. For linear-time specifications such as *ω-regular properties*, counterexamples are simply paths in the Kripke structure $\mathcal{K}$ modeling the system. For instance, for a Büchi automaton specification $\mathcal{A}$ corresponding to the negation of an LTL formula φ, a counterexample is an infinite path in the Kripke structure $\mathcal{K}$ that is admitted by $\mathcal{A}$, i. e., a path that visits one of $\mathcal{A}$'s accepting states infinitely often thus violating φ. The nested depth-first search LTL model-checking algorithm straightforwardly generates such counterexamples while performing the state space exploration without an additional time penalty. Infinite counterexamples are represented in a finite way by a finite path leading to an accepting state followed by a loop containing that state. For *branching-time logics* such as CTL or modal μ-calculus, counterexamples can be (much) more complex, and in general have a tree-like shape [9] instead of a simple path representation as for Büchi automata.

Probabilistic model checking is a variation of traditional model checking that uses system models equipped with randomness such as transition probabilities and/or random delays. Prevailing models in this field are *discrete-time Markov decision processes (MDPs)* and deterministic simplifications thereof, so-called *discrete-time Markov chains (DTMCs)*. MDPs are well-suited to model—amongst others—randomized distributed algorithms. Randomization is used in distributed algorithms to break the symmetry between identical processes in leader election and mutual exclusion algorithms, for routing purposes, or for obtaining consensus—a problem that is known to be practically unsolvable in a deterministic setting as indicated by various results (e. g., [12]). Markov chains are typically used in performance and reliability analysis as for instance in fault tree analysis. Properties that can be model checked on MDP models are safety properties like "The maximal probability to reach a safety-critical state is at most 10^{-3}" or, more generally, maximal probabilities of satisfying ω-regular properties [13, 14] can be obtained. Solving linear programming problems is at the heart of MDP model checking algorithms, whereas for DTMCs this reduces to solving linear equation systems. Tools that support MDP model checking are PRISM [15] and LiQuoR [16]; DTMC model checking is supported by, e. g., MRMC [17] and FMurphi [18]. The PRISM set of case studies [19] convincingly witnesses the applicability of MDP and DTMC model checking.

An important limitation of probabilistic model checking is the lack of *diagnostic feedback* in case a property is violated. Preferably a user would obtain information about why a given property is refuted. It is, however, not clear upfront what counterexamples in the probabilistic setting actually are, let alone on how to determine them algorithmically and efficiently. For instance, if the probability to reach a safety-critical state in a DTMC exceeds the required threshold 10^{-3}, this cannot be illustrated by a single path. In fact, a *set of paths* all reaching the safety-critical state which together carry a probability mass exceeding 10^{-3} would be needed. In case of an MDP, additionally a scheduler is required whose induced Markov chain exceeds the probability threshold 10^{-3}. In the last couple of years, the lack of diagnostic feedback has received more and more attention. Initial approaches [20–23] have focused on computing such sets of paths with sufficient probability mass. Recently, tree-based counterexamples have been proposed to provide evidence that an MDP is not simulated by another one [24, 25].

For DTMCs, it was shown in [20] that computing the smallest number of such paths whose joint probability mass maximally exceeds the threshold (thus yielding the largest possible deviation from the threshold with a minimal number of witnesses) boils down to a k shortest paths problem. Here, k indicates the number of paths in the counterexample and can be computed in an on-the-fly manner. Although this provides a rather intuitive notion of a counterexample that can be efficiently computed (in pseudo-polynomial time in k), the number of paths in many cases is however excessive. In some cases, it is even doubly exponential in the problem size [20], rendering the counterexample practically unusable for debugging purposes. Different proposals have been made to alleviate this problem. To mention a few, [20] represents the path set as a (weighted) regular expression, [21] detects loops on paths, and [22] shrinks paths through strongly connected components (SCCs) into single transitions.

As an alternative to these path-based counterexamples, the usage of winning strategies in probabilistic games [26, 27] and of *critical subsystems* have been proposed in [5, 28, 29]. A critical subsystem is a sub-DTMC of the Markov chain at hand such that the probability to reach a safety-critical state (or, more generally, to satisfy an ω-regular property) inside this sub-DTMC exceeds the probability threshold. This

induces a path-based counterexample by considering all paths leading through this subsystem. Put differently, the sub-DTMC can be viewed as a representation of the set of paths constituting the counterexample. Contrary to the path-based representation, the size of a critical subsystem is bounded by the size of the model under consideration. So as to obtain comprehensive counterexamples, the aim is to obtain *small* critical subsystems. Different heuristic methods have been proposed for computing small critical subsystems: Aljazzar and Leue [28] apply best first search to identify a critical subsystem, while Jansen *et al.* [29] propose a technique that is based on a hierarchical SCC-based abstraction of DTMCs in combination with heuristics for the selection of the states to be contained in the subsystem. Both approaches use heuristic methods to select the states of a critical subsystem and are implemented by the tools `DiPro` [30] and `COMICS` [31], respectively. Although experimental results for these approaches show encouraging results, *minimality of the generated critical subsystems is not guaranteed* (as we show). Moreover, the size is often significantly larger than the minimum (up to two orders of magnitude in some cases).

This paper attempts to fill this gap by presenting an approach to compute a globally *minimal* critical subsystem (MCS) of a given Markov chain or an MDP. Here, minimality refers to the number of states of the subsystem, but our approach can straightforwardly be adapted to minimize the number of transitions. With the notable exception of [32], most approaches for counterexample generation in probabilistic model checking focus on reachability properties. Instead, this paper focuses on generating MCSs for the more general class of ω-regular properties. So, the problem that we are considering is: Given an MDP, an ω-regular property, and a probability threshold λ such that the actual probability violates this threshold, provide a minimal sub-MDP whose maximal probability to satisfy the property exceeds λ. This problem has been proven to be NP-complete [5]. We first consider DTMCs and provide two formulations to this MCS problem: A SAT-modulo theories (SMT) formulation [33] and a mixed integer linear program (MILP) [34]. As the MILP approach clearly outperforms the SMT-approach we focus on the MILP technique and extend this towards MDPs. We will present a number of optimizations which significantly speed up the computation times of the MILP formulation in many cases. Experimental results on a large set of benchmarks are provided, which show the effectiveness of our approach and our optimizations. We show that our MILP approach often yields considerably more compact counterexamples than the heuristic methods [28, 29]. Even in cases where the MILPs cannot be solved to optimality due to time restrictions, the resulting critical subsystems are often substantially smaller than for the heuristic methods [28, 29]. For the sake of understandability, we first present our algorithms for reachability properties and then show how they can be extended to the more general class of ω-regular properties.

Organization of the paper. In Section 2 we introduce the foundations that are needed for this paper. Section 3 presents the generation of MCSs for DTMCs; in Section 4 the approaches are extended to MDPs. In Section 5 we report on experiments on a number of case studies. Finally, we conclude the paper in Section 6.

This paper is an extended and refined version of the papers [35] and [36] that mainly covered DTMCs. This paper discusses the underlying theory in much more depth and extends the theoretical and experimental results to MDPs and ω-regular properties. The correctness of the approach is based on a series of theorems (Theorems 3–6, 8, 9), which are deduced in this paper. Their proofs are provided in the Appendixes.

2. Foundations

We first introduce the probabilistic models and properties that we consider in this paper, briefly describe the model checking algorithms for them, define minimal critical subsystems, and introduce the solver techniques used in this paper.

2.1. Discrete-time Markov Decision Processes

Let S be a countable set. A (sub-stochastic) *distribution* on S is a function $\mu : S \to [0,1] \subseteq \mathbb{R}$ such that $\sum_{s \in S} \mu(s) \leq 1$. We denote the set of all distributions on S by $\mathrm{Distr}(S)$.

Definition 1 (Discrete-time Markov decision process) *Let* AP *be a finite set of atomic propositions. A* discrete-time Markov decision process (MDP) *is a tuple* $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$*, where*

- S *is a countable set of states,*
- $s_{\text{init}} \in S$ *is an initial state,*
- Act *is a finite set of actions,*
- $P : S \times Act \times S \to [0,1] \subseteq \mathbb{R}$ *assigns to each state a set of action-distribution*[1] *pairs such that* $\forall s \in S\ \forall \alpha \in Act : \sum_{s' \in S} P(s, \alpha, s') \leq 1$*, and*
- $L : S \to 2^{AP}$ *is a labeling function which assigns to each state* $s \in S$ *the set of atomic propositions that are true in* s*.*

The MDP $\mathcal{M}$ *is called* finite *if* S *is finite and otherwise* infinite*.*

In the following, if not stated differently, we assume that all MDPs we are dealing with are finite.

If $s \in S$ is the current state of an MDP $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$, its successor state is determined as follows: First a *non-deterministic* choice between the entries of Act is made; say α is chosen. Then the successor state of s is determined *probabilistically* according to the distribution $P(s, \alpha, \cdot)$. We fix the sets

$$\begin{aligned} \operatorname{succ}_{\mathcal{M}}(s,\alpha) &= \{s' \in S \mid P(s,\alpha,s') > 0\}, & \operatorname{succ}_{\mathcal{M}}(s) &= \textstyle\bigcup_{\alpha \in Act} \operatorname{succ}_{\mathcal{M}}(s,\alpha), \\ \operatorname{pred}_{\mathcal{M}}(s,\alpha) &= \{s' \in S \mid P(s',\alpha,s) > 0\}, & \operatorname{pred}_{\mathcal{M}}(s) &= \textstyle\bigcup_{\alpha \in Act} \operatorname{pred}_{\mathcal{M}}(s,\alpha), \text{ and} \\ & & E_{\mathcal{M}} &= \big\{(s,s') \in S \times S \,\big|\, s' \in \operatorname{succ}_{\mathcal{M}}(s)\big\}. \end{aligned}$$

We sometimes skip the index $\mathcal{M}$ when it is clear from the context. Additionally, we define the number of states $\#^S_{\mathcal{M}} = |S|$ and the number of transitions $\#^T_{\mathcal{M}} = \sum_{s \in S} \sum_{\alpha \in Act} |\operatorname{succ}(s,\alpha)|$ of $\mathcal{M}$.

A *finite path* π of $\mathcal{M}$ is a sequence $\pi = s_0 \alpha_0 s_1 \alpha_1 \dots s_n$ with $s_i \in S$ for $i \in \{0, \dots, n\}$ and $\alpha_i \in Act$ for $i \in \{0, \dots, n-1\}$ such that $s_{i+1} \in \operatorname{succ}_{\mathcal{M}}(s_i, \alpha_i)$ for all $i \in \{0, \dots, n-1\}$. We write $\operatorname{last}(\pi)$ for the last state of π, i. e., $\operatorname{last}(\pi) = s_n$. We denote the set of all finite paths in $\mathcal{M}$ by $\operatorname{Paths}^{\text{fin}}_{\mathcal{M}}$ and all finite paths that start in $s \in S$ by $\operatorname{Paths}^{\text{fin}}_{\mathcal{M}}(s)$.

An *infinite path* π of $\mathcal{M}$ is an infinite sequence $\pi = s_0 \alpha_0 s_1 \alpha_1 \dots$ with $s_i \in S$, $\alpha_i \in Act$ and $s_{i+1} \in \operatorname{succ}_{\mathcal{M}}(s_i, \alpha_i)$ for all $i \geq 0$. We use the notation $\operatorname{Paths}^{\text{inf}}_{\mathcal{M}}$ for the set of all infinite paths and $\operatorname{Paths}^{\text{inf}}_{\mathcal{M}}(s)$ for those starting in $s \in S$. The state at position i of path π is denoted by $\pi^{(i)}$, i. e., $\pi^{(i)} = s_i$. The *trace* of a (finite or infinite) path $\pi = s_0 \alpha_0 s_1 \alpha_1 \dots$ is the sequence $\operatorname{trace}(\pi) = L(s_0) L(s_1) \dots$

Before probability measures can be defined for MDPs, the non-determinism has to be resolved. This is done by an entity called *scheduler*.

Definition 2 (Scheduler) *A* scheduler *for an MDP* $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ *is a function* $\sigma : \operatorname{Paths}^{\text{fin}}_{\mathcal{M}}(s_{\text{init}}) \to \operatorname{Distr}(Act)$*. We denote the set of schedulers on* $\mathcal{M}$ *by* $\operatorname{Sched}_{\mathcal{M}}$*.*

A scheduler can be used to transform the non-deterministic choice of the next action into a probabilistic choice, which depends on the path along which the current state is reached from the initial state. The resulting MDP is deterministic regarding the choice of actions.

Definition 3 (Discrete-time Markov chain) *A* discrete-time Markov chain *(DTMC) is an MDP* $\mathcal{D} = (S, s_{\text{init}}, Act, P, L)$ *with* $|Act| = 1$*.*

We use $\mathcal{M}$ as notation for arbitrary MDPs and $\mathcal{D}$ for DTMCs. In the case of DTMCs we omit the action and write, e. g., $P(s, s')$ instead of $P(s, \alpha, s')$ for transition probabilities, $s_0 s_1 \dots$ instead of $s_0 \alpha_0 s_1 \alpha_1 \dots$ for paths and $(S, s_{\text{init}}, P, L)$ instead of $(S, s_{\text{init}}, Act, P, L)$. For a DTMC $\mathcal{D}$, a probability measure is defined on certain sets of infinite paths using the following construction: The *cylinder set* of a finite path $\pi \in \operatorname{Paths}^{\text{fin}}_{\mathcal{D}}$

[1] Please note that we allow sub-stochastic distributions. Usually, the sum of probabilities is required to be exactly 1. This can be obtained by defining $\mathcal{M}' = (S \uplus \{s_\perp\}, s_{\text{init}}, Act, P', L')$ such that (i) $s_\perp$ is a fresh sink state, (ii) P' extends P with $P'(s_\perp, \alpha, s_\perp) = 1$, $P'(s, \alpha, s_\perp) = 1 - \sum_{s' \in S} P(s, \alpha, s')$ and $P'(s_\perp, \alpha, s) = 0$ for all $s \in S$ and $\alpha \in Act$, and (iii) L' extends L with $L'(s_\perp) = \emptyset$.

is the set $\mathrm{cyl}(\pi) = \{\pi' \in \mathrm{Paths}^{\inf}_{\mathcal{D}} \mid \pi \text{ is a prefix of } \pi'\}$ of all infinite extensions of π. For the DTMC $\mathcal{D}$ and a state $s_0 \in S$, a *probability space* $(\Omega, \mathcal{F}, \Pr^{s_0}_{\mathcal{D}})$ can be defined as follows: The *sample space* $\Omega = \mathrm{Paths}^{\inf}_{\mathcal{D}}(s_0)$ is the set of all infinite paths starting in s_0. The *events* $\mathcal{F} \subseteq 2^{\Omega}$ are given by the unique smallest σ-algebra that contains the cylinder sets of all finite paths in $\mathrm{Paths}^{\mathrm{fin}}_{\mathcal{D}}(s_0)$, i. e., it is the closure of the cylinder sets under complement and countable union, including Ω. The *probability measure* $\Pr^{s_0}_{\mathcal{D}} : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ is the unique measure extending $\Pr^{s_0}_{\mathcal{D}}\big(\mathrm{cyl}(s_0 s_1 \dots s_n)\big) = \prod_{i=0}^{n-1} P(s_i, s_{i+1})$ to the whole σ-algebra [37]. A set $\Pi \subseteq \mathrm{Paths}^{\inf}_{\mathcal{D}}(s_0)$ of paths is *measurable* iff $\Pi \in \mathcal{F}$.

Now we return to MDPs and schedulers. A scheduler $\sigma \in \mathrm{Sched}_{\mathcal{M}}$ for an MDP $\mathcal{M} = (S, s_{\mathrm{init}}, Act, P, L)$ induces an (infinite) DTMC $\mathcal{M}^{\sigma} = (\mathrm{Paths}^{\mathrm{fin}}(s_{\mathrm{init}}), s_{\mathrm{init}}, P^{\sigma}, L^{\sigma})$ with $P^{\sigma}(\pi, \pi') = \sigma(\pi)(\alpha) \cdot P(\mathrm{last}(\pi), \alpha, s)$ if $\pi' = \pi\alpha s$, and $P^{\sigma}(\pi, \pi') = 0$ otherwise. The labeling function L^{σ} is given by $L^{\sigma}(\pi) = L\big(\mathrm{last}(\pi)\big)$. The probabilities of path properties of MDPs under scheduler σ are computed in this induced DTMC.

In the following, we do not need schedulers whose return value may depend on the complete path that led from the initial state to the current state. Instead, for our purposes—the computation of counterexamples for ω-regular properties—the subclass of *memoryless deterministic* schedulers suffices [38, Lemma 10.102]. The distribution assigned to a finite path by a memoryless scheduler only depends on the last state of the path. A scheduler is deterministic if it removes the non-determinism by choosing for each finite path a single action with probability 1.

Definition 4 (Memoryless and deterministic schedulers) *Let $\mathcal{M} = (S, s_{\mathrm{init}}, Act, P, L)$ be an MDP. A scheduler σ for $\mathcal{M}$ is* memoryless *iff for all $\pi, \pi' \in \mathrm{Paths}^{\mathrm{fin}}_{\mathcal{M}}$ with $\mathrm{last}(\pi) = \mathrm{last}(\pi')$ we have that $\sigma(\pi) = \sigma(\pi')$. A scheduler σ for $\mathcal{M}$ is* deterministic *iff for all $\pi \in \mathrm{Paths}^{\mathrm{fin}}_{\mathcal{M}}$ and $\alpha \in Act$ we have that $\sigma(\pi)(\alpha) \in \{0, 1\}$.*

Memoryless deterministic schedulers can be regarded as (potentially partial) functions $\sigma : S \to Act$. The induced DTMC of a memoryless scheduler σ is bisimilar to $\mathcal{M}^{\sigma,\mathrm{md}} = (S, s_{\mathrm{init}}, P', L)$ with $P'(s, s') = P(s, \sigma(s), s')$. Note that for finite MDPs this yields a finite DTMC. If not stated differently, in the following we always refer to $\mathcal{M}^{\sigma,\mathrm{md}}$ (instead of $\mathcal{M}^{\sigma}$) as the DTMC induced by a memoryless deterministic scheduler.

2.2. Reachability Properties and their Model Checking

A *linear-time property* over the set AP of atomic propositions is a set $\mathcal{L}$ of traces $\gamma_0\gamma_1\gamma_2\dots$ with $\gamma_i \subseteq \mathrm{AP}$ for all i. In this paper we will deal with a certain class of linear-time properties, namely *ω-regular properties*. Before dealing with this more general case, we address the important subclass of *reachability properties*.

2.2.1. Reachability Properties

A reachability property is a linear-time property which contains all traces that have a sequence element containing a given proposition.

Definition 5 (Reachability property) *The* reachability property $\Diamond\, a$ *for proposition $a \in \mathrm{AP}$ is the linear-time property*

$$\Diamond\, a = \{\gamma_0\gamma_1 \dots \in (2^{\mathrm{AP}})^{\omega} \mid \exists i \geq 0 : a \in \gamma_i\} \ .$$

A path π of a DTMC $\mathcal{D}$ satisfies a reachability property $\Diamond\, a$ with $a \in \mathrm{AP}$, written $\pi \models \Diamond\, a$, if $\mathrm{trace}(\pi) \in \Diamond\, a$. We are interested in the total probability $\Pr^{s_{\mathrm{init}}}_{\mathcal{D}}(\Diamond\, a)$ of all paths[2] starting in the initial state and satisfying the reachability property $\Diamond\, a$. To be more precise, we want to check whether this total probability is between some bounds. The case for lower bounds can be led back to upper bounds. In the following we restrict ourselves to non-strict upper bounds; the case for strict upper bounds is similar. We use the notation $\mathcal{D} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ to express that $\Pr^{s_{\mathrm{init}}}_{\mathcal{D}}(\Diamond\, a)$ is at most the bound $\lambda \in [0,1] \subseteq \mathbb{R}$. For MDPs, $\mathcal{M} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ expresses that for all schedulers σ of $\mathcal{M}$ we have that $\mathcal{M}^{\sigma} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$.

[2] In the notation $\Pr^{s}_{\mathcal{D}}(\Diamond\, a)$ we overload $\Diamond\, a$ to denote the set of paths of $\mathcal{D}$ starting in s and satisfying $\Diamond\, a$. Note that this set of paths is measurable in the probability space introduced in Section 2.1, see [39].

2.2.2. Model Checking Reachability Properties

Prior to checking $\mathcal{D} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ for a DTMC $\mathcal{D} = (S, s_{\text{init}}, P, L)$, we compute the set $S_{\mathcal{D}}^{\text{rel}(a)} = \{s \in S \mid \Pr_{\mathcal{D}}^{s}(\Diamond\, a) > 0\}$ of those states of $\mathcal{D}$ from which an a-state is reachable. These states, which we call *relevant* for a, can be determined in linear time by applying a backward reachability analysis from the set of a-states. Paths containing *irrelevant* states $s \notin S_{\mathcal{D}}^{\text{rel}(a)}$ do not contribute to the probability $\Pr_{\mathcal{D}}^{s_{\text{init}}}(\Diamond\, a)$. Optionally, the set of relevant states can be reduced by removing all states that are unreachable from the initial state. Also this computation can be done in linear time using a forward reachability analysis from s_{init}.

After this pre-processing, the probabilities $p_s = \Pr_{\mathcal{D}}^{s}(\Diamond\, a)$ for all states[3] $s \in S$ are obtained as the unique solution of the following linear equation system [38, p. 760]:

- $p_s = 1$ for all states $s \in S_{\mathcal{D}}^{\text{rel}(a)}$ with $a \in L(s)$,
- $p_s = 0$ for all states $s \notin S_{\mathcal{D}}^{\text{rel}(a)}$, and
- $p_s = \sum_{s' \in S} P(s, s') \cdot p_{s'}$ for all other states.

Finally, $\mathcal{D} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ holds iff $p_{s_{\text{init}}} \leq \lambda$.

The procedure for checking $\mathcal{M} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ for an MDP $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ is similar. A state $s \in S$ is *relevant* for a if there exists a scheduler under which an a-state is reachable from s, i. e.,

$$S_{\mathcal{M}}^{\text{rel}(a)} = \{s \in S \mid \exists \sigma \in \text{Sched}_{\mathcal{M}} : \Pr_{\mathcal{M}^{\sigma}}^{s}(\Diamond\, a) > 0\}.$$

Again, the set of relevant states can be computed in linear time by a backward reachability analysis on $\mathcal{M}$ [38, Algorithm 46]. To check whether $\mathcal{M}^{\sigma} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ holds for all schedulers σ of MDP $\mathcal{M}$, it suffices to consider a memoryless deterministic scheduler σ^* that maximizes the reachability probability for $\Diamond\, a$, and to check whether $\Pr_{\mathcal{M}^{\sigma^*}}^{s_{\text{init}}}(\Diamond\, a) \leq \lambda$ [38, Lemma 10.102]. The maximal probabilities $p_s = \Pr_{\mathcal{M}^{\sigma^*}}^{s}(\Diamond\, a)$ for each $s \in S$ can be characterized by the following equation system:

- $p_s = 1$ for all $s \in S_{\mathcal{M}}^{\text{rel}(a)}$ with $a \in L(s)$,
- $p_s = 0$ for all $s \notin S_{\mathcal{M}}^{\text{rel}(a)}$ and
- $p_s = \max\{\sum_{s' \in S} P(s, \alpha, s') \cdot p_{s'} \mid \alpha \in Act\}$ for all other states.

This equation system can be transformed into a linear optimization problem that yields the maximal reachability probability together with an optimal scheduler [38, Theorem 10.105].

2.3. ω-Regular Properties and their Model Checking

Now we consider the more general class of *ω-regular properties* and briefly describe the model checking algorithms for them.

2.3.1. ω-Regular Properties

For defining and model checking ω-regular properties on MDPs, we follow the standard automata-theoretic approach, as described, e. g., in [38, 40–42], making use of deterministic Rabin automata.

Definition 6 (Deterministic Rabin automaton) *A* deterministic Rabin automaton (DRA) *is a tuple $\mathcal{A} = (Q, q_{\text{init}}, \Sigma, \delta, F)$ such that Q is a finite, nonempty set of states, $q_{\text{init}} \in Q$ is an initial state, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, and $F \subseteq 2^Q \times 2^Q$ is an acceptance condition.*

[3] For all reachable states if the unreachable ones are declared to be irrelevant.

A *run* r of $\mathcal{A}$ is a state sequence $q_0 q_1 q_2 \ldots \in Q^\omega$ with $q_0 = q_{\text{init}}$ such that for all $i \geq 0$ there is a $\gamma_i \in \Sigma$ with $q_{i+1} = \delta(q_i, \gamma_i)$. We say that r is the (unique) run of $\mathcal{A}$ on the infinite word $\gamma_0 \gamma_1 \ldots$ over Σ. By $\inf(r)$ we denote the set of all states which appear infinitely often in the run r. Given the acceptance condition $F = \big\{(R_i, A_i) \,|\, i = 1, \ldots, n\big\}$, a run r is *accepting* if, for some $i \in \{1, \ldots, n\}$, $\inf(r) \cap R_i = \emptyset$ and $\inf(r) \cap A_i \neq \emptyset$. We denote the set of infinite words over Σ with an accepting run of $\mathcal{A}$ by $\mathcal{L}(\mathcal{A})$.

Definition 7 (ω-Regular property, Safra [43]) *A linear-time property $\mathcal{L}$ is* ω-regular *iff there is a DRA $\mathcal{A}$ with $\mathcal{L} = \mathcal{L}(\mathcal{A})$.*

Assume a set AP of atomic propositions, a DRA $\mathcal{A}$ with alphabet 2^{AP} and the ω-regular property $\mathcal{L} = \mathcal{L}(\mathcal{A})$. A path π of a DTMC $\mathcal{D}$ satisfies $\mathcal{L}$ if the run of $\mathcal{A}$ on $\text{trace}(\pi)$ is accepting. We are interested in the question whether $\mathcal{D} \models \mathcal{P}_{\leq\lambda}(\mathcal{L})$, i. e., whether the total probability[4] $\Pr_{\mathcal{D}}^{s_{\text{init}}}(\mathcal{L})$ to walk along a path in $\mathcal{D}$ which starts in s_{init} and satisfies $\mathcal{L}$ is at most a given upper bound $\lambda \in [0, 1] \subseteq \mathbb{R}$. An MDP $\mathcal{M}$ satisfies the property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ iff the property is satisfied for all schedulers, i. e., if $\mathcal{M}^\sigma \models \mathcal{P}_{\leq\lambda}(\mathcal{L})$ for all $\sigma \in \text{Sched}_{\mathcal{M}}$.

2.3.2. Model Checking ω-Regular Properties

We consider an ω-regular property $\mathcal{L}$ and assume that a DRA $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ with $\mathcal{L} = \mathcal{L}(\mathcal{A})$ is given. Checking the property $\mathcal{L}$ for an MDP $\mathcal{M}$ can be carried out by building the product automaton of the MDP $\mathcal{M}$ with the DRA $\mathcal{A}$ and computing reachability probabilities therein.

Definition 8 (Product automaton) *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP over the atomic propositions* AP *and $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA with $F = \big\{(R_i, A_i) \,\big|\, i = 1, \ldots, n\big\}$. The* product automaton of $\mathcal{M}$ and $\mathcal{A}$ *is an MDP $\mathcal{M} \otimes \mathcal{A} = (S \times Q, (s, q)_{\text{init}}, Act, P', L')$ over the set* AP' *of atomic propositions such that*

- $(s, q)_{\text{init}} = \big(s_{\text{init}}, \delta(q_{\text{init}}, L(s_{\text{init}}))\big)$,
- $P'\big((s, q), \alpha, (s', q')\big) = \begin{cases} P(s, \alpha, s') & \text{if } q' = \delta\big(q, L(s')\big), \\ 0 & \text{otherwise,} \end{cases}$
- $\text{AP}' = \{R_i, A_i \,|\, i = 1, \ldots, n\}$, *and*
- $A_i \in L'(s, q)$ *iff* $q \in A_i$, *and* $R_i \in L'(s, q)$ *iff* $q \in R_i$, *for* $i = 1, \ldots, n$.

We first explain how to check ω-regular properties on the simpler model of DTMCs and cover the model checking of MDPs afterwards. Given a DTMC $\mathcal{D}$ and an ω-regular property $\mathcal{L}$, we consider the product automaton of $\mathcal{D}$ with the DRA $\mathcal{A}$ of $\mathcal{L}$. Note that the product automaton in this case is again a DTMC. The next step is to determine the strongly connected components (SCCs) of the product DTMC.

Definition 9 (Strongly connected component) *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC and $\emptyset \neq S' \subseteq S$.*

1. *S' is* strongly connected *iff for all $s, s' \in S'$ there is a path $s_0 s_1 \ldots s_n \in \text{Paths}_{\mathcal{D}}^{\text{fin}}$ with $s_0 = s$, $s_n = s'$ and $s_i \in S'$ for all $i = 0, \ldots, n$.*
2. *S' is a* strongly connected component (SCC) *of $\mathcal{D}$ iff it is strongly connected and maximal, i. e., for all strongly connected sets $S'' \subseteq S$ we have that $S' \not\subset S''$.*
3. *The set of* input states *of an SCC S' is defined as* $\text{In}(S') = \{s \in S' \,|\, \text{pred}_{\mathcal{D}}(s) \not\subseteq S'\}$.
4. *The set of* output states *of an SCC S' is defined as* $\text{Out}(S') = \{s \in S \setminus S' \,|\, \text{pred}_{\mathcal{D}}(s) \cap S' \neq \emptyset\}$.
5. *S' is a* bottom SCC (BSCC) *iff it is an SCC and for all $s \in S'$ we have that $\sum_{s' \in S'} P(s, s') = 1$.*

[4] Again, in $\Pr_{\mathcal{D}}^{s}(\mathcal{L})$ we overload $\mathcal{L}$ to denote the set $\{\pi \in \text{Paths}_{\mathcal{D}}^{\inf}(s) \mid \text{trace}(\pi) \in \mathcal{L}\}$ of paths of $\mathcal{D}$ starting in s and satisfying $\mathcal{L}$. For each ω-regular property $\mathcal{L}$, this set of paths is measurable in the probability space defined in Section 2.1, see [39].

The SCC structure of a DTMC can be determined by Tarjan's algorithm in linear time [44]. Paths can enter (exit) an SCC through its input (output) states. BSCCs have no output states, i. e., the probability to visit each state in a BSCC infinitely often is one. However, $\mathrm{Out}(S') = \emptyset$ is not sufficient to assure that an SCC S' is bottom, since we allow sub-stochastic distributions.

Definition 10 (Accepting BSCC) *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC over* AP*, $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA and $\mathcal{D} \otimes \mathcal{A} = (S \times Q, (s,q)_{\text{init}}, Act, P', L')$ their product. A BSCC $B \subseteq S \times Q$ of $\mathcal{D} \otimes \mathcal{A}$ is called* accepting *if there are $(R_i, A_i) \in F$ and $(s,q) \in B$ with $A_i \in L'(s,q)$ and $R_i \notin L'(s',q')$ for all $(s',q') \in B$.*

We introduce the proposition *accept* and extend the labeling by *accept* $\in L'(s,q)$ iff (s,q) is a state in an accepting BSCC of $\mathcal{D} \otimes \mathcal{A}$. Then the following theorem holds:

Theorem 1 ([40]) *Let $\mathcal{D}$ be a DTMC, $\mathcal{L}$ an ω-regular property, and $\mathcal{A}$ a DRA with $\mathcal{L} = \mathcal{L}(\mathcal{A})$. Then:*

$$\Pr_{\mathcal{D}}^{s_{\text{init}}}(\mathcal{L}) = \Pr_{\mathcal{D}\otimes\mathcal{A}}^{(s,q)_{\text{init}}}(\Diamond\ \mathit{accept})\ .$$

Due to the dependence on a scheduler, the notion of accepting BSCCs is not directly applicable to MDPs, instead, so-called *end components* are introduced.

Definition 11 (Accepting end component) *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP over* AP*.*

1. *A* sub-MDP *of $\mathcal{M}$ is a non-empty set of states $S' \subseteq S$ such that there exists an action function $A : S' \to 2^{Act} \setminus \emptyset$ with $\mathrm{succ}_{\mathcal{M}}(s,\alpha) \subseteq S'$ for all states $s \in S'$ and actions $\alpha \in A(s)$.*
2. *A sub-MDP S' with action function A is an* end component *of $\mathcal{M}$ if the directed graph $G = (S', V)$ with $V = \big\{(s,s') \in S' \times S' \mid \exists \alpha \in A(s) : s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha)\big\}$ is strongly connected and $\sum_{s' \in S'} P(s,\alpha,s') = 1$ for all $s \in S'$ and $\alpha \in A(s)$.*
3. *Let $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ be a DRA. An end component $B \subseteq S \times Q$ of $\mathcal{M} \otimes \mathcal{A}$ is* accepting *if there are $(R_i, A_i) \in F$ and $(s,q) \in B$ such that $A_i \in L'(s,q)$ and $R_i \notin L'(s',q')$ for all $(s',q') \in B$.*

Intuitively speaking, S' is an end component iff there is a scheduler σ such that S' is a BSCC of the induced DTMC. An end component is accepting iff there is a pair $(R_i, A_i) \in F$ such that the label A_i occurs in the end component while R_i does not. We again extend the labeling of $\mathcal{M} \otimes \mathcal{A}$ such that *accept* $\in L'(s,q)$ iff (s,q) belongs to an accepting end component. To determine whether $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ is satisfied by $\mathcal{M}$, it suffices to compute whether $\Pr_{\mathcal{M}^{\sigma^*}}^{s_{\text{init}}}(\mathcal{L}) = \max\limits_{\sigma \in \mathrm{Sched}_{\mathcal{M}}} \Pr_{\mathcal{M}^{\sigma}}^{s_{\text{init}}}(\mathcal{L})$ is at most λ.

Theorem 2 ([40]) *Let $\mathcal{M}$ be an MDP, $\mathcal{L}$ an ω-regular property and $\mathcal{A}$ a DRA with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Then*

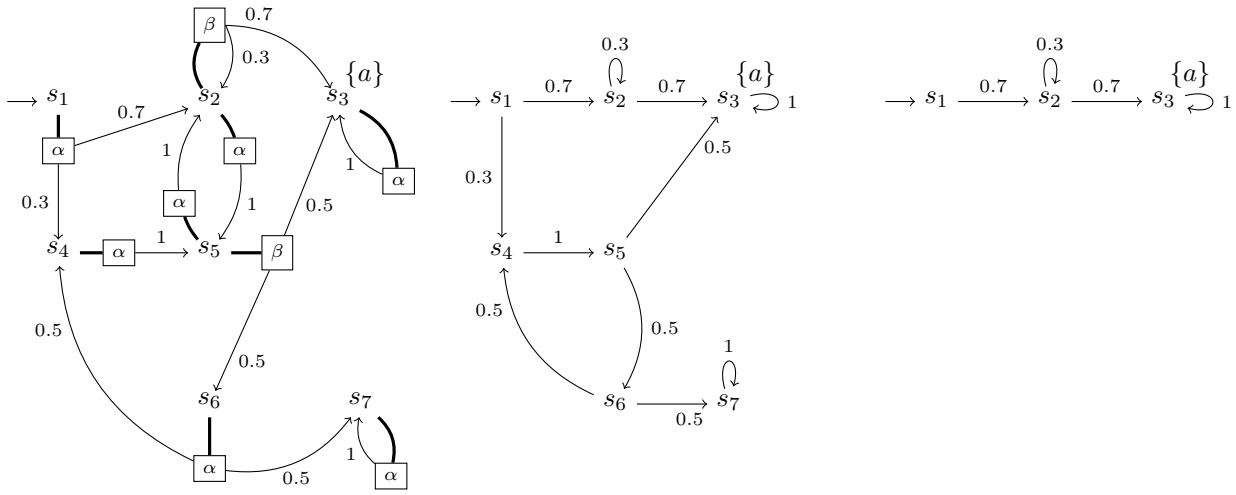
$$\Pr_{\mathcal{M}^{\sigma^*}}^{s_{\text{init}}}(\mathcal{L}) = \Pr_{\mathcal{M}^{\sigma^*}\otimes\mathcal{A}}^{(s,q)_{\text{init}}}(\Diamond\ \mathit{accept})\ .$$

2.4. Minimal Critical Subsystems

Let $\mathcal{M}$ be an MDP and consider $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ for an ω-regular property $\mathcal{L}$. Assume that $\Pr_{\mathcal{M}}^{s_{\text{init}}}(\mathcal{L}) > \lambda$. The goal is to identify a smallest possible part $\mathcal{M}'$ of $\mathcal{M}$ such that $\Pr_{\mathcal{M}'}^{s_{\text{init}}}(\mathcal{L}) > \lambda$.

Definition 12 (Minimal critical subsystem) *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP.*

1. *An MDP $\mathcal{M}' = (S', s'_{\text{init}}, Act', P', L')$ is a* subsystem *of $\mathcal{M}$ if $S' \subseteq S$, $s'_{\text{init}} = s_{\text{init}}$, $L'(s) = L(s)$ for all $s \in S'$, $Act' \subseteq Act$, and $P'(s,\alpha,s') > 0$ implies $P'(s,\alpha,s') = P(s,\alpha,s')$ for all $s, s' \in S'$ and $\alpha \in Act'$.*
2. *A subsystem $\mathcal{M}'$ of $\mathcal{M}$ is* critical *for property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ if $\mathcal{M}' \not\models \mathcal{P}_{\leq\lambda}(\mathcal{L})$.*
3. *A* minimal *critical subsystem (MCS) of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ is a critical subsystem of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with a minimal number of states among all critical subsystems of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$.*



(a) Example MDP (b) DTMC induced by scheduler σ (c) Minimal critical subsystem

Figure 1: MDP $\mathcal{M}$, induced DTMC $\mathcal{D}$ and minimal critical subsystem $\mathcal{D}'$ for $\mathcal{P}_{\leq 0.6}(\Diamond\, a)$.

Alternatively, minimality of critical subsystems could be defined in terms of the number of transitions. Although in this paper we focus on state-minimality, our approach can be easily adapted to transition-minimality.

Given that an MCS $\mathcal{M}'$ violates $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$, there exists a memoryless deterministic scheduler σ on $\mathcal{M}'$ such that the probability of $\Diamond\, a$ in the induced DTMC $\mathcal{M}'^{\sigma}$ exceeds λ. For ω-regular properties $\mathcal{L}$, the same holds for the product with a DRA for $\mathcal{L}$. Thus, in order to determine an MCS of an MDP, it suffices to consider memoryless deterministic schedulers. This fact is exploited in our approach later on.

Example 1 *Consider the MDP $\mathcal{M}$ shown in Figure 1(a) with initial state s_1. The bold edges indicate a nondeterministic choice with actions α and β, resp. From the corresponding action nodes, a probabilistic choice of the successor states is indicated by the edges labeled with probabilities. We define a memoryless deterministic scheduler $\sigma\colon S \to Act$ as in Definition 4, which assigns β to s_2 and s_5, and α to all other states. The scheduler σ maximizes the probability of reaching s_3; the corresponding induced DTMC $\mathcal{D}$ is depicted in Figure 1(b).*

Consider a reachability property $\mathcal{P}_{\leq 0.6}(\Diamond\, a)$. State s_3 is the only target state. The overall probability of reaching s_3 in $\mathcal{D}$ is 0.9 *which means the property is violated both for $\mathcal{D}$ and $\mathcal{M}$. An* MCS *for $\mathcal{D}$ and $\mathcal{P}_{\leq 0.6}(\Diamond\, a)$ is given in Figure 1(c), where the probability of reaching s_3 is* 0.7*.*

2.5. SAT-Modulo-Theories

SAT-modulo-theories (SMT) [33] refers to a generalization of the classical propositional satisfiability problem (SAT). Compared to SAT problems, in an SMT formula atomic propositions can be replaced by atoms of a given theory, e. g., linear or polynomial (in)equalities. We use linear real arithmetic (LRA) as theory for the computation of MCSs. SMT problems are typically solved by the combination of a DPLL-procedure (as used for deciding SAT problems) with a theory solver that is able to decide the satisfiability of conjunctions of theory atoms. For a description of such a combined algorithm for SMT problems over LRA see [45]. Several tools for solving SMT formulae over LRA are available, e. g., Z3 [46], CVC [47], and MathSAT [48].

2.6. Mixed Integer Linear Programming

A *mixed integer linear program* optimizes an objective function under a condition specified by a conjunction of linear inequalities. A subset of the variables in the inequalities is restricted to take only integer values, which makes solving MILPs NP-hard [49, Problem MP1].

Definition 13 (Mixed integer linear program) *Let $A \in \mathbb{Q}^{m\times n}$, $B \in \mathbb{Q}^{m\times k}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$, and $d \in \mathbb{Q}^k$. A* mixed integer linear program *(MILP) consists in computing $\min c^T x + d^T y$ such that $Ax + By \leq b$ and $x \in \mathbb{R}^n,\ y \in \mathbb{Z}^k$.*

MILPs are typically solved by a combination of a branch-and-bound algorithm with the generation of so-called cutting planes. These algorithms heavily rely on the fact that relaxations of MILPs which result by removing the integrality constraints can be solved efficiently. MILPs are widely used in operations research, hardware-software co-design, and numerous other applications. Efficient open source as well as commercial implementations are available like Gurobi [50], Scip [51] or Cplex [52]. We refer the reader to, e. g., [34] for more information on solving MILPs.

3. Minimal Critical Subsystems for DTMCs

In this section we present two approaches for computing MCSs of DTMCs: one using SMT and one using MILP solvers. We start with reachability properties. Since our practical experiments revealed that the MILP approach is clearly superior in terms of computation times, we only generalize the MILP approach to ω-regular properties. An important advantage of using MILP solvers is that during the solving process a lower bound on the optimal solution is obtained while both the current solution (i. e., the currently obtained critical subsystem) and the lower bound are successively improved. That is to say, on halting the MILP solver, a user obtains the best solution so far, as well as a precise indication of the size of an MCS. We start with a basic encoding of the problem to find an MCS, and then provide several optimizations in the form of *redundant constraints* that are aimed at speeding up the solving process by detecting conflicts at an earlier stage.

3.1. Reachability Properties: An SMT Formulation

In order to obtain an MCS for a property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ violated by a DTMC $\mathcal{D} = (S, s_{\text{init}}, P, L)$, we provide an SMT formula over LRA whose satisfying variable assignments correspond to the critical subsystems (of arbitrary size) of $\mathcal{D}$. Let $T = \{s \in S \mid a \in L(s)\}$ be the set of target states. For simplicity we assume that all for a irrelevant states have been removed from the DTMC $\mathcal{D}$. An MCS is then obtained by minimizing over the number of (relevant) states in $\mathcal{D}$.

For our SMT formulation we introduce for each state $s \in S$ a *characteristic variable* $x_s \in [0,1] \subseteq \mathbb{R}$ where $x_s = 1$ or $x_s = 0$ will be ensured by the formula. A state $s \in S$ is contained in the subsystem iff $x_s = 1$ in the satisfying assignment. Additionally, we use a real-valued variable $p_s \in [0,1] \subseteq \mathbb{R}$ for each state $s \in S$ to keep track of the reachability probability of a target state from s within the subsystem. The SMT formulation reads:

$$\text{minimize} \quad \sum_{s\in S} x_s \tag{1a}$$

$$\text{such that}$$

$$\forall s \in T: \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \big(x_s = 1 \wedge p_s = 1\big) \tag{1b}$$

$$\forall s \in S \setminus T: \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \big(x_s = 1 \wedge p_s = \sum_{s'\in \text{succ}(s)} P(s,s') \cdot p_{s'}\big) \tag{1c}$$

$$p_{s_{\text{init}}} > \lambda, \tag{1d}$$

where $\oplus$ denotes exclusive or. As we are interested in a *minimal* critical subsystem, we have to minimize the number of x_s-variables with value 1. This corresponds to minimizing the sum over all x_s-variables (line 1a). If x_s is zero, the corresponding state s does not belong to the subsystem. Then its reachability probability is zero (first part of lines 1b and 1c). Target states that are contained in the subsystem have probability one (second part of line 1b). Note that an MCS does not need to contain all target states. The reachability probability of all non-target states in the subsystem is given as the weighted sum over the probabilities of

their successor states (line 1c). In order to obtain a critical subsystem we additionally require $p_{s_{\text{init}}}$ to exceed λ (line 1d). Note that the size of the resulting SMT formula is in $O(\#_{\mathcal{D}}^{S} + \#_{\mathcal{D}}^{T})$.

Soundness and completeness of the SMT formulation are stated in the following theorem, whose proof is given in Appendix A. Soundness in this case means that each satisfying assignment induces an MCS, completeness means that each MCS is induced by some satisfying assignment.

Theorem 3 *The SMT formulation* (1a)–(1d) *is sound and complete.*

Since most state-of-the-art SMT solvers for LRA cannot cope with minimizing objective functions, we apply a binary search in the range $\{1, \ldots, |S|\}$ to obtain the optimal value of the objective function. Starting with $k_l = 1$ and $k_u = |S|$, we iteratively search for critical subsystems whose number of states is between k_l and $k_m := k_l + (k_u - k_l)/2$. If we find such a subsystem with k states, then we set k_u to $k-1$; otherwise, we set k_l to k_m+1. The search is repeated until $k_u < k_l$. The smallest k for which a satisfying assignment was found yields the size of the MCS at hand. The SMT encoding yields a suitable and intuitive method to compute MCSs. However, our experiments reveal that obtaining a satisfying assignment for larger DTMCs is rather time-consuming. This is mainly due to the high number of disjunctions in the formula which trigger relatively few implications, forcing the solver to attempt many different cases while searching for a satisfying assignment.

3.2. Reachability Properties: An MILP Formulation

To overcome this limitation, we now provide an MILP formulation for finding an MCS for reachability properties. As before, we assume the DTMC at hand to only contain relevant states. In order to avoid disjunctions, we explicitly require the characteristic variables x_s for each $s \in S$ to be integer. As before, we have variables $p_s \in [0,1] \subseteq \mathbb{R}$. The MILP formulation of finding an MCS for reachability properties on DTMCs is as follows:

$$\text{minimize} \quad -\frac{1}{2}\, p_{s_{\text{init}}} + \sum_{s \in S} x_s \tag{2a}$$

such that

$$\forall s \in T: \quad p_s = x_s \tag{2b}$$

$$\forall s \in S \setminus T: \quad p_s \leq x_s \tag{2c}$$

$$\forall s \in S \setminus T: \quad p_s \leq \sum_{s' \in \text{succ}(s)} P(s,s') \cdot p_{s'} \tag{2d}$$

$$p_{s_{\text{init}}} > \lambda\ . \tag{2e}$$

The probability p_s of a state $s \in T$ is 1 iff the state is contained in the MCS, i. e., iff $x_s = 1$ (cf. line (2b)). Analogously, for every state $s \in S \setminus T$ that is not in the subsystem (i. e., $x_s = 0$), p_s is zero. This is achieved by requiring $p_s \leq x_s$ (line 2c). Note that for states in the critical subsystem, this does not restrict the value of p_s. An additional upper bound on the probability p_s is given by the weighted sum of the reachability probabilities $p_{s'}$ of the successor states s' (line 2d). The final constraint (line 2e) is as before. Constraints (2b)–(2e) together with the same objective function as in the SMT formulation (line 1a) yield an MCS. The objective function can be improved in two aspects. Since constraint (2d) only imposes an upper bound on p_s, we do not obtain—in contrast to the SMT formulation—the desired reachability probability as the value of $p_{s_{\text{init}}}$, but only a lower bound. Additionally it is desirable to obtain an MCS with maximal probability. Both can be achieved by maximizing the value of $p_{s_{\text{init}}}$. To that end, we add $p_{s_{\text{init}}}$ to the minimizing objective function with a negative coefficient. A factor $0 < c < 1$ is needed because, if we only subtract $p_{s_{\text{init}}}$, then the solver may add an additional state if this would yield $p_{s_{\text{init}}} = 1$. We choose $c = \frac{1}{2}$. This yields the objective function (2a).

Both the number of real and integer variables is in $O(\#_{\mathcal{D}}^{S})$ as well as the number of constraints. The number of non-zero variable coefficients in the MILP formulation is in $O(\#_{\mathcal{D}}^{S} + \#_{\mathcal{D}}^{T})$.

On the one hand, soundness of the MILP formulation assures that each satisfying assignment induces an MCS in that the probability to reach a target state from the initial state is maximal under all MCSs. On the other hand, completeness assures that for each MCS with maximal probability to reach the target set there is satisfying assignment inducing it. The proof of the following theorem can be found in Appendix B.

Theorem 4 *The MILP formulation* (2a)–(2e) *is sound and complete.*

Example 2 *Consider again the DTMC $\mathcal{D}$ in Figure 1(b) and the reachability property $\mathcal{P}_{\leq 0.6}(\Diamond\, a)$, which is violated by $\mathcal{D}$. The only state that satisfies the proposition a is s_3. Determining the relevant states for the label a yields the set $\{s_1, s_2, s_3, s_4, s_5, s_6\} = S \setminus \{s_7\}$. For the MILP formulation we can therefore ignore s_7 with all incident edges.*

We introduce the binary variables $x_{s_1}, \ldots, x_{s_6} \in \{0, 1\} \subseteq \mathbb{Z}$ and the real variables $p_{s_1}, \ldots, p_{s_6} \in [0, 1] \subseteq \mathbb{R}$. The MILP is then given by:

$$
\begin{array}{lll}
\textit{minimize} & -\frac{1}{2}p_{s_1} + x_{s_1} + x_{s_2} + x_{s_3} + x_{s_4} + x_{s_5} + x_{s_6} & \\
\textit{such that} & p_{s_3} = x_{s_3} & \\
& p_{s_1} \leq x_{s_1} & p_{s_1} \leq 0.7p_{s_2} + 0.3p_{s_4} \\
& p_{s_2} \leq x_{s_2} & p_{s_2} \leq 0.3p_{s_2} + 0.7p_{s_3} \\
& p_{s_4} \leq x_{s_4} & p_{s_4} \leq p_{s_5} \\
& p_{s_5} \leq x_{s_5} & p_{s_5} \leq 0.5p_{s_3} + 0.5p_{s_6} \\
& p_{s_6} \leq x_{s_6} & p_{s_6} \leq 0.5p_{s_4} \\
& p_{s_1} > 0.6\ . &
\end{array}
$$

Solving this MILP yields the following optimal satisfying assignment:

Variable	x_{s_1}	p_{s_1}	x_{s_2}	p_{s_2}	x_{s_3}	p_{s_3}	x_{s_4}	p_{s_4}	x_{s_5}	p_{s_5}	x_{s_6}	p_{s_6}
Value	1	0.7	1	1	1	1	0	0	0	0	0	0

This satisfying assignment corresponds to the MCS shown in Figure 1(c).

3.3. Optimizations

As our experiments revealed that the MILP formulation yields substantially shorter computation times compared to the SMT formulation, we will focus on the MILP approach in the remainder of this paper. We first consider several optimizations.

The optimizations consist of adding *redundant* constraints to the MILP formulation. These constraints are aimed to detect unsatisfiable or non-optimal branches in the search space at an early stage of the solving process. As we will show, they do not affect the correctness. Imposing extra constraints to the MILP formulations intuitively means adding cutting planes which cut off non-optimal solutions, tighten the LP-relaxation of the MILP, and may lead to better lower bounds on the optimal value which allow to prune parts of the search tree. All our constraints aim at guiding the MILP solver to only add states that are on paths from the initial state to a target state (in the MCS), as only such states will be part of an MCS.

3.3.1. Forward and Backward Constraints

We require that every non-target state has a successor state in the MCS. These constraints are called *forward cuts* (line 3a). Likewise, we add *backward cuts*, which enforce every state except s_{init} to have a predecessor in the MCS (line 3b). To avoid self-loops, we exclude a state itself from its successor and predecessor states.

$$\forall s \in S \setminus T: \quad -x_s + \sum_{s' \in \text{succ}(s) \setminus \{s\}} x_{s'} \geq 0 \tag{3a}$$

$$\forall s \in S \setminus \{s_{\text{init}}\}: \quad -x_s + \sum_{s' \in \text{pred}(s) \setminus \{s\}} x_{s'} \geq 0\ . \tag{3b}$$

These constraints are trivially satisfied if state s is not contained in the subsystem as x_s is 0. If state s is chosen (i. e., $x_s = 1$), then at least one successor/predecessor state s' must be contained (i. e., $x_{s'} = 1$) to achieve a positive number of successors/predecessors.

3.3.2. SCC Constraints

The forward/backward cuts do not precisely encode forward/backward reachability from the initial/target states: During the assignment process a connected subset of states could be selected even if its states are neither connected to the initial nor to any target state inside the subsystem. To partially remedy this situation, we utilize the SCC decomposition of the input DTMC. States of an SCC S' (not containing the initial state s_{init}) can be reached from outside S' through one of the input states $\text{In}(S')$ only. Therefore we ensure that a state of an SCC can only be selected if at least one of the SCC's input states is selected. The corresponding constraints are referred to as the *SCC input cuts* (line 4a). Analogously we define *SCC output cuts*: Paths from a state inside an SCC S' that does not contain a target state have to lead through one of the SCC's output states $\text{Out}(S')$ to reach a target state. Therefore, if no output state of an SCC S' is selected, we do not select any state of the SCC (line 4b). Note that these SCC cuts do not enforce that the subsystem corresponding to a satisfying assignment contains only states on a path from the initial state to target states. This is still only ensured by minimizing the objective function.

$$\forall \text{SCC } S', s_{\text{init}} \notin S' \; \forall s \in S' \setminus \text{In}(S') : \quad x_s \;\leq \sum_{s' \in \text{In}(S')} x_{s'} \tag{4a}$$

$$\forall \text{SCC } S', S' \cap T = \emptyset \; \forall s \in S' : \quad x_s \;\leq \sum_{s' \in \text{Out}(S')} x_{s'} \; . \tag{4b}$$

3.3.3. Reachability Constraints

If an SCC is selected which is connected to the initial state and to one of the target states, nevertheless an isolated loop inside the SCC may be selected. We now present a set of constraints which precisely encode reachability. An assignment will satisfy these additional constraints only if all selected states lie on a path from the initial to a target state. Without these constraints, this is only ensured by the state-minimality as forced by the objective function. We introduce the notions of *forward* and *backward reachability*. For the encoding of forward reachability, we use a variable $r^{\rightarrow}_s \in [0,1] \subseteq \mathbb{R}$ for each state s except the initial state. These variables define a partial order on the states. For each transition $(s,s') \in E_{\mathcal{D}}$ ($s' \neq s_{\text{init}}$) we introduce a characteristic integer variable $t^{\rightarrow}_{s,s'} \in [0,1] \subseteq \mathbb{Z}$. The constraints for forward reachability are as follows:

$$\forall s' \in S \setminus \{s_{\text{init}}\} \; \forall s \in \text{pred}(s') : \quad t^{\rightarrow}_{s,s'} \leq x_s \tag{5a}$$

$$\forall s' \in S \setminus \{s_{\text{init}}\} \; \forall s \in \text{pred}(s') : \quad r^{\rightarrow}_s < r^{\rightarrow}_{s'} + (1 - t^{\rightarrow}_{s,s'}) \tag{5b}$$

$$\forall s' \in S \setminus \{s_{\text{init}}\} : \quad \sum_{s \in \text{pred}(s')} t^{\rightarrow}_{s,s'} = x_{s'} \; . \tag{5c}$$

If $s \in S$ is selected and reachable from s_{init} then there is a loop-free path $s_{\text{init}} = s_0 \dots s_n = s$ such that $r^{\rightarrow}_{s_i} < r^{\rightarrow}_{s_{i+1}}$ for all $0 \leq i < n$ and all states on the path are selected, i. e., $x_{s_i} = 1$ for all $0 \leq i \leq n$. This is reflected in the constraints: Each transition $(s,s') \in E_{\mathcal{D}}$ with $t^{\rightarrow}_{s,s'} = 1$ emanates from a selected state s (line 5a). If $x_{s'} = 0$ then constraint (5c) ensures that all variables $t_{s,s'}$ equal 0 for $s \in \text{pred}(s')$, i. e., $t_{s,s'} = 1$ implies $x_{s'} = 1$. Therefore $t_{s,s'} = 1$ implies that both s and s' are contained in the subsystem.

If $t^{\rightarrow}_{s,s'} = 1$, $r^{\rightarrow}_s < r^{\rightarrow}_{s'}$ has to hold (line 5b), which defines the partial order on selected states. The constraints defined in line (5c) imply that from each selected state s' not being the initial state, one incoming transition $t^{\rightarrow}_{s,s'}$ has to be selected. One can show by induction that this ensures that for each selected state s there is a path in the subsystem from s_{init} to s.

The constraints defining backward reachability from the target states are built analogously with variables $r^{\leftarrow}_s$ for all states $s \in S$ and variables $t^{\leftarrow}_{s,s'}$ for all transitions $(s,s') \in E_{\mathcal{D}}$ ($s \notin T$):

$$\forall s \in S \setminus T \; \forall s' \in \text{succ}(s) : \quad t^{\leftarrow}_{s,s'} \leq x_{s'} \tag{6a}$$

$$\forall s \in S \setminus T \; \forall s' \in \text{succ}(s) : \quad r^{\leftarrow}_s < r^{\leftarrow}_{s'} + (1 - t^{\leftarrow}_{s,s'}) \tag{6b}$$

$$\forall s \in S \setminus T: \quad \sum_{s' \in \mathrm{succ}(s)} t^{\leftarrow}_{s,s'} = x_s \ . \tag{6c}$$

In the assignment process, the forward and backward reachability constraints eliminate all critical subsystems with unreachable states. However, as there are additional variables for all states and for all transitions, the usage of these cuts is expensive, as we discuss in detail when presenting the experiments in Section 5. For MDPs (cf. Section 4), the backward reachability constraints are not only used as optimizations, but they are needed for correctness.

The following theorem, whose proof is given in Appendix C, states soundness and completeness of the optimization constraints. Soundness means that any satisfying assignment of the SMT or MILP formulation with optimization constraints induces an MCS. Completeness means that for both the SMT and MILP formulations with optimization constraints, for each MCS[5] there is a satisfying assignment inducing it.

Theorem 5 *Both the SMT formulation* (1a)–(1d) *and the MILP formulation* (2a)–(2e) *together with any (combination) of the three above optimizations are sound and complete.*

3.4. ω-Regular Properties

We now generalize our MILP formulation to arbitrary ω-regular properties. Let $\mathcal{L}$ be such a property and $\mathcal{A}$ a DRA with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. As before, we want to compute an MCS $\mathcal{D}'$ for which $\Pr^{s_{\text{init}}}_{\mathcal{D}'}(\mathcal{L}) > \lambda$ holds. We follow the model-checking algorithm for ω-regular properties on DTMCs as described in Section 2.3.2. We consider the product $\mathcal{D} \otimes \mathcal{A}$ of the DTMC $\mathcal{D}$ and the DRA $\mathcal{A}$ with distribution function $P_{\mathcal{D}\otimes\mathcal{A}}$ as in Definition 8 and assume (as before) that all irrelevant states have been removed. Let $T_1, \ldots, T_n$ be the accepting BSCCs of $\mathcal{D} \otimes \mathcal{A}$ and $T = \bigcup_{i=1}^{n} T_i$. We introduce characteristic variables $x_{T_i} \in \{0,1\} \subseteq \mathbb{Z}$ for all $T_1, \ldots, T_n$ and $x_s \in \{0,1\} \subseteq \mathbb{Z}$ for all states $s \in S$. We emphasize that the x_s variables are not defined for every state of $\mathcal{D} \otimes \mathcal{A}$, but for all states of the DTMC $\mathcal{D}$. This corresponds to our aim to obtain an MCS of $\mathcal{D}$. As the reachability probabilities for all states of the product automaton are needed, we use a variable $p_{(s,q)}$ for every state $(s,q) \in S \times Q$. We obtain:

$$\text{minimize} \quad -\frac{1}{2}\, p_{(s,q)_{\text{init}}} + \sum_{s \in S} x_s \tag{7a}$$

such that

$$\forall i = 1, \ldots, n \ \forall (s,q) \in T_i: \quad p_{(s,q)} = x_{T_i} \tag{7b}$$

$$\forall i = 1, \ldots, n \ \forall (s,q) \in T_i: \quad x_s \geq x_{T_i} \tag{7c}$$

$$\forall (s,q) \in S_{\mathcal{D}\otimes\mathcal{A}} \setminus T: \quad p_{(s,q)} \leq x_s \tag{7d}$$

$$\forall (s,q) \in S_{\mathcal{D}\otimes\mathcal{A}} \setminus T: \quad p_{(s,q)} \leq \sum_{(s',q') \in \mathrm{succ}_{\mathcal{D}\otimes\mathcal{A}}((s,q))} P_{\mathcal{D}\otimes\mathcal{A}}\big((s,q),(s',q')\big) \cdot p_{(s',q')} \tag{7e}$$

$$p_{(s,q)_{\text{init}}} > \lambda \ . \tag{7f}$$

Intuitively, the probability of a state in an accepting BSCC of $\mathcal{D} \otimes \mathcal{A}$ is one iff that BSCC is selected (line 7b). A BSCC can only be selected if (the projections of) all of its states on $\mathcal{D}$ are selected (line 7c). If the probability contribution of a state (s,q) exceeds 0, the DTMC-state s is selected (line 7d). Using constraint (7e), the probability of reaching accepting BSCCs inside the MCS is computed. This constraint is similar as in the initial MILP formulation for reachability probabilities.

The MILP formulation contains $O(\#^S_{\mathcal{D}\otimes\mathcal{A}})$ real variables, $O(\#^S_{\mathcal{D}})$ integer variables, $O(\#^S_{\mathcal{D}\otimes\mathcal{A}})$ constraints and $O(\#^S_{\mathcal{D}\otimes\mathcal{A}} + \#^T_{\mathcal{D}\otimes\mathcal{A}})$ non-zero coefficients.

The following theorem states soundness and completeness of the MILP formulation for ω-regular properties. Soundness means that the satisfying assignments induce MCSs, completeness expresses that each MCS with

[5] For MILP only those MCSs with maximal probability to reach target states under all MCSs.

maximal probability to satisfy the given property under all MCSs is induced by some satisfying assignment. The proof can be found in Appendix D.

Theorem 6 *The MILP formulation* (7a)–(7f) *is sound and complete.*

Remark 1 If we treat a reachability property $\varphi = \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ as an ω-regular property and use the MILP (7a)–(7f) instead of (2a)–(2e) to generate an MCS for φ, we obtain different results. While the latter formulation only preserves finite paths ending in a target state (which is sufficient for reachability properties), the former preserves infinite paths, which is necessary for general ω-regular properties. However, if the validity of a property can be certified by a finite path (which holds for the subclass of *regular* linear-time properties, including reachability), it suffices to preserve the corresponding finite paths. In this case, a deterministic finite automaton (DFA) can be used to represent the property instead of a DRA. A similar construction as for general ω-regular properties can be used to obtain a MCS, preserving only finite paths.

4. Minimal Critical Subsystems for MDPs

In this section we extend the approaches described in Section 3 to find MCSs for MDPs. This task is more complicated than for DTMCs as we additionally have to find a scheduler which yields a critical subsystem of minimum size. As before, we start by considering reachability probabilities and then treat ω-regular properties.

4.1. Reachability Properties

Whereas the theoretical complexity of computing MCSs for reachability properties of DTMCs is (to our knowledge) unknown[6], for MDPs the following theorem holds, whose proof can be found in Appendix E:

Theorem 7 ([5]) *Let $\mathcal{M}$ be an MDP with $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ and $k \in \mathbb{N}$. The problem to decide whether there exists a critical subsystem of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with at most k states is NP-complete.*

Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP, $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ a property violated by $\mathcal{M}$ and $T = \big\{s \in S \;\big|\; a \in L(s)\big\} \subseteq S$ the set of target states. We assume that all irrelevant states for a (and their adjacent edges) have been removed from $\mathcal{M}$.

It is easy to see that there is a DTMC under the MCSs of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$: Assume an MDP $\mathcal{M}'$ that is an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. Since $\mathcal{M}'$ is critical, it violates the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. Then there is a memoryless deterministic scheduler inducing a DTMC $\mathcal{D}'$ with a probability mass exceeding λ. Furthermore, since $\mathcal{M}'$ is minimal and $\mathcal{D}'$ is a subsystem of $\mathcal{M}'$, also $\mathcal{D}'$ is minimal.

To encode such a scheduler, we use a binary variable $\sigma_{s,\alpha} \in \{0,1\} \subseteq \mathbb{Z}$ for each state $s \in S \setminus T$ and each action $\alpha \in Act$ such that $\sigma_{s,\alpha} = 1$ iff action α is selected in state s by the scheduler under consideration. Like for DTMCs, we use a binary characteristic variable $x_s \in \{0,1\} \subseteq \mathbb{Z}$ for each state $s \in S$ to encode whether s belongs to the subsystem or not, and a real-valued variable $p_s \in [0,1] \subseteq \mathbb{R}$ to encode the reachability probability under the given scheduler (determined by the variables $\sigma_{s,\alpha}$) within the selected subsystem (determined by the variables x_s).

The core MILP formulation (i. e., the formulation without any optimizations) for reachability properties of MDPs is more complicated than for DTMCs. This is due to the fact that the reachability of target states in MDP subsystems does not exclusively depend on the states but also on the actions of the subsystem. Recall that a state $s \in S$ is irrelevant if there is no scheduler yielding T to be reachable from s. However, for a relevant state s, T might be reachable under some schedulers and might not be reachable under others. We therefore impose additional constraints to assure that we consider only schedulers under which the target state set is reachable from all subsystem states. Note that these constraints are not optional: The

[6]The problem of finding an MCS for a PCTL-formula on DTMCs is NP-complete [5]. This result, however, exploits nested PCTL-formulae.

reachability properties are encoded based on backward reachability from the target states. Without these additional constraints, the reachability probabilities for states in a bottom SCC of the induced DTMC could be incorrectly determined to be 1 if it does not contain a target state, leading to wrong results. Let

$$S_{\mathcal{M}}^{\mathrm{probl}(a)} = \{s \in S \mid \exists \sigma \in \mathrm{Sched}_{\mathcal{M}} : \Pr_{\mathcal{M}^\sigma}^{s}(\Diamond\, a) = 0\}$$

be the set of *problematic* states in MDP $\mathcal{M}$ for proposition a. If $s \notin S_{\mathcal{M}}^{\mathrm{probl}(a)}$ then s is called *unproblematic* for a.

Example 3 *To illustrate the issue with problematic states, consider again the MDP in Figure 1(a). States s_2 and s_5 are both problematic since the scheduler which selects α in both s_2 and s_5 prevents reaching the target state s_3. We cannot remove the outgoing transitions belonging to action α in a preprocessing step since a scheduler may choose α in one state and β in the other one. However, if a scheduler chooses α in both states, we obtain the following constraints:*

$$\begin{aligned} p_{s_2} &\leq 1.0 \cdot p_{s_5} \\ p_{s_5} &\leq 1.0 \cdot p_{s_2}\ . \end{aligned}$$

A solution is $p_{s_2} = p_{s_5} = 1$ and thereby the maximum probability for reaching a target state can be assigned to both states, although s_3 is not reachable under this scheduler.

Our additional constraints prevent from obtaining a scheduler that chooses the "wrong" actions in problematic states (i. e., actions that yield the T states in the MCS to be unreachable) by requiring that such states are backward reachable from some unproblematic state. These MILP constraints are defined in a similar way to the backward reachability constraints (6a)–(6c) for DTMCs. Let $Act_{\mathcal{M}}^{\mathrm{probl}(a)} = \left\{(s,\alpha) \in S \times Act \,\middle|\, \mathrm{succ}_{\mathcal{M}}(s,\alpha) \subseteq S_{\mathcal{M}}^{\mathrm{probl}(a)}\right\}$ be the set of state-action pairs such that selecting α in s yields a problematic state (for a). We use a real-valued variable $r_s^{\leftarrow} \in [0,1] \subseteq \mathbb{R}$ for each problematic state $s \in S_{\mathcal{M}}^{\mathrm{probl}(a)}$ that defines a partial order on the problematic states (for a). The binary variables $t_{s,s'}^{\leftarrow} \in \{0,1\} \subseteq \mathbb{Z}$ are used to indicate the existence of an edge in the MCS between states s and $s' \in S_{\mathcal{M}}^{\mathrm{probl}(a)}$ where $(s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)}$ for an action $\alpha \in Act$. We thus propose the following MILP formulation:

$$\text{minimize} \quad -\frac{1}{2}\, p_{s_{\mathrm{init}}} + \sum_{s \in S} x_s \tag{8a}$$

such that

$$p_{s_{\mathrm{init}}} > \lambda \tag{8b}$$

$$\forall s \in T: \quad p_s = x_s \tag{8c}$$

$$\forall s \in S \setminus T: \quad p_s \leq x_s \tag{8d}$$

$$\forall s \in S \setminus T: \quad \sum_{\alpha \in Act} \sigma_{s,\alpha} = x_s \tag{8e}$$

$$\forall s \in S \setminus T\ \forall \alpha \in Act: \quad p_s \leq (1 - \sigma_{s,\alpha}) + \sum_{s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha)} P(s,\alpha,s') \cdot p_{s'} \tag{8f}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)}\ \forall s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha): \quad t_{s,s'}^{\leftarrow} \leq x_{s'} \tag{8g}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)}\ \forall s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha): \quad r_s^{\leftarrow} < r_{s'}^{\leftarrow} + (1 - t_{s,s'}^{\leftarrow}) \tag{8h}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)}: \quad (1 - \sigma_{s,\alpha}) + \sum_{s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha)} t_{s,s'}^{\leftarrow} \geq x_s\ . \tag{8i}$$

The constraints (8a)–(8d) are the same as for DTMCs. Eq. (8e) ensures that in each selected non-target state a single action is selected by the scheduler. Line (8f) corresponds to line (2d) of the MILP for DTMCs.

The only change is that if the action α, to which the constraint belongs, is not selected by the scheduler, i. e., if $\sigma_{s,\alpha} = 0$, then the constraint is automatically satisfied due to the term $(1 - \sigma_{s,\alpha})$. The following three constraints (8g)–(8i) ensure for each problematic state the backward reachability from an unproblematic state.

The number of real variables of the MILP is in $O(\#^S_{\mathcal{M}})$, the number of integer variables and the number of non-zero coefficients in $O(\#^S_{\mathcal{M}} + \#^T_{\mathcal{M}})$.

The following theorem formalizes the one-to-one correspondence between the satisfying assignments of the MILP formulation and those MCSs in which the probability to satisfy the considered property is maximal under all MCSs. The proof is provided in Appendix F.

Theorem 8 *The MILP formulation* (8a)–(8i) *is sound and complete.*

In addition, our MILP formulation yields a memoryless deterministic scheduler σ such that the reachability probability of $\Diamond\, a$ in the DTMC induced by σ on the MCS exceeds λ. The optimizations for DTMCs in Section 3.3 can, with the exception of the SCC cuts, be directly transferred to MDPs. For the sake of brevity, we omit the details here.

Example 4 *Consider the MDP $\mathcal{M}$ shown in Figure 1(a) with the violated property $\mathcal{P}_{\leq 0.6}(\Diamond\, a)$. State s_7 is irrelevant as the only target state s_3 is unreachable from s_7. States $S^{probl(a)}_{\mathcal{M}} = \{s_1, s_2, s_4, s_5\}$ are problematic, as choosing action α in both s_2 and s_5 makes s_3 unreachable from these states, see Example 3. The problematic actions are $Act^{probl(a)}_{\mathcal{M}} = \{(s_1, \alpha), (s_2, \alpha), (s_4, \alpha), (s_5, \alpha)\}$. State s_6 is not problematic as the irrelevant state s_7 is reached with probability > 0 under all schedulers.*

For the MILP formulation we introduce the following variables: decision variables $x_{s_1}, \ldots, x_{s_6} \in \{0, 1\} \subseteq \mathbb{Z}$, probability variables $p_{s_1}, \ldots, p_{s_6} \in [0, 1] \subseteq \mathbb{R}$, scheduler variables $\sigma_{s_1,\alpha}$, $\sigma_{s_2,\alpha}$, $\sigma_{s_2,\beta}$, $\sigma_{s_4,\alpha}$, $\sigma_{s_5,\alpha}$, $\sigma_{s_5,\beta}$, $\sigma_{s_6,\alpha} \in \{0, 1\} \subseteq \mathbb{Z}$. For the reachability constraints (8g)–(8i) *we need the variables $r^{\leftarrow}_{s_1}, r^{\leftarrow}_{s_2}, r^{\leftarrow}_{s_4}, r^{\leftarrow}_{s_5} \in [0, 1] \subseteq \mathbb{R}$ and $t^{\leftarrow}_{s_1,s_2}, t^{\leftarrow}_{s_1,s_4}, t^{\leftarrow}_{s_2,s_5}, t^{\leftarrow}_{s_4,s_5}, t^{\leftarrow}_{s_5,s_2} \in [0, 1] \subseteq \mathbb{R}$.*

$$
\begin{aligned}
&\textit{minimize} \quad -\tfrac{1}{2} p_{s_1} + x_{s_1} + x_{s_2} + x_{s_3} + x_{s_4} + x_{s_5} + x_{s_6} \\
&\textit{such that} \\
&\quad p_{s_1} > 0.6 \\
&\quad p_{s_3} = x_{s_3} \\
&\quad p_{s_1} \leq x_{s_3} \qquad \sigma_{s_1,\alpha} = x_{s_1} \qquad p_{s_1} \leq (1 - \sigma_{s_1,\alpha}) + 0.7 p_{s_2} + 0.3 p_{s_4} \\
&\quad p_{s_2} \leq x_{s_2} \qquad \sigma_{s_1,\alpha} + \sigma_{s_1,\beta} = x_{s_2} \qquad p_{s_2} \leq (1 - \sigma_{s_2,\alpha}) + p_{s_5} \\
&\qquad\qquad\qquad\qquad\qquad\quad\; p_{s_2} \leq (1 - \sigma_{s_2,\beta}) + 0.7 p_3 + 0.3 p_2 \\
&\quad p_{s_4} \leq x_{s_4} \qquad \sigma_{s_4,\alpha} = x_{s_4} \qquad p_{s_4} \leq (1 - \sigma_{s_4,\alpha}) + p_{s_5} \\
&\quad p_{s_5} \leq x_{s_5} \qquad \sigma_{s_5,\alpha} + \sigma_{s_5,\beta} = x_{s_5} \qquad p_{s_5} \leq (1 - \sigma_{s_5,\alpha}) + p_{s_2} \\
&\qquad\qquad\qquad\qquad\qquad\quad\; p_{s_5} \leq (1 - \sigma_{s_5,\beta}) + 0.5 p_{s_3} + 0.5 p_{s_6} \\
&\quad p_{s_6} \leq x_{s_6} \qquad \sigma_{s_6,\alpha} = x_{s_6} \qquad p_{s_6} \leq (1 - \sigma_{s_6,\alpha}) + 0.5 p_{s_4} \\
&\qquad \underbrace{\qquad\qquad}_{\text{(8b)–(8d)}} \qquad \underbrace{\qquad\qquad}_{\text{(8e)}} \qquad \underbrace{\qquad\qquad\qquad\qquad}_{\text{(8f)}} \\
&\quad t^{\leftarrow}_{s_1,s_2} \leq x_{s_2} \qquad r^{\leftarrow}_{s_1} < r^{\leftarrow}_{s_2} + (1 - t^{\leftarrow}_{s_1,s_2}) \qquad (1 - \sigma_{s_1,\alpha}) + t^{\leftarrow}_{s_1,s_2} + t^{\leftarrow}_{s_1,s_4} \geq x_{s_1} \\
&\quad t^{\leftarrow}_{s_1,s_4} \leq x_{s_4} \qquad r^{\leftarrow}_{s_1} < r^{\leftarrow}_{s_4} + (1 - t^{\leftarrow}_{s_1,s_4}) \\
&\quad t^{\leftarrow}_{s_2,s_5} \leq x_{s_5} \qquad r^{\leftarrow}_{s_2} < r^{\leftarrow}_{s_5} + (1 - t^{\leftarrow}_{s_2,s_5}) \qquad (1 - \sigma_{s_2,\alpha}) + t^{\leftarrow}_{s_2,s_5} \geq x_{s_2} \\
&\quad t^{\leftarrow}_{s_4,s_5} \leq x_{s_5} \qquad r^{\leftarrow}_{s_4} < r^{\leftarrow}_{s_5} + (1 - t^{\leftarrow}_{s_4,s_5}) \qquad (1 - \sigma_{s_4,\alpha}) + t^{\leftarrow}_{s_4,s_5} \geq x_{s_4} \\
&\quad t^{\leftarrow}_{s_5,s_2} \leq x_{s_2} \qquad r^{\leftarrow}_{s_5} < r^{\leftarrow}_{s_2} + (1 - t^{\leftarrow}_{s_5,s_2}) \qquad (1 - \sigma_{s_5,\alpha}) + t^{\leftarrow}_{s_5,s_2} \geq x_{s_2} \\
&\qquad \underbrace{\qquad\qquad}_{\text{(8g)}} \qquad \underbrace{\qquad\qquad}_{\text{(8h)}} \qquad \underbrace{\qquad\qquad\qquad\qquad}_{\text{(8i)}}
\end{aligned}
$$

Solving this MILP yields the following optimal variable assignment:

Variable	x_{s_1}	p_{s_1}	x_{s_2}	p_{s_2}	x_{s_3}	p_{s_3}	x_{s_4}	p_{s_4}	x_{s_5}	p_{s_5}	x_{s_6}	p_{s_6}
Value	1	0.7	1	1	1	1	0	0	0	0	0	0

Variable	$\sigma_{s_1,\alpha}$	$\sigma_{s_2,\alpha}$	$\sigma_{s_2,\beta}$	$\sigma_{s_4,\alpha}$	$\sigma_{s_5,\alpha}$	$\sigma_{s_5,\beta}$	$\sigma_{s_6,\alpha}$
Value	1	0	1	0	0	0	0

Variable	$t^{\leftarrow}_{s_1,s_2}$	$t^{\leftarrow}_{s_1,s_4}$	$t^{\leftarrow}_{s_2,s_5}$	$t^{\leftarrow}_{s_4,s_5}$	$t^{\leftarrow}_{s_5,s_2}$	$r^{\leftarrow}_{s_1}$	$r^{\leftarrow}_{s_2}$	$r^{\leftarrow}_{s_4}$	$r^{\leftarrow}_{s_5}$
Value	1	0	0	0	0	0	1	0	0

The resulting subsystem that corresponds to this variable assignment contains the states $\{s_1, s_2, s_3\}$. It is shown in Figure 1(c).

4.2. ω-Regular Properties

Determining MCSs for ω-regular properties of MDPs is more involved than for DTMCs, as we need to know the set of accepting end components of the product MDP. Their number can be exponential in the size of the MDP. Instead of computing them in a pre-processing step (as we did for BSCCs in the DTMC setting), we pursue a different way: We encode the state sets that almost surely satisfy the ω-regular property directly into the MILP and use these state sets as target states.

Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP and $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA with $F = \big\{(R_i, A_i) \,\big|\, i = 1, \ldots, n\big\}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ for an ω-regular property $\mathcal{L}$. We assume that $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\mathcal{L})$ and that $\mathcal{M} \otimes \mathcal{A}$ has no irrelevant states. To determine the relevant states of $\mathcal{M} \otimes \mathcal{A}$, we compute its *maximal* end components. This can be done efficiently [53]. States from which a maximal end component containing a state in $\bigcup_{i=1}^{n} A_i$ is reachable under at least one scheduler, are relevant.[7]

To simplify notation we use $U = S \times Q$, $u = (s, q)$, and $u' = (s', q')$. We have variables $x_s \in \{0, 1\} \subseteq \mathbb{Z}$ for all $s \in S$ indicating whether a state of the original MDP is contained in the subsystem and $p_u \in [0, 1] \subseteq \mathbb{R}$ which stores the probability of satisfying the property within the subsystem. The variables $\sigma_{u,\alpha} \in \{0, 1\} \subseteq \mathbb{Z}$ for $u \in U$ and $\alpha \in Act$ store the selected scheduler. Please note that, as deterministic memoryless schedulers on the product-MDP suffice for ω-regular properties, this encoding suffices. The identification of the set of target states is based on the following lemma, whose proof can be found in Appendix G:

Lemma 1 *Let $(R_i, A_i) \in 2^Q \times 2^Q$ be a pair of a Rabin acceptance condition, $\sigma : U \to Act$ a scheduler, and $M_i \subseteq U$ a set of states with the following properties:*

1. $\forall u \in M_i : \sum_{u' \in \text{succ}(u,\sigma(u)) \cap M_i} P'(u, \sigma(u), u') = 1$,
2. $M_i \cap (S \times R_i) = \emptyset$, *and*
3. *for each state $u \in M_i$ there is a path from u to a state in $S \times A_i$.*

Then the probability of satisfying the acceptance condition F in $\mathcal{M}$ because of the pair (R_i, A_i) is 1 for all $u \in M_i$.

For each $(R_i, A_i) \in F$ and $u \in U$ we introduce a characteristic variable $m^i_u \in \{0, 1\} \subseteq \mathbb{Z}$ where $m^i_u = 1$ iff state u is contained in set M_i. For satisfying the third condition of Lemma 1, we need to ensure backward reachability from A_i and use variables $t^i_{u,u'} \in \{0, 1\} \subseteq \mathbb{Z}$ for all $(u, u') \in E_{\mathcal{M}\otimes\mathcal{A}}$ and $r_u \in [0, 1] \subseteq \mathbb{R}$ for all states $u \in U$. Recall that $E_{\mathcal{M}\otimes\mathcal{A}}$ is the set $E_{\mathcal{M}\otimes\mathcal{A}} = \{(u, u') \in U \times U \,|\, u' \in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u)\}$. Let $n_{u,\alpha} = |\text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u, \alpha)|$ denote the number of successor states of u under action α.

[7] Strictly speaking, this condition is not sufficient since end components additionally have to satisfy a condition on the R_i states to be accepting. However, exactly identifying the relevant states would require to determine all end components, which is in general computationally infeasible. Therefore we resort to an over-approximation of the relevant states. Since we explicitly add reachability constraints, this does not affect the correctness (as we will show).

The MILP for computing a minimal critical subsystem of $\mathcal{M}$ such that $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ is violated is shown below.

$$\text{minimize} \quad -\frac{1}{2}\, p_{(s,q)_{\text{init}}} + \sum_{s\in S} x_s \tag{9a}$$

such that

• selection of at most one action per state:

$$\forall u=(s,q)\in U: \quad \sum_{\alpha\in Act} \sigma_{u,\alpha} \leq x_s \tag{9b}$$

• for all $i=1,\ldots,n$ the definition of set M_i (closure w. r. t. $\text{succ}(u,\alpha)$ for $\alpha\in Act$):

$$\forall u\in U\ \forall \alpha\in Act \text{ with } \sum_{u'\in U} P'(u,\alpha,u') < 1: \quad m_u^i \leq 1-\sigma_{u,\alpha} \tag{9c}$$

$$\forall u\in U\ \forall \alpha\in Act: \quad n_{u,\alpha}\cdot(2-\sigma_{u,\alpha}-m_u^i) + \sum_{u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} m_{u'}^i \ \geq n_{u,\alpha} \tag{9d}$$

$$\forall u\in S\times R_i: \quad m_u^i = 0 \tag{9e}$$

• for all $i=1,\ldots,n$ backward reachability of $S\times A_i$ within M_i:

$$\forall u\in U\ \forall\alpha\in Act\ \forall u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha): \quad t_{u,u'}^i \leq m_{u'}^i + (1-\sigma_{u,\alpha}) \tag{9f}$$

$$\forall u\in U\ \forall\alpha\in Act\ \forall u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha): \quad r_u^i < r_{u'}^i + (1-t_{u,u'}^i) + (1-\sigma_{u,\alpha}) \tag{9g}$$

$$\forall u\in S\times(Q\setminus A_i)\ \forall\alpha\in Act: \quad (1-\sigma_{u,\alpha}) + \sum_{u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} t_{u,u'}^i \geq m_u^i \tag{9h}$$

• probability computation:

$$p_{(s,q)_{\text{init}}} > \lambda \tag{9i}$$

$$\forall i=1,\ldots,n\ \forall u\in U: \quad p_u \geq m_u^i \tag{9j}$$

$$\forall u\in U: \quad p_u \leq \sum_{\alpha\in Act}\sigma_{u,\alpha} \tag{9k}$$

$$\forall u\in U\ \forall\alpha\in Act: \quad p_u \leq (1-\sigma_{u,\alpha}) + \sum_{i=1}^{n} m_u^i + \sum_{u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} P(u,\alpha,u')\cdot p_{u'} \tag{9l}$$

• backward reachability of $M=\bigcup_{i=1}^n M_i$ within the subsystem:

$$\forall u=(s,q)\in U\ \forall\alpha\in Act\ \forall u'=(s',q')\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}\big(u,\alpha\big): \quad t_{u,u'}^M \leq x_{s'} + (1-\sigma_{u,\alpha}) \tag{9m}$$

$$\forall u\in U\ \forall\alpha\in Act\ \forall u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha): \quad r_u^M < r_{u'}^M + (1-t_{u,u'}^M) + (1-\sigma_{u,\alpha}) \tag{9n}$$

$$\forall u=(s,q)\in U\ \forall\alpha\in Act: \quad (1-\sigma_{u,\alpha}) + \sum_{i=1}^{n} m_u^i + \sum_{u'\in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}\left(u,\alpha\right)} t_{u,u'}^M \geq x_s\ . \tag{9o}$$

The target function is defined as before. Constraint (9b) defines a valid scheduler by ensuring that for each selected state at most one action is chosen. The reason for selecting at most one action is the following: If a subsystem $S'\subseteq S$ of the MDP is selected, we select the subsystem $S'\times Q$ of the product automaton.

By this it is not guaranteed that in the DTMC induced by the scheduler of the product automaton from each state (s, q) an accepting BSCC is reachable. Since we later require that from each state in $S' \times Q$ an accepting BSCC is reachable under the selected action, we solve this problem by allowing not to select an action. If no action is chosen, (9k) ensures that the probability p_u is zero.

The next step is to define the sets M_i $(i = 1, \ldots, n)$ according to Lemma 1. The first condition, i. e., that for each $u \in M_i$ the probability of staying in M_i is 1, is ensured in two steps: First we forbid in (9c) that a state u is in M_i if under the selected action the sum of the probabilities of the out-going edges is less than one. Note that for each state in M_i at least one out-going action is selected, since the probability of states without selected action is zero, but (9j) sets the probability of M_i-states to one.

Second we ensure in (9d) the closure of M_i under successors. If state u belongs to M_i (i. e., $m_u^i = 1$) and action α is chosen by the scheduler (i. e., $\sigma_{u,\alpha} = 1$), all successors of u w. r. t. action α have to belong to M_i. The term $n_{u,\alpha}(2 - \sigma_{u,\alpha} - m_u^i)$ is zero iff α is selected in u and $u \in M_i$. In this case the sum over the corresponding variables $m_{u'}^i$ of the successors u' of u has to be at least the number of the successors of u.

Eq. (9e) ensures that M_i does not contain an R_i state (second condition of Lemma 1).

In order to ensure backward reachability from $S \times A_i$ within M_i, we use the constraints known from the DTMC optimizations and MDP reachability properties (cf. Section 3.3.3). The corresponding constraints are given in (9f)–(9h). These constraints are defined separately for all sets $(R_i, A_i) \in F$. They ensure that, under the chosen scheduler, from each state in M_i an A_i-state is reachable, as requested in the third condition of Lemma 1. They are satisfied for a set M_i that contains accepting BSCCs of the induced DTMC, which are reachable from all states in M_i. If no element of $S \times A_i$ is contained, no partial order on the states can be defined by (9f)–(9h) (see also Appendix C).

The remaining constraints are analogous to the MILP for reachability properties: Constraint (9i) ensures criticality of the subsystem. Constraints (9j), (9k), and (9b) force the states of the sets M_i (i. e., target states) to be included in the subsystem and to have probability 1:

$$\forall u = (s, q) \in U : \quad m_u^i \overset{(9j)}{\leq} p_u \overset{(9k)}{\leq} \sum_{\alpha \in Act} \sigma_{u,\alpha} \overset{(9b)}{\leq} x_s \ .$$

Constraints (9k) and (9b) assign probability 0 to all states not in the subsystem and (9l) computes the probability of reaching a state in M_i for all remaining states. Since we do not know the target states in advance, we have to ensure that (9l) is also satisfied for target states. This is the case due to the expression $\sum_{i=1}^{n} m_u^i$ which is at least 1 if u is a target state.

The last three constraints are again backward reachability constraints, analogous to the reachability constraints for problematic states in the case of reachability properties. They ensure that from each state with a selected action in the subsystem an M_i state is reachable with non-zero probability.

The following theorem states that the solutions of the MILP formulation (9a)–(9o) encode MCSs and vice versa, each MCS with maximal probability to satisfy the property under all MCSs is encoded by a solution. The proof is given in Appendix G.

Theorem 9 *The MILP formulation* (9a)–(9o) *is sound and complete.*

A proof of this theorem can be found in Appendix G.

The number of integer variables in the MILP, its number of constraints, and the number of non-zero coefficients are in $O(n \cdot (\#^S_{\mathcal{M}\otimes\mathcal{A}} + \#^T_{\mathcal{M}\otimes\mathcal{A}}))$, while the number of real variables is in $O(n \cdot \#^S_{\mathcal{M}\otimes\mathcal{A}})$, where n is the number of acceptance pairs of $\mathcal{A}$.

A remark on the result of the MCS computation is in order. Whereas for reachability properties, the result of our MILP formulation is a DTMC (in fact, an MDP plus a memoryless deterministic scheduler on this MDP) this is not the case for ω-regular properties. Instead our MILP formulation yields a DTMC as substructure of the product $\mathcal{M} \otimes \mathcal{A}$. Projecting this onto the MDP $\mathcal{M}$ however yields (in general) an MDP, as e. g., states of the form (s, q) and (s, q') are projected onto the state s in $\mathcal{M}$ but may have different outgoing distributions.

5. Experimental Evaluation

5.1. Implementation and Experimental Setup

To demonstrate the feasibility of our approaches, we implemented the algorithms described in the previous sections using C++ in a tool named `LTLSubsys`. It supports the generation of MCSs for DTMCs and MDPs with LTL properties [54]. LTL is a popular specification language for linear-time properties, which form a subclass of ω-regular properties, including reachability. We use the symbols $\Box$ for "globally", $\Diamond$ for "finally", and $\bigcirc$ for "next". E. g., $\Box \Diamond \varphi$ holds if infinitely often φ holds at some time point in the future.

We use `Gurobi` version 5.6 [50] as the MILP solver, and Microsoft `Z3` 4.0 [46] as the SMT solver for linear real arithmetic. We started `Gurobi` with a single thread, since the tools we compare `LTLSubsys` with do not support multi-threading. For all other options we used the default settings. To generate DRAs from LTL properties, we use the tool `ltl2dstar` [55] in version 0.5.1 which first calls `ltl2ba` [56] (version 1.1) to generate nondeterministic Büchi automata and afterward turns them into deterministic Rabin automata.

We compare the results of our tool `LTLSubsys` with the tools `COMICS` [31] and `DiPro` [30] that apply (different) heuristics to obtain small critical subsystems. To the best of our knowledge these are the only available tools for counterexample generation in form of a critical subsystem.

For `COMICS` we use both its global and its local search algorithm on the non-abstracted DTMC. We did not make use of its graphical interface, but used the provided command line tool. `COMICS` supports only reachability properties of DTMCs; MDPs and ω-regular properties are not supported yet.

`DiPro` comes with several different search methods: eXtended Best First search (XBF) [57], Eppstein's k shortest paths algorithm [58], K* [59], and K* with X improvement (XK*) [60]. It is possible to extend `DiPro` with user-defined heuristics to guide the search, which can lead to considerable speed-ups. Such a heuristic, however, has to exploit the user's knowledge about the structure of the model under consideration. Since all other approaches work for arbitrary models without knowing their internals, we did not develop any heuristics for `DiPro` to make the comparison fair. `DiPro` is also restricted to reachability properties, but besides DTMCs it can handle MDPs using the Eppstein, K*, and XK* engines; XBF does not support MDPs.

5.2. Benchmarks

As benchmark models we use the following randomized protocols and algorithms, which are all publicly available from the `PRISM` benchmark repository [19] at `http://www.prismmodelchecker.org/casestudies`. While `DiPro` can read PRISM models directly, we used the export function of `PRISM` to convert them into `MRMC`'s input format [17], which is essentially a list of states and transitions. `MRMC`'s input format can be read both by `LTLSubsys` and `COMICS`.

The following benchmarks are DTMCs:

- `sleader`-N-K is a *synchronous leader election protocol* [61]. Its purpose is to identify a leader node in a symmetric synchronous network ring of N participants. Each node randomly chooses a value from $\{1, \ldots, K\}$ and sends its drawn number around the ring. The node with the highest unique number becomes the leader. If there is no unique number, a new round starts. We check if a leader is finally elected with a large enough probability (Property 1) and if the probability to need at least three election rounds is small enough (Property 2).
  - Property 1: $\mathcal{P}_{\leq \lambda_1}(\Diamond \text{ elected})$
  - Property 2: $\mathcal{P}_{\leq \lambda_2}(\text{start} \wedge \bigcirc\Diamond(\text{start} \wedge \bigcirc\Diamond(\text{start} \wedge \bigcirc\Diamond \text{ elected})))$
- `crowds`-N-R is a model of the *crowds protocol* [62], which provides a mechanism for anonymous surfing on the Internet. The idea is that each node sends a packet with probability $p = 0.8$ directly to the target node, but with probability $1 - p$ it is sent to a randomly chosen node in the crowd. A fixed percentage of the members are corrupt and try to identify the sender of a packet. The parameter R denotes the number of rounds in which packets are sent, N is the number of non-corrupt crowd members. We check the property that the sender gets identified by a corrupt crowds member once (Property 1) and infinitely often (Property 2), respectively.

– Property 1: $\mathcal{P}_{\leq\lambda_1}(\Diamond\,\text{identified})$ (identified once)
– Property 2: $\mathcal{P}_{\leq\lambda_2}(\Box\,\Diamond\,\text{identified})$ (identified infinitely often)

- `nand`-N-K: This benchmark is about constructing reliable computation from unreliable components [63, 64]. It uses a redundancy technique called *NAND multiplexing*. The model operates in stages, each of which contains N NAND gates. K is the number of stages. We check the property that never a reliable state is reached.

  Property: $\mathcal{P}_{\leq\lambda_1}(\Box\neg\text{reliable})$

- `brp`-N-K is the *bounded retransmission protocol* [65, 66]. A file, which consists of N chunks, has to be transferred over an unreliable network. On the way to the target node, chunks might get lost. Therefore each chunk is transferred up to K times until the target node has received it properly and the sender node has obtained an acknowledgment thereof. We check the property that the sender is unsure whether the target node has successfully received the file.

  – Property: $\mathcal{P}_{\leq\lambda_1}(\Diamond\,\text{sender is unsure})$

We additionally used the following MDP benchmarks:

- `aleader`-N is the *asynchronous leader election protocol* [61]. Here, a leader is chosen from an asynchronous ring of N nodes in a network. Every node sends a number 0 or 1, each with probability 0.5, to the next node in the ring. If a node chooses 0 while his predecessor has sent 1, the node is deactivated. When only one node remains active, it becomes the leader. As the ring is not synchronized, the message sending has to be regulated by a scheduler.

  – Property: $\mathcal{P}_{\leq\lambda_1}(\Diamond\,\text{one node is elected as leader})$

- `consensus`-N-K is the *randomized consensus shared coin protocol* [67] that establishes agreement between N asynchronous processes. The processes access a global counter which is increased or decreased in dependence of a coin flipping which is performed when a process enters the protocol. Dependent on the current counter value and the values of N and K the process decides whether it agrees or not. The protocol proceeds in rounds as long as no agreement is achieved. As different processes may try to access the protocol at the same time, it is nondeterministically decided which process may flip a coin.

  – Property: $\mathcal{P}_{\leq\lambda_1}(\Diamond\,\text{all processes have flipped their coin and made their decision})$

- `csma`-N-K is a `PRISM`-model of the *IEEE 802.3 CSMA/CD communication protocol*, which is described in [68]. The protocol aims at the minimization of data collisions in a network of N processes with one single channel. If a process tries to send data while the channel is busy, the process waits a number of time slots, which is determined by K.

  – Property: $\mathcal{P}_{\leq\lambda_1}(\Diamond\,\text{all processes have delivered their message})$

Table 1 provides information about the used benchmarks. It contains besides the name of the instance its number of states ("$|S|$") and transitions ("$|E|$"). For the considered properties we give the actual probability ("$\Pr(\mathcal{L}_i)$") and the imposed upper bound ("λ_i"). The probabilities were determined using `PRISM` version 4.0.3 [15], which took only a few seconds per instance.

All experiments were carried out on a Quad-Core Intel Xeon E5-2450 Processor running at 2.1 GHz clock frequency with 32 GB of main memory under Kubuntu 12.04 Linux running in 64 bit mode. Unless otherwise stated, a time limit of 3 600 seconds and a memory limit of 16 GB were set.

5.3. Evaluation

Table 2 lists the benchmark results, obtained using `LTLSubsys`, for the series of instances of the DTMC and MDP benchmarks described above. The first block of columns contains the name of the benchmark, the considered property and the number of states in the MCS. In case `LTLSubsys` was not able to determine the

	Model	$\lvert S\rvert$	$\lvert E\rvert$	Property $\mathcal{L}_1$		Property $\mathcal{L}_2$	
				$\Pr(\mathcal{L}_1)$	λ_1	$\Pr(\mathcal{L}_2)$	λ_2
DTMCs	brp32-2	1 349	1 731	$2.61 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$		
	brp512-2	21 509	27 651	$2.61 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$		
	crowds5-4	3 515	6 035	0.235	0.1	0.235	0.1
	crowds5-6	18 817	32 677	0.427	0.1	0.427	0.1
	crowds5-8	68 740	120 220	0.591	0.1	0.591	0.1
	crowds12-6	829 669	2 166 277	0.332	0.1	0.332	0.1
	nand5-2	1 728	2 505	0.389	0.2		
	nand5-3	2 526	3 639	0.384	0.2		
	nand5-4	3 324	4 773	0.386	0.2		
	nand25-2	347 828	541 775	0.435	0.1		
	sleader4-4	782	1 037	1.0	0.5	0.02441	0.01
	sleader4-6	3 902	5 197	1.0	0.5	0.005487	0.001
	sleader4-8	12 302	16 397	1.0	0.5	0.001846	0.0005
	sleader8-4	458 847	524 382	1.0	0.5	0.057478	0.01
MDPs	aleader3	364	573	1.0	0.5		
	aleader4	3 172	6 252	1.0	0.5		
	consensus2-2	272	400	1.0	0.1		
	consensus2-4	528	784	1.0	0.1		
	csma2-2	1 038	1 054	1.0	0.1		
	csma2-4	7 958	7 988	1.0	0.1		
	csma2-6	66 718	66 788	1.0	0.1		

Table 1: Model statistics of the benchmarks

optimal subsystem (or prove optimality) due to resource restrictions (entries marked with *), we give the size of the smallest subsystem `LTLSubsys` was able to find.

The next block ("Without cuts") shows the results obtained by `LTLSubsys` using the MILP formulations (2a)–(2e) for reachability properties of DTMCs, (7a)–(7f) for other ω-regular properties of DTMCs, and (8a)–(8i) for reachability properties of MDPs without adding any of the redundant constraints described in Section 3.3. We give the number of variables ("|Vars|") and constraints ("|Constr|") in the MILP formulation, the computation time ("Time") in seconds and the memory consumption ("Mem.") in megabytes. The running times include reading the model from the file, generating the DRA and the product automaton, and the computation of the subsystem. If the time limit was exceeded, we give instead in parentheses the computed lower bound on the size of the subsystem. For sleader8-4 with the second property and without additional constraints, the Simplex algorithm on the LP-relaxation did not terminate within the time bound; therefore we cannot give a lower bound on the optimal solution in this case.

For the last block of columns ("Best cut combination") we ran `LTLSubsys` with all possible combinations of redundant constraints and report one which lead to the smallest computation time or—in case none terminated within one hour—one with the best lower bound on the size of the MCS. For the reported combination of constraints, the first four columns show which optimizations were enabled: forward cuts ("F"), backward cuts ("B"), SCC cuts ("S"), and reachability constraints ("R"). Note that for MDPs we do not have SCC cuts. For the SCC cuts, we specify whether input cuts ("in"), output cuts ("out") or both ("both") are used. For reachability constraints, either forward ("fwd"), backward ("bwd") constraints or both ("both") can be used. Additionally we report, as before, the computation time and memory consumption.

Our optimizations, presented for DTMCs in Section 3.3, have a great impact on the solving times. Especially the forward and backward constraints improved the feasibility of our approaches for all benchmarks. However, it was not always predictable which cut improved the running-times on individual benchmarks, e. g., the complete reachability constraints sometimes slowed down the computations due to the high amount of variables while they highly enhanced the running times for both leader election protocols. Consider the sleader4-4 benchmark and the second property, where the computation took without optimizations more than 500 seconds while the MILP together with forward constraints and backward reachability constraints

	Model	φ	$\lvert S_{\min}\rvert$	Without cuts |Vars|	|Constr|	Time	Mem.	Best cut combination F	B	S	R	Time	Mem.
DTMCs	brp32-2	1	212	1 992	1 988	0.09	8	×	×	×	×	0.09	8
	brp512-2	1	3 263	31 752	31 748	18.85	70	×	×	×	×	18.85	70
	crowds5-4	1	83	2 161	2 119	5.25	17	√	√	×	×	5.10	21
	crowds5-4	2	188	2 811	2 769	11.80	24	√	√	out	×	8.82	31
	crowds5-6	1	83	14 436	14 184	190.50	129	×	√	×	×	42.99	99
	crowds5-6	2	415	21 604	21 352	554.33	134	√	√	in	×	230.52	151
	crowds5-8	1	83	56 156	55 232	310.39	347	√	√	×	×	178.32	326
	crowds5-8	2	*1 037**	94 208	93 284	TO *(828)*	850	×	×	out	×	TO *(830)*	1 016
	crowds12-6	1	*270**	395 488	391 848	TO *(223)*	3 944	×	√	×	×	TO *(249)*	3 047
	crowds12-6	2	*1 995**	509 098	505 458	TO *(1 710)*	2 256	√	√	×	×	TO *(1 762)*	3 013
	nand5-2	1	394	3 457	3 447	19.50	21	×	×	×	×	19.50	21
	nand5-3	1	614	5 053	5 043	50.01	36	×	×	in	×	48.22	36
	nand5-4	1	854	6 649	6 639	299.62	101	√	√	in	×	268.40	57
	nand25-2	1	*344 829**	695 567	695 521	TO *(2 661)*	2 531	√	√	×	×	TO *(2 743)*	2 471
	sleader4-4	1	392	1 565	1 563	0.33	10	×	√	×	bwd	0.09	10
	sleader4-4	2	394	1 809	2 051	506.90	53	√	×	×	bwd	2.88	18
	sleader4-6	1	1 950	7 805	7 803	2.24	23	×	√	out	bwd	0.67	31
	sleader4-6	2	949	8 385	8 963	119.75	100	√	√	×	bwd	14.22	40
	sleader4-8	1	6 149	24 605	24 603	13.00	156	×	√	both	bwd	3.62	83
	sleader4-8	2	3 712	25 665	26 723	1 100.20	161	√	√	both	bwd	264.44	157
	sleader8-4	1	229 389	917 695	917 693	1 021.33	1 018	×	×	both	×	508.50	1 010
	sleader8-4	2	*116 113**	1 137 667	1 357 637	TO	3 616	√	√	×	×	TO *(8 086)*	2 808
MDPs	aleader3	1	*66**	2 677	1 295	TO *(18)*	1 702	√	×		×	TO *(27)*	874
	aleader4	1	*215**	26 076	12 588	TO *(10)*	2 454	×	×		×	TO *(10)*	2 454
	consensus2-2	1	15	1 964	928	TO *(9)*	3 529	√	√		×	2 167.19	217
	consensus2-4	1	*35**	3 852	1 824	TO *(8)*	4 732	√	√		×	TO *(12)*	1 848
	csma2-2	1	195	6 482	3 124	TO *(184)*	1 017	√	×		fwd	638.52	161
	csma2-4	1	410	50 400	23 890	TO *(408)*	1 154	√	√		both	1 342.57	234
	csma2-6	1	415	426 678	200 170	2 364.10	910	×	×		×	2 364.10	910

Table 2: Benchmark results of `LTLSubsys` for DTMCs and MDPs. All times are measured in seconds, memory consumption in MB. The time limit was set to 3 600 seconds, the memory limit to 16 GB.

	Model	φ	XBF $\lvert S_{\min}\rvert$	Time	Mem.	Eppstein $\lvert S_{\min}\rvert$	Time	Mem.	K* $\lvert S_{\min}\rvert$	Time	Mem.	XK* $\lvert S_{\min}\rvert$	Time	Mem.
DTMCs	brp32-2	1	989	3.05	148	218	1.02	57	218	1.09	59	235	1.94	61
	brp512-2	1	15875	89.28	2384		TO			TO			TO	
	crowds5-4	1	117	1.29	38	670	8.0	300	670	9.06	323	154	3.31	79
	crowds5-6	1	117	1.91	38	670	22.27	805	670	10.26	373	154	3.42	79
	crowds5-8	1	117	1.27	38	670	90.05	2253	670	10.02	373	154	3.55	79
	crowds12-6	1	1260	10.42	1011		TO			TO		726	52.63	1190
	sleader4-4	1	563	2.37	84	400	1.11	40	400	1.1	40	472	3.56	46
	sleader4-6	1	3541	10.3	428	1957	4.85	200	1957	4.91	200	2101	12.43	222
	sleader4-8	1	6221	27.85	1305	6160	23.92	629	6160	24.48	629	7342	48.02	719
	sleader8-4	1		TO			TO			TO			TO	
MDPs	aleader3	1				107	8.98	11855	107	10.19	11856	107	10.27	11856
	aleader4	1					MO			MO			MO	
	consensus2-2	1				25	0.6	17	25	0.65	15	25	0.65	15
	consensus2-4	1					MO			MO			MO	
	csma2-2	1				611	2.05	2257	614	2.08	2311	614	2.09	2311
	csma2-4	1				788	8.36	2762	788	5.84	2611	788	5.82	2611
	csma2-6	1				797	70.39	4980	518	4.35	3103	518	4.36	3103

Table 3: Benchmark results of `DiPro` for DTMCs and MDPs. All times are measured in seconds, memory consumption in MB. The time limit was set to 3 600 seconds, the memory limit to 16 GB. Note that the XBF algorithm does not support MDPs. `DiPro` only supports reachability properties. Therefore the NAND instances are missing in the table.

			global search			local search		
	Model	φ	$\lvert S_{\min}\rvert$	Time	Mem.	$\lvert S_{\min}\rvert$	Time	Mem.
DTMCs	brp32-2	1	219	0.01	2.31	828	0.18	10.82
	brp512-2	1	9 140	0.74	136.67	15 713	138.26	3251.87
	crowds5-4	1	143	0.01	3.86	89	0.06	4.09
	crowds5-6	1	143	0.04	16.34	89	0.63	15.83
	crowds5-8	1	143	0.19	55.42	89	2.55	53.09
	crowds12-6	1	591	4.21	759.00		TO	
	sleader4-4	1	398	0.01	1.66	462	0.05	4.82
	sleader4-6	1	1 960	0.13	5.58	1 962	1.45	58.42
	sleader4-8	1	6 160	0.26	15.41	6 426	13.68	564.81
	sleader8-4	1	229 438	385.56	510.39		TO	

Table 4: Benchmark results of `COMICS` for DTMCs. All times are measured in seconds, memory consumption in MB. The time limit was set to 3 600 seconds, the memory limit to 16 GB. Note that `COMICS` supports neither MDPs nor ω-regular properties besides reachability properties. Therefore the NAND instances are missing in the table.

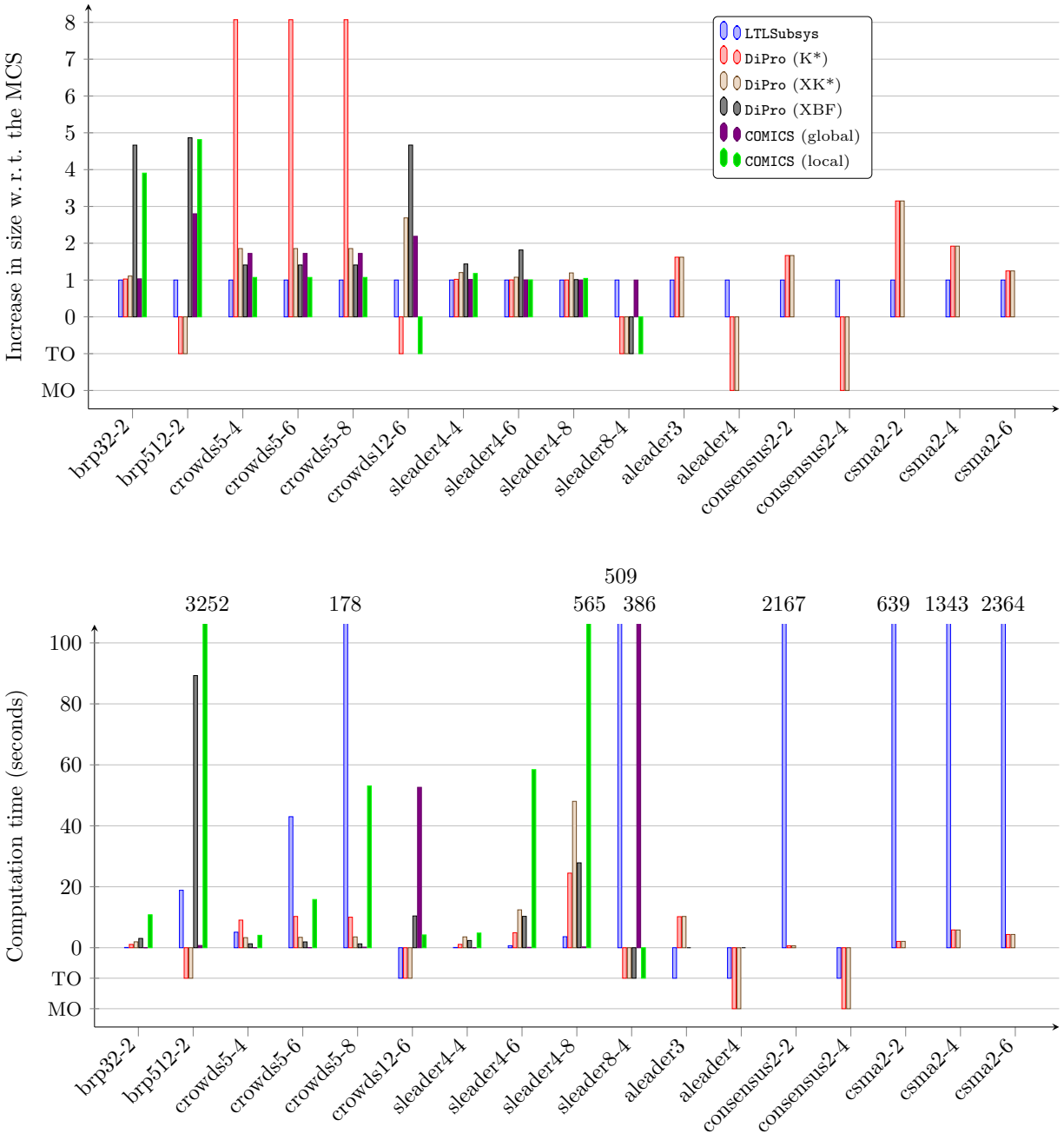


Figure 2: Comparison of the sizes of the computed systems and the computation times of the different tools. The time limit was set to 1 hour, the memory limit to 16 GB.

was solved to optimality within 2.88 seconds.

We compare the MILP formulation for DTMCs (Section 3.2) against the SMT formulation (Section 3.1). However, `Z3` runs into a timeout for all instances in Table 2. We therefore applied `Z3` to the smaller instance crowds3-3 with $\lambda = 0.1$. It consists of 396 states and has an MCS with 39 states. `Z3` needed for this small instance 8 526.30 seconds, while `Gurobi` solved the MILP formulation within 0.09 seconds.

Tables 3 and 4 list the results of `DiPro` and `COMICS`, respectively, on our model instances. For each of the four algorithms that are available in `DiPro` we give the size of the computed subsystem ("$|S_{\min}|$"), the computation time in seconds ("Time") and the memory consumption in megabytes ("Mem."). For `COMICS` we give the same data both for local and for global search.

Regarding the sizes of the computed subsystems we can observe that *none of the heuristic tools was able to find an MCS.* No heuristic algorithm dominates the others w. r. t. the size of the subsystem. `COMICS`' global search seems to be the fastest, but other methods sometimes yield smaller subsystems.

In some cases the differences in size to the MCS are considerable, cf. crowds12-6, for which `DiPro`'s XBF algorithm returned 1 260 states, while the MCS contains at most 270 states. For brp512-2, both local search of `COMICS` and `DiPro`'s XBF algorithm returns subsystems with more than 15 000 states, whereas the MCS consists of only 3 263 states.

The running time of `LTLSubsys` is often significantly larger than the times of the heuristic tools. However, `LTLSubsys` solves the optimization problem exactly, while `COMICS` and `DiPro` apply heuristics without any guarantees on the solution quality. Therefore `LTLSubsys` is only able to solve smaller instances of a few thousand states to optimality. In many cases in which the computation has to be terminated prematurely, `LTLSubsys` returns a subsystem that is much smaller than the heuristically computed subsystems by `COMICS` and `DiPro`. State-of-the-art MILP solvers apply very sophisticated heuristics to find good solutions quickly. Additionally a lower bound on the value of the best solution is obtained from an MILP solver. This allows to judge how far the found solution is at most from the optimum. For some instances, the gap between the best solution and the lower bound is fairly small—see, for example, crowds12-6 (property 2) with a solution of 1 995 states and a lower bound of 1 710 states. In other cases, the gap is much larger, e. g., for aleader4 with 215 compared to 10.

Figure 2 provides a direct comparison of the different algorithms on all considered instances with reachability properties. The upper diagram shows the sizes of the different computed subsystems divided by the size of the MCS computed by `LTLSubsys`. For the sake of readability, we abstain from showing the results of `DiPro` with Eppstein's k shortest paths algorithm since it always performed worse than `DiPro` with the K* algorithm. One the one hand, most algorithms yield subsystems whose size is smaller than twice the size of the MCS. On the other hand, we can observe that some of the subsystems are up to eight times as large as the MCS.

The lower diagram shows the computation times. If an experiment was finished within the time limit, but took more than 100 seconds, we give the running time above the diagram. With a few exceptions (e. g., brp512-2), the heuristic approaches are faster than `LTLSubsys` at the price of larger subsystems.

In Figure 3 we study the evolution of the sizes of computed critical subsystems and the computation times, depending on the value of λ. We computed a critical subsystem of crowds5-6 for $\lambda \in [0, 0.42]$ with each of the three tools. The left graphic shows the sizes of the subsystems, the right one the computation times. We can observe that the gap between the heuristically computed and the minimal subsystems increases with increasing λ. The sizes of the subsystems computed by the various heuristic approaches are similar, and no clear winner can be identified. On crowds5-6, the fastest approach is the local search of `COMICS`, closely followed by the XBF algorithm implemented in `DiPro`. This trend can, however, not be generalized to other benchmark models. `DiPro` using the XK* algorithm fails for values $\lambda \geq 0.35$ with an out-of-memory error after about three hours of computation. We observed the same for `COMICS` for $\lambda \geq 0.33$ when using global search. Since all heuristic tools increase the subsystem until it becomes critical, no result is returned if the computation is aborted prematurely. `LTLSubsys` runs into a timeout for most values of $\lambda \geq 0.23$. In this case the best found solution is shown.

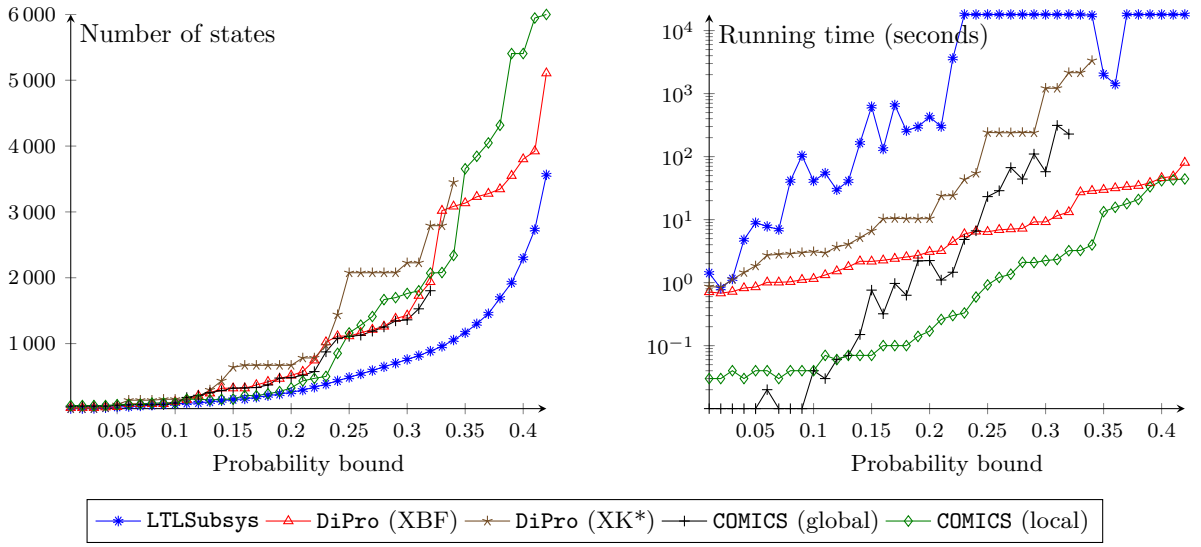


Figure 3: Size of the computed subsystem of crowds5-6 and its computation time for different values of λ, comparing `COMICS` and `DiPro` with `LTLSubsys`. The time limit was set to 5 hours. `DiPro` with the XK* algorithm failed on $\lambda \geq 0.35$ due to the memory limit. The same holds for `COMICS` using global search for $\lambda \geq 0.33$.

In principle, the heuristic tools and `LTLSubsys` can also be combined: One can first compute a small critical subsystem using `COMICS` or `DiPro` and feed its solution into the MILP solver. If a good heuristic solution is available early during the search for an optimal solution it enables the solver to prune branches of the search space which cannot contain a better solution. This can speed up the computation in some cases.

6. Conclusion

In this paper we presented methods for the computation of *optimal* counterexamples in the form of minimal critical subsystems for DTMCs and MDPs. Our algorithms are based on mixed integer linear programming. We presented the MILP formulation, proved its correctness, and suggested several optimizations to speed up the MILP solver. Contrary to available tools, our methods are not restricted to reachability properties but can also handle arbitrary ω-regular properties. Our experiments with a prototype implementation have shown that in most cases they yield (much) smaller subsystems than the available heuristic tools, in some cases even up to two orders of magnitude. Even in case the exact minimization does not terminate within the given time limit, our methods yield very good approximative solutions together with a lower bound on the size of the MCS. This allows to judge the quality of the approximation. None of the other tools is able to give such information or the actual proof of minimality.

As future work we will investigate the complexity of MCS for reachability properties of DTMCs. For MDPs it has been proven to be NP-complete, but for DTMCs such a result is missing. Furthermore we will develop more optimizations, in particular for MDPs, to speed up the computation. As most benchmarks are given as compositional models, we want to extend our approaches such that optimal counterexamples on the basis of the single components are computed, in contrast to the monolithic composed system. We will investigate the extension of our approaches to further models whose model checking algorithms are based on the solution of linear equation systems.

References

[1] E. M. Clarke, The birth of model checking, in: 25 Years of Model Checking – History, Achievements, Perspectives, Vol. 5000 of LNCS, Springer, 2008, pp. 1–26.

[2] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement, in: Proc. of CAV, Vol. 1855 of LNCS, Springer, 2000, pp. 154–169.
[3] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith, Counterexample-guided abstraction refinement for symbolic model checking, Journal of the ACM 50 (5) (2003) 752–794.
[4] H. Hermanns, B. Wachter, L. Zhang, Probabilistic CEGAR, in: Proc. of CAV, Vol. 5123 of LNCS, Springer, 2008, pp. 162–175.
[5] R. Chadha, M. Viswanathan, A counterexample-guided abstraction-refinement framework for Markov decision processes, ACM Transactions on Computational Logic 12 (1) (2010) 1–45.
[6] E. M. Clarke, H. Veith, Counterexamples revisited: Principles, algorithms, applications, in: Verification: Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday, Vol. 2772 of LNCS, Springer, 2003, pp. 208–224.
[7] P. Gastin, P. Moro, M. Zeitoun, Minimization of counterexamples in SPIN, in: Proc. of SPIN, Vol. 2989 of LNCS, Springer, 2004, pp. 92–108.
[8] E. M. Clarke, O. Grumberg, K. L. McMillan, X. Zhao, Efficient generation of counterexamples and witnesses in symbolic model checking, in: Proc. of DAC, IEEE Computer Society, 1995, pp. 427–432.
[9] E. M. Clarke, S. Jha, Y. Lu, H. Veith, Tree-like counterexamples in model checking, in: Proc. of LICS, IEEE Computer Society, 2002, pp. 19–29.
[10] S. Busard, C. Pecheur, Rich counter-examples for temporal-epistemic logic model checking, in: Proc. of IWIGP, Vol. 78 of EPTCS, 2012, pp. 39–53.
[11] V. Schuppan, A. Biere, Shortest counterexamples for symbolic model checking of LTL with past, in: Proc. of TACAS, Vol. 3440 of LNCS, Springer, 2005, pp. 493–509.
[12] M. J. Fischer, N. A. Lynch, M. Paterson, Impossibility of distributed consensus with one faulty process, Journal of the ACM 32 (2) (1985) 374–382.
[13] D. Bustan, S. Rubin, M. Y. Vardi, Verifying ω-regular properties of Markov chains, in: Proc. of CAV, Vol. 3114 of LNCS, Springer, 2004, pp. 189–201.
[14] J. Kretínský, J. Esparza, Deterministic automata for the (F, G)-fragment of LTL, in: Proc. of CAV, Vol. 7358 of LNCS, Springer, Berkeley, CA, USA, 2012, pp. 7–22.
[15] M. Z. Kwiatkowska, G. Norman, D. Parker, Prism 4.0: Verification of probabilistic real-time systems, in: Proc. of CAV, Vol. 6806 of LNCS, Springer, 2011, pp. 585–591.
[16] F. Ciesinski, C. Baier, Liquor: A tool for qualitative and quantitative linear time analysis of reactive systems, in: Proc. of QEST, 2006, pp. 131–132.
[17] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, D. N. Jansen, The ins and outs of the probabilistic model checker MRMC, Performance Evaluation 68 (2) (2011) 90–104.
[18] G. D. Penna, B. Intrigila, I. Melatti, E. Tronci, M. V. Zilli, Finite horizon analysis of Markov chains with the Murphi verifier, Software Tools for Technology Transfer 8 (4-5) (2006) 397–409.
[19] M. Kwiatkowska, G. Norman, D. Parker, The PRISM benchmark suite, in: Proc. of QEST, IEEE Computer Society, 2012, pp. 203–204.
[20] T. Han, J.-P. Katoen, B. Damman, Counterexample generation in probabilistic model checking, IEEE Trans. on Software Engineering 35 (2) (2009) 241–257.
[21] R. Wimmer, B. Braitling, B. Becker, Counterexample generation for discrete-time Markov chains using bounded model checking, in: Proc. of VMCAI, Vol. 5403 of LNCS, Springer, 2009, pp. 366–380.
[22] M. E. Andrés, P. D'Argenio, P. van Rossum, Significant diagnostic counterexamples in probabilistic model checking, in: Proc. of HVC, Vol. 5394 of LNCS, Springer, 2008, pp. 129–148.
[23] M. Günther, J. Schuster, M. Siegle, Symbolic calculation of k-shortest paths and related measures with the stochastic process algebra tool Caspa, in: Proc. of DYADEM-FTS, ACM Press, 2010, pp. 13–18.
[24] A. Komuravelli, C. S. Pasareanu, E. M. Clarke, Assume-guarantee abstraction refinement for probabilistic systems, in: Proc. of CAV, Vol. 7358 of LNCS, Springer, 2012, pp. 310–326.
[25] A. Komuravelli, C. S. Pasareanu, E. M. Clarke, Learning probabilistic systems from tree samples, in: Proc. of LICS, IEEE Computer Society, 2012, pp. 441–450.
[26] M. Kattenbelt, M. Huth, Verification and refutation of probabilistic specifications via games, in: Proc. of FSTTCS, Vol. 4 of LIPIcs, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009, pp. 251–262.
[27] H. Fecher, M. Huth, N. Piterman, D. Wagner, PCTL model checking of Markov chains: Truth and falsity as winning strategies in games, Performance Evaluation 67 (9) (2010) 858–872.
[28] H. Aljazzar, S. Leue, Directed explicit state-space search in the generation of counterexamples for stochastic model checking, IEEE Trans. on Software Engineering 36 (1) (2010) 37–60.
[29] N. Jansen, E. Ábrahám, J. Katelaan, R. Wimmer, J.-P. Katoen, B. Becker, Hierarchical counterexamples for discrete-time Markov chains, in: Proc. of ATVA, Vol. 6996 of LNCS, Springer, 2011, pp. 443–452.
[30] H. Aljazzar, F. Leitner-Fischer, S. Leue, D. Simeonov, DiPro – A tool for probabilistic counterexample generation, in: Proc. of SPIN, Vol. 6823 of LNCS, Springer, 2011, pp. 183–187.
[31] N. Jansen, E. Ábrahám, M. Volk, R. Wimmer, J.-P. Katoen, B. Becker, The COMICS tool – Computing minimal counterexamples for DTMCs, in: Proc. of ATVA, Vol. 7561 of LNCS, Springer, 2012, pp. 349–353.
[32] M. Schmalz, D. Varacca, H. Völzer, Counterexamples in probabilistic LTL model checking for Markov chains, in: Proc. of CONCUR, Vol. 5710 of LNCS, Springer, 2009, pp. 587–602.
[33] L. M. de Moura, N. Bjørner, Satisfiability modulo theories: introduction and applications, Communications of the ACM 54 (9) (2011) 69–77.

[34] A. Schrijver, Theory of Linear and Integer Programming, Wiley, 1986.
[35] R. Wimmer, B. Becker, N. Jansen, E. Ábrahám, J.-P. Katoen, Minimal critical subsystems for discrete-time Markov models, in: Proc. of TACAS, Vol. 7214 of LNCS, Springer, 2012, pp. 299–314.
[36] R. Wimmer, B. Becker, N. Jansen, E. Ábrahám, J.-P. Katoen, Minimal critical subsystems as counterexamples for ω-regular DTMC properties, in: Proc. of MBMV, Verlag Dr. Kovač, 2012, pp. 169–180.
[37] J. R. Norris, Markov Chains, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997.
[38] C. Baier, J.-P. Katoen, Principles of Model Checking, The MIT Press, 2008.
[39] M. Y. Vardi, Automatic verification of probabilistic concurrent finite-state programs, in: Proc. of FOCS, IEEE Computer Society, 1985, pp. 327–338.
[40] L. de Alfaro, Formal verification of probabilistic systems, Ph.D. thesis, Stanford University (1997).
[41] M. Y. Vardi, Probabilistic linear-time model checking: An overview of the automata-theoretic approach, in: Proc. of ARTS, Vol. 1601 of LNCS, Springer, 1999, pp. 265–276.
[42] J.-M. Couvreur, N. Saheb, G. Sutre, An optimal automata approach to LTL model checking of probabilistic systems, in: Proc. of LPAR, Vol. 2850 of LNCS, Springer, 2003, pp. 361–375.
[43] S. Safra, Complexity of automata on infinite objects, Ph.D. thesis, The Weizmann Institute of Science, Rehovot, Israel (1989).
[44] R. E. Tarjan, Depth-first search and linear graph algorithms, SIAM Journal on Computing 1 (2) (1972) 146–160.
[45] B. Dutertre, L. M. de Moura, A fast linear-arithmetic solver for DPLL(T), in: Proc. of CAV, Vol. 4144 of LNCS, Springer, 2006, pp. 81–94.
[46] L. M. de Moura, N. Bjørner, Z3: An efficient SMT solver, in: Proc. of TACAS, Vol. 4963 of LNCS, Springer, 2008, pp. 337–340.
[47] C. Barrett, C. Tinelli, CVC3, in: Proc. of CAV, Vol. 4590 of LNCS, Springer, 2007, pp. 298–302.
[48] A. Griggio, A Practical Approach to Satisfiability Modulo Linear Integer Arithmetic, Journal on Satisfiability, Boolean Modeling and Computation 8 (2012) 1–27.
[49] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co Ltd, 1979.
[50] Gurobi Optimization, Inc., Gurobi optimizer reference manual, `http://www.gurobi.com` (2013).
[51] T. Achterberg, SCIP: Solving constraint integer programs, Mathematical Programming Computation 1 (1) (2009) 1–41.
[52] IBM CPLEX optimization studio, version 12.4, `http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/` (2012).
[53] K. Chatterjee, M. Henzinger, Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification, in: Proc. of SODA, ACM Press, 2011, pp. 1318–1336.
[54] A. Pnueli, The temporal logic of programs, in: Proc. of FOCS, IEEE Computer Society, 1977, pp. 46–57. doi:10.1109/SFCS.1977.32.
[55] J. Klein, C. Baier, Experiments with deterministic ω-automata for formulas of linear temporal logic, Theoretical Computer Science 363 (2) (2006) 182–195.
[56] P. Gastin, D. Oddoux, Fast LTL to Büchi automata translation, in: Proc. of CAV, Vol. 2102 of LNCS, Springer, 2001, pp. 53–65.
[57] H. Aljazzar, S. Leue, Extended directed search for probabilistic timed reachability, in: Proc. of FORMATS, Vol. 4202 of LNCS, Springer, 2006, pp. 33–51.
[58] D. Eppstein, Finding the k shortest paths, SIAM Journal on Computing 28 (2) (1998) 652–673.
[59] H. Aljazzar, S. Leue, K*: A heuristic search algorithm for finding the k shortest paths, Artificial Intelligence 175 (18) (2011) 2129–2154.
[60] H. Aljazzar, M. Kuntz, F. Leitner-Fischer, S. Leue, Directed and heuristic counterexample generation for probabilistic model checking – a comparative evaluation, in: Proc. of QUOVADIS, ACM Press, 2010, pp. 25–32.
[61] A. Itai, M. Rodeh, Symmetry breaking in distributed networks, Information and Computation 88 (1) (1990) 60–87.
[62] M. K. Reiter, A. D. Rubin, Crowds: Anonymity for web transactions, ACM Trans. on Information and System Security 1 (1) (1998) 66–92.
[63] J. von Neumann, Probabilistic logics and synthesis of reliable organisms from unreliable components, in: Automata Studies, Princeton University Press, 1956, pp. 43–98.
[64] G. Norman, D. Parker, M. Kwiatkowska, S. Shukla, Evaluating the reliability of NAND multiplexing with PRISM, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 24 (10) (2005) 1629–1637.
[65] P. D'Argenio, B. Jeannet, H. Jensen, K. Larsen, Reachability analysis of probabilistic systems by successive refinements, in: Proc. of PAPM/PROBMIV, Vol. 2165 of LNCS, Springer, 2001, pp. 39–56.
[66] P. R. D'Argenio, J.-P. Katoen, T. C. Ruys, J. Tretmans, The bounded retransmission protocol must be on time!, in: Proc. of TACAS, Vol. 1217 of LNCS, Springer, 1997, pp. 416–431.
[67] J. Aspnes, M. Herlihy, Fast randomized consensus using shared memory, Journal of Algorithms 15 (1) (1990) 441–460.
[68] M. Kwiatkowska, G. Norman, J. Sproston, F. Wang, Symbolic model checking for probabilistic timed automata, Information and Computation 205 (7) (2007) 1027–1077.

Appendix A. SMT-Formulation for Reachability Properties of DTMCs

Let Var be the set of variables of an SMT or MILP problem formulation as defined in Sections 3 and 4. Each variable $v \in \text{Var}$ has a domain $\text{dom}(v)$, which is either $\text{dom}(v) = [0,1] \subseteq \mathbb{R}$ for real-valued variables or $\text{dom}(v) = \{0,1\} \subseteq \mathbb{Z}$ for integer-valued variables. A *variable assignment* is a function $\nu : \text{Var} \to \mathbb{R}$ such that $\nu(v) \in \text{dom}(v)$ for all $v \in \text{Var}$. A constraint is satisfied by an assignment ν, if replacing each variable $v \in \text{Var}$ by $\nu(v)$ yields a tautology.

In the whole appendix, let AP be a set of atomic propositions over which the labelings of all considered DTMCs and MDPs are defined, and $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC. When we consider reachability properties $\mathcal{P}_{\leq\lambda}(\lozenge\, a)$, let furthermore $a \in \text{AP}$ and assume that all BSCCs of $\mathcal{D}$ contain at least one state labeled with a, and $T = \big\{ s \in S \,\big|\, a \in L(s) \big\}$. (Note that if a DTMC has only states relevant for a then all of its BSCCs contain a-labeled states.) For a state set $S' \subseteq S$ with $s_{\text{init}} \in S'$ we use $\mathcal{D}_{S'} = (S', s_{\text{init}}, P', L')$ to denote the DTMC with $P'(s,s') = P(s,s')$ and $L'(s) = L(s)$ for all $s, s' \in S'$.

Lemma 2 *Let $S' \subseteq S$ with $s_{\text{init}} \in S'$. Then the linear equation system*

$$\forall s \in S' \setminus T : \quad p_s = \sum_{s' \in S' \setminus T} P(s,s') \cdot p_{s'} + \sum_{s' \in T} P(s,s') \tag{A.1a}$$

has a unique satisfying assignment mapping the probability $\Pr^s_{\mathcal{D}_{S'}}(\lozenge\, a)$ to each variable p_s.

Proof. First we observe that the assignment $\nu : \text{Var} \to \mathbb{R}$ that maps to each variable p_s the probability $\nu(p_s) = \Pr^s_{\mathcal{D}}(\lozenge\, a)$ to reach T from s in $\mathcal{D}_{S'}$ is a satisfying assignment [38, Theorem 10.15].

Following the proof idea of [38, Theorem 10.19] we show that this satisfying assignment is unique. Suppose that there are two different satisfying assignments $\mu_1, \mu_2 : \text{Var} \to \mathbb{R}$, $\mu_1 \neq \mu_2$, and let $\mu : \text{Var} \to \mathbb{R}$ be their absolute difference, i. e., $\mu(p_s) = |\mu_1(p_s) - \mu_2(p_s)| \geq 0$ for each $s \in S'$.

Since S' is finite, there exists a state $s^* \in S' \setminus T$ such that $\mu(p_{s^*}) \geq \mu(p_s)$ for all $s \in S' \setminus T$. Let s^* be such a state. Because $P(s^*, s) \geq 0$ for all $s \in S'$ and $\sum_{s \in S'} P(s^*, s) \leq 1$, by the definition of μ and by the choice of s^* the following (in)equations hold:

$$\begin{aligned}
\mu(p_{s^*}) &= \big|\mu_1(p_{s^*}) - \mu_2(p_{s^*})\big| \\
&= \Big|\big(\sum_{s \in S' \setminus T} P(s^*,s) \cdot \mu_1(p_s) + \sum_{s \in T} P(s^*,s)\big) - \big(\sum_{s \in S' \setminus T} P(s^*,s) \cdot \mu_2(p_s) + \sum_{s \in T} P(s^*,s)\big)\Big| \\
&= \Big|\sum_{s \in S' \setminus T} P(s^*,s) \cdot (\mu_1(p_s) - \mu_2(p_s))\Big| \\
&\leq \sum_{s \in S' \setminus T} P(s^*,s) \cdot \big|\mu_1(p_s) - \mu_2(p_s)\big| \\
&= \sum_{s \in S' \setminus T} P(s^*,s) \cdot \mu(p_s) \\
&\leq \mu(p_{s^*}) \cdot \sum_{s \in S' \setminus T} P(s^*,s) \\
&\leq \mu(p_{s^*})\,.
\end{aligned}$$

We conclude that

$$\mu(p_{s^*}) = \sum_{s \in S' \setminus T} P(s^*,s) \cdot \mu(p_s)\,.$$

Since the inequality of μ_1 and μ_2 and the maximal property for s^* imply $\mu(p_{s^*}) > 0$, the equations $\sum_{s \in S' \setminus T} P(s^*,s) = 1$ and $\mu(p_s) = \mu(p_{s^*})$ must hold for all $s \in S' \setminus T$. By induction it follows that $\sum_{s' \in S' \setminus T} P(s,s') = 1$ for all states $s \in S' \setminus T$ which are reachable from s^* in $\mathcal{D}$.

Since all BSCCs of $\mathcal{D}$ contain target states, this holds also for the restriction of $\mathcal{D}$ to S'. Therefore there is a path $s_0 s_1 \dots s_n$ in $\mathcal{D}$ starting at $s^* = s_0$ such that $s_i \in S'$ for $i = 0, \dots, n$ and either (i) $s_n \in T$ or (ii) $s_n \notin T$ and $\sum_{s' \in S'} P(s_n, s') < 1$. If $s_n \in T$ then with $P(s_{n-1}, s_n) > 0$ we get $\sum_{s' \in S' \setminus T} P(s_{n-1}, s') < \sum_{s' \in S} P(s_{n-1}, s') \leq 1$, which is a contradiction. Thus $s_n \notin T$ and $\sum_{s' \in S' \setminus T} P(s_n, s') \leq \sum_{s' \in S'} P(s_n, s') < 1$, which again leads to a contradiction. Therefore our assumption $\mu_1 \neq \mu_2$ was wrong. □

Now we prove the soundness and completeness of the SMT-formulation for reachability properties of DTMCs:

$$\text{minimize} \quad \sum_{s \in S} x_s \tag{A.2a}$$

such that

$$\forall s \in T: \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \big(x_s = 1 \wedge p_s = 1\big) \tag{A.2b}$$

$$\forall s \in S \setminus T: \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \Big(x_s = 1 \wedge p_s = \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s)} P(s, s') \cdot p_{s'}\Big) \tag{A.2c}$$

$$p_{s_{\mathrm{init}}} > \lambda. \tag{A.2d}$$

Lemma 3 *The SMT formulation* (A.2a)–(A.2d) *is sound.*

Proof. We prove that for each satisfying assignment ν of the SMT formulation (A.2a)–(A.2d) the DTMC $\mathcal{D}_{S'}$ with $S' = \{s \in S \mid \nu(x_s) = 1\}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$.

Let ν be a satisfying assignment for (A.2a)–(A.2d) and $S' = \{s \in S \mid \nu(x_s) = 1\}$.

1. We show that $\mathcal{D}_{S'}$ is a subsystem of $\mathcal{D}$. From (A.2d) we can conclude $\nu(p_{s_{\mathrm{init}}}) > \lambda \geq 0$, and therefore by the satisfaction of (A.2b)–(A.2c) we have that $\nu(x_{s_{\mathrm{init}}}) = 1$, i. e., $s_{\mathrm{init}} \in S'$. The remaining conditions for $\mathcal{D}_{S'}$ being a subsystem of $\mathcal{D}$ hold by the definition of $\mathcal{D}_{S'}$.
2. We show that $\mathcal{D}_{S'}$ is critical with $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. The constraints (A.2b)–(A.2c) assure that (i) $\nu(p_s) = 0$ for all $s \in S \setminus S'$ and (ii) $\nu(p_s) = 1$ for all $s \in S' \cap T$. Therefore, due to the satisfaction of (A.2c), ν is also a satisfying assignment to

$$\forall s \in S' \setminus T: \quad p_s = \sum_{s' \in S' \setminus T} P'(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P'(s, s')\,. \tag{A.3}$$

   Lemma 2 implies that this satisfying assignment is unique, mapping to each variable p_s, $s \in S'$, the probability $\Pr^{s}_{\mathcal{D}_{S'}}(\Diamond\, a)$. From (A.2d) we conclude that $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a) > \lambda$.
3. It remains to show that $\mathcal{D}_{S'}$ is minimal. Assume the opposite. Then there is some $S'' \subseteq S$ with $|S''| < |S'|$ such that $\mathcal{D}_{S''}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. In (A.2a)–(A.2d) we syntactically replace x_s by 1 if $s \in S''$ and by 0 otherwise. Lemma 2 applied to S'' implies that the constraint system resulting from the above substitution has a unique satisfying assignment. However, for this satisfying assignment the number of positive x_s variables is smaller than for ν, which contradicts our assumption that ν is a satisfying assignment to the optimization problem. □

Lemma 4 *The SMT formulation* (A.2a)–(A.2d) *is complete.*

Proof. We prove that for each MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with state set S' there is a satisfying assignment ν of the SMT formulation (A.2a)–(A.2d) with $S' = \{s \in S \mid \nu(x_s) = 1\}$ and $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$.

Assume that $\mathcal{D}$ violates the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$, and assume a DTMC with state set S' that is an MCS for $\mathcal{D}$ and the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. Then $\mathcal{D}_{S'}$ is also an MCS for $\mathcal{D}$ and the property (it has the same state set but possibly more transitions). We define the assignment $\nu : \mathrm{Var} \to \mathbb{R}$ by (i) $\nu(x_s) = 1$ and $\nu(p_s) = \Pr^{s}_{\mathcal{D}_{S'}}(\Diamond\, a)$ for $s \in S'$ and (ii) $\nu(x_s) = \nu(p_s) = 0$ otherwise. We show that ν satisfies the SMT constraints (A.2a)–(A.2d).

1. By syntactically replacing the x_s variables by their values under ν, the constraints (A.2b)–(A.2c) reduce to $p_s = 0$ for each $s \notin S'$, $p_s = 1$ for each $s \in S' \cap T$ and

$$\forall s \in S' \setminus T: \quad p_s = \sum_{s' \in S' \setminus T} P(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P(s, s') \,. \tag{A.4a}$$

   By Lemma 2, ν is the unique satisfying assignment for this constraint system. That means, ν is a satisfying assignment to (A.2b)–(A.2c).
2. Since $\mathcal{D}_{S'}$ is critical, $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a) > \lambda$, therefore (A.2d) also holds.
3. If the defined assignment did not minimize the number of states, then there would be another satisfying assignment that evaluates x_s to 1 for a smaller number of states. Due to soundness of the SMT formulation (Lemma 3), there would exist an MCS smaller than $\mathcal{D}_{S'}$, which contradicts the minimality assumption for $\mathcal{D}_{S'}$. □

Theorem 3 *The SMT formulation* (A.2a)–(A.2d) *is sound and complete.*

Proof. The SMT formulation is sound by Lemma 3 and complete by Lemma 4. □

Appendix B. MILP-Formulation for Reachability Properties of DTMCs

In all our MILP encodings we require that the values of the probability variables p_s are *at most* the probability to go to a direct successor state times the p_s value of the successor state. In contrast, the model checking equations require equality at this place. Enforcing only an upper bound is necessary, as we have to assign 0 to p_s if s is not part of the subsystem we want to compute. Therefore, we first show that the satisfying assignments for the inequalities stay below the actual reachability probabilities.

Lemma 5 *Let $S' \subseteq S$ with $s_{\text{init}} \in S'$. Then for each satisfying assignment ν of the constraint system*

$$\forall s \in S' \setminus T: \quad p_s \leq \sum_{s' \in S' \setminus T} P(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P(s, s') \tag{B.1a}$$

we have that $\nu(p_s) \leq \Pr^{s}_{\mathcal{D}_{S'}}(\Diamond\, a)$ for each $s \in S' \setminus T$.

Proof. According to Lemma 2, the assignment ν with $\nu(p_s) = \Pr^{s}_{\mathcal{D}_{S'}}(\Diamond\, a)$ for each $s \in S' \setminus T$ is the unique satisfying assignment fulfilling all constraints (B.1a) with equality.

Now we show that for each satisfying assignment μ of (B.1a) we have that $\mu(p_s) \leq \nu(p_s)$ for each $s \in S' \setminus T$. Assume that the converse is true. Then there exists a satisfying assignment μ^* for (B.1a) such that $\mu^*(p_{s^*}) > \nu(p_{s^*})$ for some $s^* \in S' \setminus T$. Let μ^* be such an assignment and s^* such a state, and let $\varepsilon = \mu^*(p_{s^*})$. Then

$$\text{maximize} \quad \sum_{s \in S' \setminus T} p_s \tag{B.2a}$$

such that

$$\forall s \in S' \setminus T: \quad p_s \leq \sum_{s' \in S' \setminus T} P(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P(s, s') \tag{B.2b}$$

$$p_{s^*} \geq \varepsilon \tag{B.2c}$$

has a satisfying assignment, since μ^* is a satisfying assignment to (B.2b)–(B.2c) and the maximum under all satisfying assignments exists because the variable domains are all bounded by closed intervals and all involved constraints are non-strict (linear) inequalities. Let $\mu_{\max}$ be a satisfying assignment to (B.2a)–(B.2c).

From (B.2c) we conclude that $\mu_{\max} \neq \nu$. Since ν satisfies all constraints (B.2b) with equalities and $\mu_{\max} \neq \nu$, there exists at least one $s_{\max} \in S' \setminus T$ such that

$$\mu_{\max}(p_{s_{\max}}) \quad < \quad \sum_{s' \in S' \setminus T} P(s_{\max}, s') \cdot \mu_{\max}(p_{s'}) + \sum_{s' \in S' \cap T} P(s_{\max}, s') \quad =: \quad d\,.$$

Let $s_{\max}$ be such a state. We define the assignment $\mu'_{\max}$ by $\mu'_{\max}(p_{s_{\max}}) = d$ and $\mu'_{\max}(p_s) = \mu_{\max}(p_s)$ for all other states $s \in S' \setminus (T \cup \{s_{\max}\})$. Note that $P(s, s') \geq 0$ and $\mu'_{\max}(p_s) \geq \mu_{\max}(p_s)$ for all $s, s' \in S' \setminus T$, therefore $\mu'_{\max}$ also satisfies (B.2b)–(B.2c):

$$\begin{array}{lrcl}
 & \mu'_{\max}(p_{s_{\max}}) & = & d \\
 & & = & \sum_{s' \in S' \setminus T} P(s_{\max}, s') \cdot \mu_{\max}(p_{s'}) + \sum_{s' \in S' \cap T} P(s_{\max}, s') \\
 & & \leq & \sum_{s' \in S' \setminus T} P(s_{\max}, s') \cdot \mu'_{\max}(p_{s'}) + \sum_{s' \in S' \cap T} P(s_{\max}, s') \\
\forall s \in S' \setminus (T \cup \{s_{\max}\}) : & \mu'_{\max}(p_s) & = & \mu_{\max}(p_s) \\
 & & \leq & \sum_{s' \in S' \setminus T} P(s, s') \cdot \mu_{\max}(p_{s'}) + \sum_{s' \in S' \cap T} P(s, s') \\
 & & \leq & \sum_{s' \in S' \setminus T} P(s, s') \cdot \mu'_{\max}(p_{s'}) + \sum_{s' \in S' \cap T} P(s, s') \\
 & \mu'_{\max}(p_{s^*}) & \geq & \mu_{\max}(p_{s^*}) \\
 & & \geq & \varepsilon \quad \big(= \mu^*(p_{s^*})\big)\,.
\end{array}$$

However, $\mu'_{\max}$ yields a larger sum over the p_s variable values than $\mu_{\max}$, which contradicts the fact that $\mu_{\max}$ is optimal with respect to (B.2a). That means, our assumption about the existence of μ^* was wrong, which proves the statement. □

Now we prove soundness and completeness of the MILP encoding for computing MCSs of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$:

$$\text{minimize} \quad -\frac{1}{2} p_{s_{\text{init}}} + \sum_{s \in S} x_s \tag{B.3a}$$

such that

$$\forall s \in T : \quad p_s = x_s \tag{B.3b}$$

$$\forall s \in S \setminus T : \quad p_s \leq x_s \tag{B.3c}$$

$$\forall s \in S \setminus T : \quad p_s \leq \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s, s') \cdot p_{s'} \tag{B.3d}$$

$$p_{s_{\text{init}}} > \lambda. \tag{B.3e}$$

Lemma 6 *The MILP formulation* (B.3a)–(B.3e) *is sound.*

Proof. We show that for each satisfying assignment ν of the MILP formulation (B.3a)–(B.3e) the DTMC $\mathcal{D}_{S'}$ with $S' = \{s \in S \,|\, \nu(x_s) = 1\}$ is an MCS of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with a maximal probability $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$ to reach T from the initial state s_{init} under all MCSs.

Let ν be a satisfying assignment for (B.3a)–(B.3e) and $S' = \{s \in S \,|\, \nu(x_s) = 1\}$.

1. We show that $\mathcal{D}_{S'}$ is a subsystem of $\mathcal{D}$. From (B.3e) we conclude $0 \leq \lambda < \nu(p_{s_{\text{init}}})$, and therefore by the satisfaction of (B.3b)–(B.3c) we have that $\nu(x_{s_{\text{init}}}) = 1$, i. e., $s_{\text{init}} \in S'$. The remaining conditions for $\mathcal{D}_{S'}$ being a subsystem of $\mathcal{D}$ hold by the definition of $\mathcal{D}_{S'}$.
2. We show that $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. The constraints (B.3b)–(B.3d) assure that (i) $\nu(p_s) = 0$ for all $s \in S \setminus S'$ and (ii) $\nu(p_s) = 1$ for all $s \in S' \cap T$. Therefore, due to the satisfaction of (B.3d), ν is a satisfying assignment to

$$\forall s \in S' \setminus T : \quad p_s \leq \sum_{s' \in S' \setminus T} P'(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P'(s, s')\,. \tag{B.4}$$

Lemma 5 implies $\nu(p_{s_{\text{init}}}) \leq \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$, and Lemma 2 implies that there is also a satisfying assignment mapping $\Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$ to $p_{s_{\text{init}}}$ (satisfying the inequations with equalities). Since ν is a satisfying assignment maximizing $p_{s_{\text{init}}}$ in (B.3a), $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$ must hold.

3. We show that $\mathcal{D}_{S'}$ is critical. In (2) we have shown that $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. Combined with $0 \leq \lambda < \nu(p_{s_{\text{init}}})$ from (1) we get $\lambda < \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$.
4. We show that $\mathcal{D}_{S'}$ is minimal. Assume the opposite. Then there is some $S'' \subseteq S$ with $|S''| < |S'|$ such that $\mathcal{D}_{S''}$ is an MCS for $\mathcal{D}$ and the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. We define the assignment $\mu : \text{Var} \to \mathbb{R}$ by (i) $\mu(x_s) = 1$ and $\mu(p_s) = \Pr^{s}_{\mathcal{D}_{S''}}(\Diamond\, a)$ for $s \in S''$ and (ii) $\mu(x_s) = \mu(p_s) = 0$ otherwise. Lemma 2 applied to S'' implies that μ satisfies (B.3b)–(B.3d) (satisfying all constraints with equality). However, $\sum_{s\in S} \mu(x_s) < \sum_{s\in S} \nu(x_s)$, what contradicts our assumption that ν is optimal with respect to (B.3a).
5. It remains to show that the probability to reach T from s_{init} in $\mathcal{D}_{S'}$ is maximal under all MCSs. This proof is analogous to the previous item. Assume the opposite. Then there is some $S'' \subseteq S$ such that $\mathcal{D}_{S''}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with $\Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a) < \Pr^{s_{\text{init}}}_{\mathcal{D}_{S''}}(\Diamond\, a)$. We define μ as above. Again, Lemma 2 for S'' implies that μ satisfies (B.3b)–(B.3d) (satisfying all constraints with equality). Since $\mathcal{D}_{S'}$ and $\mathcal{D}_{S''}$ are both minimal, $\sum_{s\in S} \nu(x_s) = \sum_{s\in S} \mu(x_s)$. From (2) we know that $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$, i. e.,

$$\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a) < \Pr^{s_{\text{init}}}_{\mathcal{D}_{S''}}(\Diamond\, a) = \mu(p_{s_{\text{init}}})\,.$$

Thus $-\frac{1}{2}\mu(p_{s_{\text{init}}}) + \sum_{s\in S} \mu(x_s) < -\frac{1}{2}\nu(p_{s_{\text{init}}}) + \sum_{s\in S} \nu(x_s)$, contradicting the optimality of ν. □

Lemma 7 *The MILP formulation* (B.3a)–(B.3e) *is complete.*

Proof. Assume that $\mathcal{D}$ violates the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$, and assume a DTMC $\mathcal{D}'$ with state set S' that is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ such that the probability to reach T from s_{init} in $\mathcal{D}'$ is maximal under all MCSs. We show that there is a satisfying assignment ν of the MILP formulation (B.3a)–(B.3e) with $S' = \{s \in S \,|\, \nu(x_s) = 1\}$ and $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$.

Notice that $\mathcal{D}_{S'}$ has the same state set but possibly more transitions than $\mathcal{D}'$, therefore $\mathcal{D}_{S'}$ is also an MCS for $\mathcal{D}$ and the given property with $\Pr^{s_{\text{init}}}_{\mathcal{D}'}(\Diamond\, a) \leq \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. We define the assignment $\nu : \text{Var} \to \mathbb{R}$ by (i) $\nu(x_s) = 1$ and $\nu(p_s) = \Pr^{s}_{\mathcal{D}_{S'}}(\Diamond\, a)$ for $s \in S'$ and (ii) $\nu(x_s) = \nu(p_s) = 0$ otherwise. We show that ν satisfies the MILP constraints (B.3a)–(B.3e).

1. We show that ν satisfies (B.3b)–(B.3d). By syntactically replacing x_s by $\nu(x_s)$ for each $s \in S$, the constraints (B.3b)–(B.3d) reduce to $p_s = 0$ for each $s \in S \setminus S'$, $p_s = 1$ for each $s \in S' \cap T$, and

$$\forall s \in S' \setminus T: \quad p_s \leq \sum_{s' \in S' \setminus T} P(s, s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P(s, s')\,. \tag{B.5a}$$

   By Lemma 2, ν is a satisfying assignment for this constraint system.
2. Since $\mathcal{D}_{S'}$ is critical, $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a) > \lambda$, therefore (B.3e) also holds.
3. We show that ν is optimal with respect to (B.3a).
   If ν did not minimize $\sum_{s\in S} x_s$, then there would be another satisfying assignment that evaluates x_s to 1 for a fewer number of states. Due to soundness (Lemma 6), there would exist an MCS smaller than $\mathcal{D}_{S'}$, which contradicts the minimality assumption for the MCS $\mathcal{D}_{S'}$.
   Similarly, if ν did not maximize $p_{s_{\text{init}}}$, then there would be another satisfying assignment selecting the same (minimal) number of states but mapping a larger value to $p_{s_{\text{init}}}$ than ν does. By soundness (Lemma 6), there would exist an MCS in that the probability to reach T from the initial state is larger than in $\mathcal{D}_{S'}$, which contradicts the assumption that $\mathcal{D}_{S'}$ maximizes the probability to reach a state in T from s_{init} under all MCSs. □

Theorem 4 *The MILP formulation* (B.3a)–(B.3e) *is sound and complete.*

Proof. The MILP formulation is sound by Lemma 6 and complete by Lemma 7.

Appendix C. Correctness of the Optimizations

Next we prove that the following optimizations are optional for both the SMT and MILP formulations for reachability properties of DTMCs, i. e., they can be helpful to speed up the solution process but they do not modify the set of optimal satisfying assignments.

Appendix C.1. Forward/Backward Constraints

$$\forall s \in S \setminus T: \quad -x_s + \sum_{s' \in \mathrm{succ}(s) \setminus \{s\}} x_{s'} \geq 0 \tag{C.1a}$$

$$\forall s \in S \setminus \{s_{\mathrm{init}}\}: \quad -x_s + \sum_{s' \in \mathrm{pred}(s) \setminus \{s\}} x_{s'} \geq 0\ . \tag{C.1b}$$

Lemma 8 *The forward and backward cuts are satisfied by any satisfying assignment of either the SMT formulation* (A.2a)–(A.2d) *or the MILP formulation* (B.3a)–(B.3e).

Proof. Let ν be an arbitrary optimal satisfying assignment of either the SMT or the MILP formulation and let $S' = \{s \in S \mid \nu(x_s) = 1\}$. For $s \in S \setminus S'$ we have $\nu(x_s) = 0$ and $\nu(x_{s'}) \geq 0$ for all $s' \in S$, therefore the constraints are satisfied.

Assume that (C.1a) is violated by ν for a state $s \in S' \setminus T$, i. e., $\nu(x_s) = 1$, but $\nu(x_{s'}) = 0$ for all $s' \in \mathrm{succ}_{\mathcal{D}}(s) \setminus \{s\}$. Then $\nu(p_s) \leq \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s)} P(s,s') \cdot \nu(p_{s'}) = \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s') \setminus \{s\}} P(s,s') \cdot 0 + P(s,s) \cdot \nu(p_s)$, since $\nu(p_{s'}) = 0$ for all $s' \in S$ with $\nu(x_{s'}) = 0$. Since $\mathcal{D}$ has only a-relevant states (implying $P(s,s) < 1$), the only solution is $\nu(p_s) = 0$. Therefore state s can be removed from the MCS $\mathcal{D}_{S'}$ without altering the probability of the initial state. This contradicts the minimality of $\mathcal{D}_{S'}$.

Assume now that (C.1b) is violated for a state $s \in S' \setminus \{s_{\mathrm{init}}\}$, i. e., $\nu(x_s) = 1$, but $\nu(x_{s'}) = 0$ for all $s' \in \mathrm{pred}_{\mathcal{D}}(s) \setminus \{s\}$. Then s is not reachable from s_{init} in $\mathcal{D}_{S'}$, therefore $\mathcal{D}_{S' \setminus \{s\}}$ is a subsystem which is also critical (having the same probability to reach T from s_{init} as $\mathcal{D}_{S'}$) but with a smaller state set, which is a contraction to $\mathcal{D}_{S'}$ being an MCS. □

Appendix C.2. SCC Constraints

$$\forall \mathrm{SCC}\ C, s_{\mathrm{init}} \notin C\ \forall s \in C \setminus \mathrm{In}(C): \quad x_s \ \leq \sum_{s' \in \mathrm{In}(C)} x_{s'} \tag{C.2a}$$

$$\forall \mathrm{SCC}\ C, C \cap T = \emptyset\ \forall s \in C: \quad x_s \ \leq \sum_{s' \in \mathrm{Out}(C)} x_{s'}\ . \tag{C.2b}$$

Lemma 9 *The input and output SCC cuts are satisfied by any optimal satisfying assignment of either the SMT formulation* (A.2a)–(A.2d) *or the MILP formulation* (B.3a)–(B.3e).

Proof. Let ν be an arbitrary optimal satisfying assignment of either the SMT or the MILP formulation and let $S' = \{s \in S \mid \nu(x_s) = 1\}$.

Assume an SCC $C \subseteq S \setminus \{s_{\mathrm{init}}\}$ which violates (C.2a). All paths in $\mathcal{D}$ from s_{init} to T passing through C contain a state in $\mathrm{In}(C)$. If $S' \cap \mathrm{In}(C) = \emptyset$ then there is no path in $\mathcal{D}_{S'}$ from s_{init} to T containing a state from C. Therefore all states in $C \cap S' \neq \emptyset$ can be removed from $\mathcal{D}_{S'}$ without alternating the probability of s_{init}, which contradicts the minimality of $\mathcal{D}_{S'}$.

Now assume that (C.2b) is violated. With the same argument we can show that again all states in $C \cap S' \neq \emptyset$ are irrelevant in $\mathcal{D}_{S'}$, contradicting the minimality assumption. □

Appendix C.3. Forward Reachability Constraints

Consider the forward reachability constraints with $x_s \in \{0,1\} \subseteq \mathbb{Z}$, $t^{\rightarrow}_{s,s'} \in \{0,1\} \subseteq \mathbb{Z}$ and $r^{\rightarrow}_s \in [0,1] \subseteq \mathbb{R}$ for all $s, s' \in S$:

$$\forall s' \in S \setminus \{s_{\text{init}}\}\ \forall s \in \text{pred}(s') : \quad t^{\rightarrow}_{s,s'} \leq x_s \tag{C.3a}$$

$$\forall s' \in S \setminus \{s_{\text{init}}\}\ \forall s \in \text{pred}(s') : \quad r^{\rightarrow}_s < r^{\rightarrow}_{s'} + (1 - t^{\rightarrow}_{s,s'}) \tag{C.3b}$$

$$\forall s' \in S \setminus \{s_{\text{init}}\} : \quad \sum_{s \in \text{pred}(s')} t^{\rightarrow}_{s,s'} = x_{s'}\ . \tag{C.3c}$$

Lemma 10 *For each optimal satisfying assignment ν of either the SMT formulation* (A.2a)–(A.2d) *or the MILP formulation* (B.3a)–(B.3e) *there exists an extending satisfying assignment μ of* (C.3a)–(C.3c) *with $\nu(v) = \mu(v)$ for all $v \in \{x_s, p_s \mid s \in S\}$.*

Proof. Let ν be an arbitrary optimal satisfying assignment of either the SMT or the MILP formulation and let $S' = \{s \in S \mid \nu(x_s) = 1\}$. By the soundness of the SMT and MILP formulations (Theorems 3 and 6) we know that $\mathcal{D}_{S'}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.

We consider the tree which contains for each state $s \in S'$ one shortest path (in terms of the number of states) from s_{init} to s. (This tree is well-defined, since minimality of the MCS $\mathcal{D}_{S'}$ implies that all states in S' are reachable from s_{init} in $\mathcal{D}_{S'}$.) We define a function $f : S' \setminus \{s_{\text{init}}\} \to S'$ by assigning to each state $s \in S' \setminus \{s_{\text{init}}\}$ the predecessor state of s in this tree. We fix the assignment μ by

- $\nu(v) = \mu(v)$ for all $v \in \{x_s, p_s \mid s \in S\}$,
- for all $s, s' \in S$, $\mu(t^{\rightarrow}_{s,s'}) = 1$ if $s' \in S' \setminus \{s_{\text{init}}\}$ and $s = f(s')$, and $\mu(t^{\rightarrow}_{s,s'}) = 0$ otherwise, and
- for all $s \in S$, $\mu(r^{\rightarrow}_s) = n/d$ where n is the length of a shortest path from s_{init} to s in $\mathcal{D}_{S'}$, and d is the maximum of the lengths of all shortest paths from s_{init} to any state in $\mathcal{D}_{S'}$.

It is easy to see that μ satisfies the constraint system (C.3a)–(C.3c). □

We additionally show that, when seen in isolation, for all solutions for (C.3a)–(C.3c) all states in the selected subsystem are reachable from the initial state.

Lemma 11 *Let ν be a satisfying assignment of the forward reachability constraints* (C.3a)–(C.3c)*. Then for all $s' \in S$, if $\nu(x_{s'}) = 1$ then there is a path $s_0 s_1 \dots s_n = s'$ from a state $s_0 = s_{\text{init}}$ to s' with $\nu(x_{s_i}) = 1$ for all $0 \leq i \leq n$.*

Proof. Constraint (C.3c) enforces that each state $s' \in S \setminus \{s_{\text{init}}\}$ with $\nu(x_{s'}) = 1$ has a predecessor state $s \in \text{pred}(s')$ with $t^{\rightarrow}_{s,s'} = 1$. Constraint (C.3a) ensures that for this predecessor state $\nu(x_s) = 1$ holds. Constraint (C.3b) finally ensures that $\nu(r^{\rightarrow}_s) < \nu(r^{\rightarrow}_{s'})$.

Assume there is a state $u_0 \in S \setminus \{s_{\text{init}}\}$ such that the statement of the lemma is false. Then we can construct an infinite sequence $u_0 u_1 u_2 \dots$ such that $u_{i+1} \in \text{pred}(u_i)$, $\nu(x_{u_i}) = 1$, $\nu(t^{\rightarrow}_{u_{i+1},u_i}) = 1$, and $\nu(r^{\leftarrow}_{u_{i+1}}) < \nu(r^{\leftarrow}_{u_i})$ for all $i \geq 0$.

Since S is finite there are $i < k$ with $u_i = u_k$. However $\nu(r^{\leftarrow}_{u_k}) < \nu(r^{\leftarrow}_{u_i})$ holds, which is a contradiction. Therefore our assumption was wrong and the lemma is valid. □

Remark 2 In constraint (C.3c) the equality sign can be replaced by "$\geq$" without any change in the proof. However, equality yields a stronger MILP formulation and is therefore preferred where possible.

Appendix C.4. Backward Reachability Constraints

Consider the backward reachability constraints with $x_s \in \{0,1\} \subseteq \mathbb{Z}$, $t^{\leftarrow}_{s,s'} \in \{0,1\} \subseteq \mathbb{Z}$ and $r^{\leftarrow}_s \in [0,1] \subseteq \mathbb{R}$ for all $s, s' \in S$:

$$\forall s \in S \setminus T\ \forall s' \in \mathrm{succ}(s): \quad t^{\leftarrow}_{s,s'} \leq x_{s'} \tag{C.4a}$$

$$\forall s \in S \setminus T\ \forall s' \in \mathrm{succ}(s): \quad r^{\leftarrow}_s < r^{\leftarrow}_{s'} + (1 - t^{\leftarrow}_{s,s'}) \tag{C.4b}$$

$$\forall s \in S \setminus T: \quad \sum_{s' \in \mathrm{succ}(s)} t^{\leftarrow}_{s,s'} = x_s\ . \tag{C.4c}$$

Lemma 12 *For each optimal satisfying assignment ν of either the SMT formulation* (A.2a)–(A.2d) *or the MILP formulation* (B.3a)–(B.3e) *there exists an extending satisfying assignment μ of* (C.4a)–(C.4c) *with $\nu(v) = \mu(v)$ for all $v \in \{x_s, p_s \mid s \in S\}$.*

Proof. Let ν be an arbitrary optimal satisfying assignment of either the SMT or the MILP formulation and let $S' = \{s \in S \mid \nu(x_s) = 1\}$. By the soundness of the SMT and MILP formulations (Theorems 3 and 6) we now that $\mathcal{D}_{S'}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.

We consider a set Π of paths of $\mathcal{D}_{S'}$ such that Π is postfix-closed and it contains for each state $s \in S'$ exactly one shortest path from s to T in $\mathcal{D}_{S'}$. (Such a set exists, since minimality of the MCS $\mathcal{D}_{S'}$ implies that T can be reached from all states in $\mathcal{D}_{S'}$.) We define the function $f : S' \setminus T \to S'$ by assigning to each state $s \in S' \setminus T$ the unique successor state of s on the shortest path from s to T in Π. We fix the assignment μ by

- $\nu(v) = \mu(v)$ for all $v \in \{x_s, p_s \mid s \in S\}$,
- for all $s, s' \in S$, $\mu(t^{\leftarrow}_{s,s'}) = 1$ if $s \in S' \setminus T$ and $s' = f(s)$, and $\mu(t^{\leftarrow}_{s,s'}) = 0$ otherwise, and
- for all $s \in S$, $\mu(r^{\leftarrow}_s) = 1 - n/d$ where n is the length of a shortest path from s to T in $\mathcal{D}_{S'}$, and d is the maximum of the lengths of all shortest paths from any state to T in $\mathcal{D}_{S'}$.

It is easy to see that μ satisfies the constraint system (C.4a)–(C.4c). □

We additionally show that, when seen in isolation, for all solutions for (C.4a)–(C.4c) T is reachable from all states in the selected subsystem.

Lemma 13 *Let ν be a satisfying assignment of the backward reachability constraints* (C.4a)–(C.4c)*. Then for all $s \in S$, $\nu(x_s) = 1$ implies that there is a path $s = s_0 s_1 \dots s_n$ from s to a state $s_n \in T$ with $\nu(x_{s_i}) = 1$ for all $0 \leq i \leq n$.*

Proof. Constraint (C.4c) enforces that each state $s \in S \setminus T$ with $\nu(x_s) = 1$ has a successor state $s' \in \mathrm{succ}(s)$ with $t^{\leftarrow}_{s,s'} = 1$. Constraint (C.4a) ensures that for this successor state $\nu(x_{s'}) = 1$ holds. Constraint (C.4b) finally ensures that $\nu(r^{\leftarrow}_s) < \nu(r^{\leftarrow}_{s'})$.

Assume that there is a state $u_0 \in S \setminus T$ such that the statement of the lemma is false. Then we can construct an infinite path $u_0 u_1 u_2 \dots$ such that $u_{i+1} \in \mathrm{succ}(u_i)$, $\nu(x_{u_i}) = 1$, $\nu(t^{\leftarrow}_{u_i,u_{i+1}}) = 1$, and $\nu(r^{\leftarrow}_{u_i}) < \nu(r^{\leftarrow}_{u_{i+1}})$ for all $i \geq 0$.

Since S is finite there are $i < k$ with $u_i = u_k$. However, $\nu(r^{\leftarrow}_{u_i}) < \nu(r^{\leftarrow}_{u_k})$ leads to a contradiction. Therefore our assumption was wrong and the lemma is valid. □

Remark 3 In constraint (C.4c) the equality sign can be replaced by "$\geq$" without any change in the proof. However the equality yields a stronger MILP formulation and is therefore preferred where possible.

Theorem 5 *Both the SMT formulation* (A.2a)–(A.2d) *and the MILP formulation* (2a)–(2e) *together with any (combination) of the three above optimizations is sound and complete.*

PROOF. Since each satisfying assignment for the SMT or MILP formulation *with* optimization constraints is also a satisfying assignment for the SMT or MILP formulation *without* optimization constraints, soundness follows directly from the SMT and MILP formulations (Theorems 3 and 6).

For completeness assume an MCS. By the completeness results for the SMT and MILP formulations (Theorems 4 and 7) we know that they have a satisfying assignment inducing the given MCS. Above we have shown that thus satisfying assignment also satisfies the optimization constraints. □

Appendix D. MILP-Formulation for ω-Regular Properties of DTMCs

Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC, $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ an ω-regular property, which is violated by $\mathcal{D}$, and $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. We consider the product $\mathcal{D} \otimes \mathcal{A}$ of the DTMC $\mathcal{D}$ and the DRA $\mathcal{A}$ as in Definition 8 with distribution function P', and assume that all irrelevant states have been removed. To simplify notation we use $U = S \times Q$ and $u = (s, q)$, $u' = (s', q')$, etc. as typical elements from U. Let $\{T_1, \ldots, T_n\}$ be the set of accepting BSCCs of $\mathcal{D} \otimes \mathcal{A}$ and $T = \bigcup_{i=1}^{n} T_i$. The MILP-formulation for an MCS of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ is as follows:

$$\text{minimize} \quad -\frac{1}{2} p_{(s,q)_{\text{init}}} + \sum_{s \in S} x_s \tag{D.1a}$$

such that

$$p_{(s,q)_{\text{init}}} > \lambda \tag{D.1b}$$

$$\forall i = 1, \ldots, n \; \forall u = (s,q) \in T_i : \quad p_u = x_{T_i} \tag{D.1c}$$

$$\forall i = 1, \ldots, n \; \forall u = (s,q) \in T_i : \quad x_s \geq x_{T_i} \tag{D.1d}$$

$$\forall u = (s,q) \in S_{\mathcal{D}\otimes\mathcal{A}} \setminus T : \quad p_u \leq x_s \tag{D.1e}$$

$$\forall u = (s,q) \in S_{\mathcal{D}\otimes\mathcal{A}} \setminus T : \quad p_u \leq \sum_{u' \in \text{succ}_{\mathcal{D}\otimes\mathcal{A}}(u)} P\big(u, u'\big) \cdot p_{u'} \; . \tag{D.1f}$$

In the soundness and completeness proofs for the above formulation we will make use of the following fact.

Lemma 14 *Let $S' \subseteq S$ with $s_{\text{init}} \in S'$ and let $\{B_1, \ldots, B_k\}$ be the set of all accepting BSCCs of $\mathcal{D}_{S'} \otimes \mathcal{A}$. Then $\{B_1, \ldots, B_k\} = \{T_i \mid i \in \{1, \ldots, n\} \wedge T_i \subseteq S'\}$.*

PROOF. Let S' and $\{B_1, \ldots, B_k\}$ be as above. It is easy to see that each accepting BSCC T_i of $\mathcal{D}_S \otimes \mathcal{A}$ with $T_i \subseteq S'$ is also an accepting BSCC of $\mathcal{D}_{S'} \otimes \mathcal{A}$.

For the other direction fix some $j \in \{1, \ldots, k\}$. By definition, B_j is an accepting BSCC in $\mathcal{D}_{S'} \otimes \mathcal{A}$, i. e., B_j is strongly connected, maximal, bottom and accepting in $\mathcal{D}_{S'} \otimes \mathcal{A}$. We show that B_j is an accepting BSCC also in $\mathcal{D}_S \otimes \mathcal{A}$, i. e., B_j is strongly connected, maximal, bottom and accepting in $\mathcal{D}_S \otimes \mathcal{A}$.

- *Strongly connected:* Since $\mathcal{D}_{S'}$ is a subsystem of $\mathcal{D}$, also $\mathcal{D}_{S'} \otimes \mathcal{A}$ is a subsystem of $\mathcal{D}_S \otimes \mathcal{A}$, therefore B_j is a strongly connected state set also in $\mathcal{D}_S \otimes \mathcal{A}$.
- *Maximal:* Since B_j is a bottom SCC in $\mathcal{D}_{S'} \otimes \mathcal{A}$, we have that $\sum_{u' \in B_j} P'(u, u') = 1$ for all $u \in B_j$. Thus B_j cannot be extended to any larger strongly connected state set in $\mathcal{D}_S \otimes \mathcal{A}$, i.e., B_j is maximal also in $\mathcal{D}_S \otimes \mathcal{A}$.
- *Bottom:* The bottom property of B_j in $\mathcal{D}_{S'} \otimes \mathcal{A}$ directly implies its bottom property in $\mathcal{D}_S \otimes \mathcal{A}$.
- *Accepting:* As B_j is accepting in $\mathcal{D}_{S'} \otimes \mathcal{A}$, it is also accepting in $\mathcal{D}_S \otimes \mathcal{A}$.

Thus B_j is an accepting BSCC in $\mathcal{D}_S \otimes \mathcal{A}$. □

Lemma 15 *The MILP formulation* (D.1a)–(D.1f) *is sound.*

PROOF. We show that for each satisfying assignment ν of the MILP constraints (D.1a)–(D.1f) there is a corresponding MCS of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with state space $S' = \{s \in S \mid \nu(x_s) = 1\}$ and a maximal probability $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L})$ to satisfy $\mathcal{L}$ under all MCSs.

Let ν be a satisfying assignment of the MILP constraints (D.1a)–(D.1f) and $S' = \{s \in S \mid \nu(x_s) = 1\}$. Let furthermore $\{B_1, \ldots, B_k\}$ be the set of all accepting BSCCs of $\mathcal{D}_{S'} \otimes \mathcal{A}$ and $B = \cup_{j=1}^{k} B_j$. Lemma 14 states that each accepting BSCC B_j of $\mathcal{D}_{S'} \otimes \mathcal{A}$ is also an accepting BSCC of $\mathcal{D}_S \otimes \mathcal{A}$, i. e., $B_j = T_i$ for some $i \in \{1, \ldots, n\}$. For simplicity, in the following we also write x_{B_j} to denote x_{T_i} with $B_j = T_i$, and define $B' = \{u \in B_j \mid j \in \{1, \ldots, k\} \wedge \nu(x_{B_j}) = 1\} \subseteq B$ to be the set of all states in selected accepting BSCCs of $\mathcal{D}_{S'} \otimes \mathcal{A}$.

1. We show that $\mathcal{D}_{S'}$ is a subsystem of $\mathcal{D}$. From (D.1b) we imply $\nu(p_{(s,q)_{\text{init}}}) > \lambda \geq 0$. Using (D.1c)–(D.1e) we get that $\nu(x_{s_{\text{init}}}) = 1$. The other conditions for $\mathcal{D}_{S'}$ being a subsystem of $\mathcal{D}$ are straightforward by the definition of $\mathcal{D}_{S'}$.
2. Now we show that no state from $B \setminus B'$ is reachable from $(s,q)_{\text{init}}$ in $\mathcal{D}_{S'} \otimes \mathcal{A}$. Assume the opposite. Then there is an accepting BSCC B_j of $\mathcal{D}_{S'} \otimes \mathcal{A}$ that is reachable from $(s,q)_{\text{init}}$ in $\mathcal{D}_{S'} \otimes \mathcal{A}$ such that $\nu(x_{B_j}) = 0$. Assume a shortest path $\pi = u_0 \ldots u_m$ from $(s,q)_{\text{init}} = u_0$ to $B_j \ni u_m$ in $\mathcal{D}_{S'} \otimes \mathcal{A}$, and define $c_i = \Pi_{l=i}^{m-1} P'(u_l, u_{l+1}) > 0$ for $i = 0, \ldots, m$ (with $c_m = 1$) to be the probabilities of the postfixes of π starting at position i. We define an assignment μ by
   - $\mu(x_s) = \nu(x_s)$ for all $s \in S$,
   - $\mu(x_{B_j}) = 1$ and $\mu(x_{T_i}) = \nu(x_{T_i})$ for $T_i \neq B_j$, and
   - $\mu(p_u) = \begin{cases} 1, & \text{for } u \in B_j, \\ \nu(p_{u_i}) + c_i, & \text{for } u = u_i,\ i = 0, \ldots, m-1, \\ \nu(p_u), & \text{otherwise.} \end{cases}$

   Note that $\mu(p_u) \geq \nu(p_u)$ for all $u \in U$. We show that μ is a satisfying assignment to (D.1b)–(D.1e). The only interesting case is (D.1f). For those states from $U \setminus T$ that are not on the path π, the left-hand-side evaluates equal under μ and ν, and the right-hand-side evaluates under μ to a value at least as large as under ν. For states u_i, $i = 0, \ldots, m-1$, on the path π we have the following relations:

$$
\begin{array}{lll}
 & \mu(p_{u_i}) & \\
= & c_i + \nu(p_{u_i}) & \\
= & P'(u_i, u_{i+1}) \cdot c_{i+1} + \nu(p_{u_i}) & \\
\leq & P'(u_i, u_{i+1}) \cdot c_{i+1} + \sum_{u' \in U} P'(u_i, u') \cdot \nu(p_{u'}) & \\
= & P'(u_i, u_{i+1}) \cdot c_{i+1} + P'(u_i, u_{i+1}) \cdot \nu(p_{u_{i+1}}) & + \quad \sum_{u_i \neq u' \in U} P'(u_i, u') \cdot \nu(p_{u'}) \\
= & P'(u_i, u_{i+1}) \cdot (c_{i+1} + \nu(p_{u_{i+1}})) & + \quad \sum_{u_i \neq u' \in U} P'(u_i, u') \cdot \nu(p_{u'}) \\
= & P'(u_i, u_{i+1}) \cdot \mu(p_{u_{i+1}}) & + \quad \sum_{u_i \neq u' \in U} P'(u_i, u') \cdot \nu(p_{u'}) \\
\leq & P'(u_i, u_{i+1}) \cdot \mu(p_{u_{i+1}}) & + \quad \sum_{u_i \neq u' \in U} P'(u_i, u') \cdot \mu(p_{u'}) \\
= & \sum_{u' \in U} P'(u_i, u') \cdot \mu(p_{u'}) &
\end{array}
$$

   Thus (D.1f) is satisfied for all states from $U \setminus T$. However, having $\mu(p_{(s,q)_{\text{init}}}) = \nu(p_{(s,q)_{\text{init}}}) + c_0 > \nu(p_{(s,q)_{\text{init}}})$ and $\mu(x_s) = \nu(x_s)$ for all $s \in S$, the objective function would have a smaller value for μ than for ν, which contradicts the optimality of ν.
3. Now we are able to show that $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L})$ holds. Let $\mathcal{D}'$ be $\mathcal{D}_{S'} \otimes \mathcal{A}$ without the unreachable states $B \setminus B'$ and their connected transitions and let $A = (S' \times Q) \setminus (B \setminus B')$ denote its state space. By Theorem 1 and the above item (2) we have that

$$\Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L}) = \Pr^{(s,q)_{\text{init}}}_{\mathcal{D}_{S'} \otimes \mathcal{A}}(\Diamond B) = \Pr^{(s,q)_{\text{init}}}_{\mathcal{D}'}(\Diamond B') \ .$$

By (D.1c)–(D.1e) it holds that $\nu(p_u) = 0$ for all $u \in U \setminus A$. Since ν satisfies (D.1f), we have for all states $u \in A$ of $\mathcal{D}'$:

$$\begin{aligned} p_u &\leq \sum_{u' \in U} P'(u,u') \cdot p_{u'} \\ &= \sum_{u' \in A} P'(u,u') \cdot p_{u'} \\ &= \sum_{u' \in A \setminus B'} P'(u,u') \cdot p_{u'} + \sum_{u' \in A \cap B'} P'(u,u') \ . \end{aligned}$$

Using Lemma 5 we get that $\nu(p_u) \leq \Pr^u_{\mathcal{D}'}(\Diamond\, B')$ for each $u \in A$. Lemma 2 furthermore states that there is a satisfying assignment μ with $\mu(p_u) = \Pr^u_{\mathcal{D}'}(\Diamond\, B')$ for each $u \in A$. Since ν minimizes the objective function (D.1a), it maximizes the value of $p_{(s,q)_{\text{init}}}$, therefore $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{(s,q)_{\text{init}}}_{\mathcal{D}'}(\Diamond\, B')$.

4. Now it is easy to see that $\mathcal{D}_{S'}$ is critical: Item (1) above showed $\nu(p_{(s,q)_{\text{init}}}) > \lambda$ and item (3) showed $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L})$, together implying $\Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L}) > \lambda$.
5. We show that $\mathcal{D}_{S'}$ is minimal. Assume the opposite. Then there is some $S'' \subseteq S$ with $|S''| < |S'|$ such that $\mathcal{D}_{S''}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$. In (D.1a)–(D.1f) we syntactically replace x_s by 1 if $s \in S''$ and by 0 otherwise, and x_{T_i} by 1 if $T_i \subseteq S''$ and T_i is reachable from $(s,q)_{\text{init}}$ in $\mathcal{D}_{S''} \otimes \mathcal{A}$ and by 0 otherwise. Lemma 2 applied to S'' implies that the constraint system resulting from (D.1c)–(D.1f) by the above substitution has a satisfying assignment; following the argumentation in item (3) above we get that this assignment maps $\Pr^{s_{\text{init}}}_{\mathcal{D}_{S''}}(\mathcal{L}) > \lambda$ to $p_{(s,q)_{\text{init}}}$, thus also satisfying (D.1b)). However, for this satisfying assignment the number of positive x_s variables is smaller than for ν, which contradicts our assumption that ν minimizes the objective function.
6. It remains to show that the probability to satisfy $\mathcal{L}$ from s_{init} in $\mathcal{D}_{S'}$ is maximal under all MCSs. This proof is analogous to the previous item. Assume the opposite. Then there is some $S'' \subseteq S$ such that $\mathcal{D}_{S''}$ is an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with a higher probability to satisfy $\mathcal{L}$ in the initial state.
We apply the same replacement as above to (D.1a)–(D.1f) to get a satisfying assignment μ inducing $\mathcal{D}_{S''}$. Since $\mathcal{D}_{S'}$ and $\mathcal{D}_{S''}$ are both minimal, $\sum_{s \in S} \nu(x_s) = \sum_{s \in S} \mu(x_s)$, however $\nu(p_{(s,q)_{\text{init}}}) < \mu(p_{(s,q)_{\text{init}}})$, contradicting the optimality of ν. □

Lemma 16 *The MILP formulation* (D.1a)–(D.1f) *is complete.*

Proof. Let $\mathcal{D}'$ with state space $S' \subseteq S$ be an MCS of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with a maximal probability to satisfy $\mathcal{L}$ under all MCSs. We show that there is a satisfying assignment ν of the MILP constraints (D.1a)–(D.1f) such that $\nu(x_s) = 1$ iff $s \in S'$, and $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\mathcal{L})$.

Note that $\mathcal{D}_{S'}$ is also an MCS for $\mathcal{D}$ and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with the same state space and a maximal probability to satisfy $\mathcal{L}$ under all MCSs. Let $\Pi = \{\pi \in \text{Paths}^{\text{inf}}_{\mathcal{D}_{S'}}(s_{\text{init}}) \mid \pi \vDash \mathcal{L}\}$ denote the set of infinite paths within the subsystem that satisfy $\mathcal{L}$. Since $\mathcal{D}_{S'}$ is a critical subsystem, $\Pr(\Pi) > \lambda$ holds.

For $\pi = s_0 s_1 \ldots \in \Pi$ let $\pi^* = (s_0,q_0)(s_1,q_1)\ldots$ with $q_0 = \delta\big(q_{\text{init}}, L(s_{\text{init}})\big)$ and $q_{i+1} = \delta\big(q_i, L(s_{i+1})\big)$ be the unique extension of π to the product automaton $\mathcal{D} \otimes \mathcal{A}$. Let $\Pi^* = \{\pi^* \mid \pi \in \Pi\}$ and $\inf(\pi)$ the set of states which occur infinitely often on π. Since all stepwise probabilities are preserved by the extension, we have that $\Pr_{\mathcal{D}}(\Pi) = \Pr_{\mathcal{D}\otimes\mathcal{A}}(\Pi^*) > \lambda$.

We now consider the subsystem $S' \times Q$ of $\mathcal{D} \otimes \mathcal{A}$. Π^* contains only paths in $S' \times Q$. Let $\text{BSCC}(\mathcal{D} \otimes \mathcal{A})$ denote the set of bottom SCCs of $\mathcal{D} \otimes \mathcal{A}$. Then $\Pr\{\pi \in \text{Paths}^{\text{inf}}_{\mathcal{D}\otimes\mathcal{A}}((s,q)_{\text{init}}) \mid \inf(\pi) \in \text{BSCC}(\mathcal{D}\otimes\mathcal{A})\} = 1$ [38, Theorem 10.27]. Contrarily, $\Pr\{\pi \in \text{Paths}^{\text{inf}}_{\mathcal{D}\otimes\mathcal{A}}((s,q)_{\text{init}}) \mid \inf(\pi) \notin \text{BSCC}(\mathcal{D} \otimes \mathcal{A})\} = 0$. We can conclude:

$$\begin{aligned} 0 &\leq \Pr\{\pi^* \in \Pi^* \mid \inf(\pi^*) \notin \text{BSCC}(\mathcal{D} \otimes \mathcal{A})\} \\ &\leq \Pr\{\pi^* \in \text{Paths}^{\text{inf}}_{\mathcal{D}\otimes\mathcal{A}}((s,q)_{\text{init}}) \mid \inf(\pi^*) \notin \text{BSCC}(\mathcal{D} \otimes \mathcal{A})\} \\ &= 0 \end{aligned}$$

and

$$\lambda < \Pr(\Pi^*) = \Pr(\{\pi^* \in \Pi^* \mid \inf(\pi^*) \in \text{BSCC}(\mathcal{D} \otimes \mathcal{A})\}) \ .$$

We now set $C := \{\inf(\pi^*) \mid \pi^* \in \Pi^*\} \cap \text{BSCC}(\mathcal{D} \otimes \mathcal{A})$. We make the following observations:

- all elements of C are BSCCs, and
- $\forall c \in C\ \exists i \in \{1,\ldots,n\} : (\forall u \in c : R_i \notin L'(u)) \wedge (\exists u \in c : A_i \in L'(u))$, i. e., C contains only accepting BSCCs. Otherwise the paths in Π^* were not accepted.

We define the following variable assignment ν for the decision variables: $\nu(x_s) = 1$ iff $s \in S'$ and $\nu(x_{T_i}) = 1$ iff $T_i \in C$. These assignments trigger the following implications in the MILP constraints above:

$$p_u = \begin{cases} 0, & \text{if } u = (s,q),\ s \notin S', \\ 1, & \text{if } u \in T_i \in C, \\ \sum\limits_{u' \in U} P'(u,u') \cdot p_{u'}, & \text{otherwise.} \end{cases}$$

Using Lemma 2, we can show that this linear equation system has a satisfying assignment ν which describes the probability of reaching a target state within the subsystem $S' \times Q$. Therefore $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{(s,q)_{\text{init}}}_{\mathcal{D}\otimes\mathcal{A}}(\Diamond\ accept) \geq \Pr_{\mathcal{D}\otimes\mathcal{A}}(\Pi^*) = \Pr_{\mathcal{D}}(\Pi) > \lambda$.

We show that ν is optimal with respect to (D.1a).

- If ν did not minimize the number of states, then there would be another satisfying assignment that evaluates x_s to 1 for a fewer number of states. Due to soundness of the MILP formulation (Lemma 15), there would exist an MCS smaller than $\mathcal{D}_{S'}$, which contradicts the minimality assumption for $\mathcal{D}_{S'}$.
- Similarly, if ν did not maximize $p_{s_{\text{init}}}$, then there would be another satisfying assignment selecting the same (minimal) number of states but mapping a larger value to $p_{s_{\text{init}}}$ than ν does. By soundness (Lemma 15), there would exist an MCS in that the probability to satisfy $\mathcal{L}$ in the initial state is larger than in $\mathcal{D}_{S'}$, which contradicts the assumption that $\mathcal{D}_{S'}$ maximizes this probability under all MCSs.□

Theorem 6 *The MILP formulation* (D.1a)–(D.1f) *is sound and complete.*

Proof. The MILP formulation is sound by Lemma 15 and complete by Lemma 16. □

Appendix E. Complexity of MCSs for MDPs

Theorem 7 ([5]) *Let $\mathcal{M}$ be an MDP with $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ and $k \in \mathbb{N}$. The problem to decide whether there exists a critical subsystem of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with at most k states is NP-complete.*

Proof. (Adapted from [5]) The problem is in NP, since one can guess a scheduler and a subsystem of $\mathcal{M}$ and verify in polynomial time (using the DTMC model-checking algorithms) that it is critical. The NP-hardness follows from a reduction from the *exact 3-cover* (X3C) problem [49, Problem SP1]:

> Let X be a set with $|X| = 3r$, $r \in \mathbb{N}$, and $C \subseteq 2^X$ a collection with $\forall c \in C : |c| = 3$.
> Question: does there exist $B \subseteq C$ that exactly covers X?

Here, B covers X whenever the subsets in B are pairwise disjoint and $\bigcup_{c\in B} c = X$. As B covers X by sets of cardinality three, B is called an exact 3-cover of X. It is not difficult to see that an exact 3-cover B of X with $|X| = 3r$ has cardinality $|B| = r$.

The idea of the proof is to construct (starting from a set X with $|X| = 3r$) an MDP and a reachability property such that there exists a critical subsystem of bounded size iff X has an exact 3-cover. Let the MDP $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be as follows:

- $S = X \,\dot\cup\, C \,\dot\cup\, \{s_{\text{init}}, t\}$ with $L(t) = \{a\}$ and $L(s) = \emptyset$ otherwise.
- $Act = \{\alpha\} \,\dot\cup\, \{\alpha_c \mid c \in C\}$, and

- P is given by
  - $P(s_{\text{init}}, \alpha, x) = \frac{1}{3r}$ for $x \in X$ and $P(s_{\text{init}}, \alpha, y) = 0$ for all $y \in S \setminus X$,
  - for $x \in X$ we have $P(x, \alpha_c, c) = 1$ for $c \in C$ and $P(x, \alpha_c, y) = 0$ for all $y \in S \setminus C$,
  - for all $c \in C$ we have $P(c, \alpha, t) = 1$ and $P(c, \alpha, y) = 0$ for all $y \in S \setminus \{t\}$,
  - $P(t, \alpha', t) = 1$ for all $\alpha' \in Act$.

For all actions in Act that are not explicitly mentioned in the definition of P for any state $s \in S$, we assume that they form a self-loop at s with probability 1.

Let $\varphi = \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ with $\lambda = 1 - \frac{1}{3r}$. As the maximal probability to reach t from s_{init} is one, $\mathcal{M} \not\models \varphi$. We show that there is a critical subsystem of size $\leq 2 + 4r$ iff X has an exact 3-cover.

"$\Leftarrow$" Let $B \subseteq C$ be an exact 3-cover for X. Thus, $|B| = r$. Consider the subsystem with state space $\{s_{\text{init}}, t\} \,\dot\cup\, X \,\dot\cup\, B$ together with the following deterministic memoryless scheduler σ on $\mathcal{M}$: $\sigma(s_{\text{init}}) = \sigma(c) = \alpha$ for all $c \in C$ and $\sigma(x) = \alpha_c$ if c is the unique element of B such that $x \in c$.

Then for all $x \in X$ there is a path with probability 1 from x to t. We have:

$$\begin{aligned}\Pr^{s_{\text{init}}}_{\mathcal{M}^\sigma}(\Diamond\, a) &= \sum_{x \in X} P^\sigma(s_{\text{init}}, x) \cdot \Pr^{x}_{\mathcal{M}^\sigma}(\Diamond\, a) \\ &= \sum_{x \in X} P^\sigma(s_{\text{init}}, x) \cdot 1 = \sum_{x \in X} \frac{1}{3r} \cdot 1 \\ &= |X| \cdot \frac{1}{3r} = 1.\end{aligned}$$

Thus we have found a critical subsystem of $\mathcal{M}$ with $2 + |X| + |B| = 2 + 4r$ states.

"$\Rightarrow$" Let $\mathcal{M}' = (S', s_{\text{init}}, P', L')$ be a critical subsystem of $\mathcal{M}$ with state space S' of size $\leq 2 + 4r$. Then the probability to reach t from s within $\mathcal{M}'$ exceeds $1 - \frac{1}{3r}$. Since the probability is a multiple of $\frac{1}{3r}$, it must equal 1. We can conclude that all x-states must be contained in $\mathcal{M}'$ and that from each x-state there is a path with probability 1 to t. Therefore for each $x \in X$ there must be some $c \in S' \cup C$ in $\mathcal{M}'$ such that $x \in c$. The number of c-states in S' is at most $2 + 4r - |\{s, t\}| - |X| = r$. Therefore $B = S' \cap C$ is an exact 3-cover of X.

□

Appendix F. MILP-Formulation for Reachability Properties of MDPs

Let in the following $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP, $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ a reachability property violated by $\mathcal{M}$, and $T = \{s \in S \mid a \in L(s)\}$ the set of target states. The MILP formulation for the reachability property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ of $\mathcal{M}$ is as follows:

$$\text{minimize} \quad -\frac{1}{2} p_{s_{\text{init}}} + \sum_{s \in S} x_s \tag{F.1a}$$

such that

$$p_{s_{\text{init}}} > \lambda \tag{F.1b}$$

$$\forall s \in T: \quad p_s = x_s \tag{F.1c}$$

$$\forall s \in S \setminus T: \quad p_s \leq x_s \tag{F.1d}$$

$$\forall s \in S \setminus T: \quad \sum_{\alpha \in Act} \sigma_{s,\alpha} = x_s \tag{F.1e}$$

$$\forall s \in S \setminus T \; \forall \alpha \in Act: \quad p_s \leq (1 - \sigma_{s,\alpha}) + \sum_{s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha)} P(s,\alpha,s') \cdot p_{s'} \tag{F.1f}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)} \; \forall s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha): \quad t^{\leftarrow}_{s,s'} \leq x_{s'} \tag{F.1g}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)} \; \forall s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha): \quad r^{\leftarrow}_{s} < r^{\leftarrow}_{s'} + (1 - t^{\leftarrow}_{s,s'}) \tag{F.1h}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\mathrm{probl}(a)}: \quad (1 - \sigma_{s,\alpha}) + \sum_{s' \in \mathrm{succ}_{\mathcal{M}}(s,\alpha)} t^{\leftarrow}_{s,s'} \geq x_s \; . \tag{F.1i}$$

Lemma 17 *The MILP formulation* (F.1a)–(F.1i) *is sound.*

Proof. Let ν be a satisfying assignment of the MILP constraints (F.1a)–(F.1i) and let $S' = \{s \in S \mid \nu(x_s) = 1\}$. We define the (partial) memoryless deterministic scheduler $\sigma : S' \setminus T \to Act$ by $\sigma(s) = \alpha$ iff $\nu(\sigma_{s,\alpha}) = 1$. The scheduler σ is well-defined, since constraint (F.1e) ensures that for each $s \in S' \setminus T$ there is exactly one action $\alpha \in Act$ with $\sigma_{s,\alpha} = 1$. We show that the DTMC $\mathcal{D}_{S'} = (S', s_{\mathrm{init}}, P', L')$ with $P'(s,s') = P(s,\sigma(s),s')$ and $L'(s') = L(s')$ for all $s \in S' \setminus T$ and $s' \in S'$ is an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ having a maximal probability to reach T under all MCSs.

1. We show that $\mathcal{D}_{S'}$ is a subsystem of $\mathcal{M}$. From (F.1b) we conclude $0 \leq \lambda < \nu(p_{s_{\mathrm{init}}})$, and therefore by the satisfaction of (F.1c)–(F.1d) we have that $\nu(x_{s_{\mathrm{init}}}) = 1$, i. e., $s_{\mathrm{init}} \in S'$. The remaining conditions for $\mathcal{D}_{S'}$ being a subsystem of $\mathcal{M}$ hold by the definition of $\mathcal{D}_{S'}$.
2. We show that all states of $\mathcal{D}_{S'}$ are relevant for a. By definition, from all unproblematic states $s \in S' \setminus S_{\mathcal{M}}^{\mathrm{probl}(a)}$ there is a path in $\mathcal{M}^{\sigma'}$ to a target state for all schedulers σ'. This holds also for each extension of the (partial) scheduler σ. Due to the backward reachability constraints (F.1g)–(F.1i), from all states that are problematic in $\mathcal{M}$ an unproblematic state and therefore also a target state is reachable in $\mathcal{D}_{S'}$ (cf. Remark 3 about the weaker formulation of backward reachability).
3. We show that $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. The constraints (F.1c)–(F.1d) assure that (i) $\nu(p_s) = 0$ for all $s \in S \setminus S'$ and (ii) $\nu(p_s) = 1$ for all $s \in S' \cap T$. Therefore, due to the satisfaction of (F.1f) for the actions selected by σ, the assignment ν satisfies the following constraint system:

$$\begin{aligned} \forall s \in S' \setminus T: \quad p_s &\leq (1 - \sigma_{s,\sigma(s)}) + \textstyle\sum_{s' \in \mathrm{succ}_{\mathcal{M}}(s,\sigma(s))} P(s,\sigma(s),s') \cdot p_{s'} \\ &= \sum_{s' \in S' \setminus T} P'(s,s') \cdot p_{s'} + \sum_{s' \in S' \cap T} P'(s,s') \,. \end{aligned}$$

   As shown in (2), all states of $\mathcal{D}_{S'}$ are relevant for a. Lemma 5 implies $\nu(p_{s_{\mathrm{init}}}) \leq \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$, and Lemma 2 implies that there is also a satisfying assignment mapping $\Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$ to $p_{s_{\mathrm{init}}}$ (satisfying the inequations with equalities). Since ν is a satisfying assignment maximizing $p_{s_{\mathrm{init}}}$ in (F.1a), $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$ must hold.
4. We show that $\mathcal{D}_{S'}$ is critical. In (3) we have shown that $\nu(p_{s_{\mathrm{init}}}) = \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$. Combined with $0 \leq \lambda < \nu(p_{s_{\mathrm{init}}})$ from (1) we get $\lambda < \Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S'}}(\Diamond\, a)$.
5. We show that $\mathcal{D}_{S'}$ is minimal. Assume the opposite. Then there is a scheduler σ'' and a state set $S'' \subseteq S$ with $|S''| < |S'|$ such that the DTMC $\mathcal{D}_{S''} = (S'', s_{\mathrm{init}}, P'', L'')$ with $P''(s,s') = P(s,\sigma''(s),s')$ and $L''(s) = L(s)$ for all $s, s' \in S''$ is an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.
   Note that, since $\mathcal{D}_{S''}$ is minimal, it has only a-relevant states, i. e., T is reachable from all of its states. We consider a set Π of paths of $\mathcal{D}_{S''}$ such that Π is postfix-closed and it contains for each state $s \in S''$ exactly one shortest path from s to T in $\mathcal{D}_{S''}$. We define the function $f : S'' \setminus T \to S''$ by assigning to each state $s \in S'' \setminus T$ the unique successor state of s on the shortest path from s to T in Π.
   Now we define an assignment $\mu : \mathrm{Var} \to \mathbb{R}$ by
   - $\mu(x_s) = 1$ and $\mu(p_s) = \Pr^{s}_{\mathcal{D}_{S''}}(\Diamond\, a)$ for $s \in S''$, and $\mu(x_s) = \mu(p_s) = 0$ otherwise,
   - $\mu(\sigma_{s,\alpha}) = 1$ if $s \in S''$ and $\sigma''(s) = \alpha$, and $\mu(\sigma_{s,\alpha}) = 0$ otherwise,

- for all $s, s' \in S$, $\mu(t^{\leftarrow}_{s,s'}) = 1$ if $s \in S'' \setminus T$ and $s' = f(s)$, and $\mu(t^{\leftarrow}_{s,s'}) = 0$ otherwise, and
- for all $s \in S$, $\mu(r^{\leftarrow}_s) = 1 - n/d$ where n is the length of a shortest path from s to T in $\mathcal{D}_{S''}$, and d is the maximum of the lengths of all shortest paths from any state to T in $\mathcal{D}_{S''}$.

Lemma 2 applied to $\mathcal{M}^{\sigma''}$ and S'' implies that μ satisfies the constraints (F.1f) with equality. The satisfaction of the other constraints is easy to see.
However, $\sum_{s\in S} \mu(x_s) < \sum_{s\in S} \nu(x_s)$, what contradicts our assumption that ν is optimal with respect to (F.1a).

6. It remains to show that the probability to reach T from s_{init} in $\mathcal{D}_{S'}$ is maximal under all MCSs. This proof is analogous to the previous item (5). Assume the opposite. Then there is a scheduler σ'' and some $S'' \subseteq S$ such that $\mathcal{D}_{S''}$ defined as above is an MCS for $\mathcal{M}$ and $\mathcal{P}_{\leq\lambda}(\Diamond a)$ with $\Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond a) < \Pr^{s_{\text{init}}}_{\mathcal{D}_{S''}}(\Diamond a)$. We define μ as above. Again, with the help of Lemma 2 we can show that μ satisfies all constraints. Since $\mathcal{D}_{S'}$ and $\mathcal{D}_{S''}$ are both minimal, $\sum_{s\in S} \nu(x_s) = \sum_{s\in S} \mu(x_s)$. From (3) we know that $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond a)$, i. e.,

$$\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond a) < \Pr^{s_{\text{init}}}_{\mathcal{D}_{S''}}(\Diamond a) = \mu(p_{s_{\text{init}}})\,.$$

Thus $-\frac{1}{2}\mu(p_{s_{\text{init}}}) + \sum_{s\in S} \mu(x_s) < -\frac{1}{2}\nu(p_{s_{\text{init}}}) + \sum_{s\in S} \nu(x_s)$, contradicting the optimality of ν. □

Lemma 18 *The MILP formulation* (F.1a)–(F.1i) *is complete.*

Proof. Since $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\Diamond a)$, there is a scheduler σ and a state set S' such that the DTMC $\mathcal{D}_{S'} = (S', s_{\text{init}}, P', L')$ with $P'(s, s') = P(s, \sigma(s), s')$ and $L'(s) = L(s)$ for all $s, s' \in S'$ is an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\Diamond a)$ having a maximal probability to reach T under all MCSs. We show that there is a satisfying assignment ν of the MILP formulation (F.1a)–(F.1i) with $S' = \{s \in S \mid \nu(x_s) = 1\}$ and $\nu(\sigma_{s,\alpha}) = 1$ iff $\sigma(s) = \alpha$ such that $\nu(p_{s_{\text{init}}}) = \Pr^{s_{\text{init}}}_{\mathcal{D}_{S'}}(\Diamond a)$.

As in the proof of Lemma 12, we consider a set Π of paths of $\mathcal{D}_{S'}$ such that Π is postfix-closed and it contains for each state $s \in S'$ exactly one shortest path from s to T in $\mathcal{D}_{S'}$. (Such a set exists, since minimality of the MCS $\mathcal{D}_{S'}$ implies that T can be reached from all states in $\mathcal{D}_{S'}$.) We define the function $f : S' \setminus T \to S'$ by assigning to each state $s \in S' \setminus T$ the unique successor state of s on the shortest path from s to T in Π.

We define the assignment $\nu : \text{Var} \to \mathbb{R}$ by

- $\nu(x_s) = 1$ and $\nu(p_s) = \Pr^{s}_{\mathcal{D}_{S'}}(\Diamond a)$ for $s \in S'$, and $\nu(x_s) = \nu(p_s) = 0$ otherwise,
- $\nu(\sigma_{s,\alpha}) = 1$ if $s \in S'$ and $\sigma(s) = \alpha$, and $\nu(\sigma_{s,\alpha}) = 1$ otherwise,
- for all $s, s' \in S$, $\nu(t^{\leftarrow}_{s,s'}) = 1$ if $s \in S' \setminus T$ and $s' = f(s)$, and $\nu(t^{\leftarrow}_{s,s'}) = 0$ otherwise, and
- for all $s \in S$, $\nu(r^{\leftarrow}_s) = 1 - n/d$ where n is the length of a shortest path from s to T in $\mathcal{D}_{S'}$, and d is the maximum of the lengths of all shortest paths from any state to T in $\mathcal{D}_{S'}$.

Lemma 2 implies that μ satisfies the constraints (F.1f) with equality. The satisfaction of the other constraints is easy to show by replacing the variables by their values under ν (see also the proof of Lemma 7). □

Theorem 8 *The MILP formulation* (F.1a)–(F.1i) *is sound and complete.*

Proof. The MILP formulation is sound by Lemma 17 and complete by Lemma 18. □

Appendix G. MILP-Formulation for ω-Regular Properties of MDPs

Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP, $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ an ω-regular property which is violated by $\mathcal{M}$, and $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA with $F = \big\{(R_i, A_i) \,\big|\, i = 1, \ldots, n\big\}$ and $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

We consider the product $\mathcal{M} \otimes \mathcal{A}$ of the MDP $\mathcal{M}$ and the DRA $\mathcal{A}$ as in Definition 8 with distribution function P', and assume that all irrelevant states have been removed. To simplify notation we use $U = S \times Q$ and $u = (s, q)$, $u' = (s', q')$, etc. as typical elements from U.

Lemma 1 *Let $(R_i, A_i) \in 2^Q \times 2^Q$ be a pair of a Rabin acceptance condition, $\sigma : U \to Act$ a scheduler, and $M_i \subseteq U$ a set of states with the following properties:*

1. $\forall u \in M_i : \sum_{u' \in \text{succ}(u, \sigma(u)) \cap M_i} P'(u, \sigma(u), u') = 1$,
2. $M_i \cap (S \times R_i) = \emptyset$, *and*
3. *for each state $u \in M_i$ there is a path from u to a state in $S \times A_i$.*

Then the probability of satisfying the acceptance condition F in $\mathcal{M}$ because of the pair (R_i, A_i) is 1 for all $u \in M_i$.

Proof. Since M_i is closed under successors w. r. t. scheduler σ, this set forms a sub-MDP of $\mathcal{M}$. The probability to reach a BSCC under scheduler σ is 1 for every state of M_i. Let $M_i' \subseteq M_i$ be such a BSCC. As M_i' is strongly connected, it forms an end component of $\mathcal{M}$. As a state out of $S \times A_i$ is reachable from every state of M_i, at least one state of $S \times A_i$ has to be included in M_i'. Hence, M_i' is an accepting end component of $\mathcal{M}$. As this holds for every BSCC included in M_i, the probability to reach an accepting end component inside M_i is one. □

The MILP-formulation for an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ is as follows:

$$\text{minimize} \quad -\frac{1}{2}\, p_{(s,q)_{\text{init}}} + \sum_{s \in S} x_s \tag{G.1a}$$

such that

• selection of at most one action per state:

$$\forall u = (s, q) \in U : \quad \sum_{\alpha \in Act} \sigma_{u,\alpha} \leq x_s \tag{G.1b}$$

• for all $i = 1, \ldots, n$ the definition of set M_i (closure w. r. t. $\text{succ}(u, \alpha)$ for $\alpha \in Act$):

$$\forall u \in U\ \forall \alpha \in Act \text{ with } \sum_{u' \in U} P'(u, \alpha, u') < 1 : \quad m_u^i \leq 1 - \sigma_{u,\alpha} \tag{G.1c}$$

$$\forall u \in U\ \forall \alpha \in Act : \quad n_{u,\alpha} \cdot (2 - \sigma_{u,\alpha} - m_u^i) + \sum_{u' \in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} m_{u'}^i \geq n_{u,\alpha} \tag{G.1d}$$

$$\forall u \in S \times R_i : \quad m_u^i = 0 \tag{G.1e}$$

• for all $i = 1, \ldots, n$ backward reachability of $S \times A_i$ within M_i:

$$\forall u \in U\ \forall \alpha \in Act\ \forall u' \in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u, \alpha) : \quad t_{u,u'}^i \leq m_{u'}^i + (1 - \sigma_{u,\alpha}) \tag{G.1f}$$

$$\forall u \in U\ \forall \alpha \in Act\ \forall u' \in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u, \alpha) : \quad r_u^i < r_{u'}^i + (1 - t_{u,u'}^i) + (1 - \sigma_{u,\alpha}) \tag{G.1g}$$

$$\forall u \in S \times (Q \setminus A_i)\ \forall \alpha \in Act : \quad (1 - \sigma_{u,\alpha}) + \sum_{u' \in \text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} t_{u,u'}^i \geq m_u^i \tag{G.1h}$$

• probability computation:

$$p_{(s,q)_{\text{init}}} > \lambda \tag{G.1i}$$

$$\forall i = 1, \ldots, n \ \forall u \in U: \quad p_u \geq m_u^i \tag{G.1j}$$

$$\forall u \in U: \quad p_u \leq \sum_{\alpha \in Act} \sigma_{u,\alpha} \tag{G.1k}$$

$$\forall u \in U \ \forall \alpha \in Act: \quad p_u \leq (1 - \sigma_{u,\alpha}) + \sum_{i=1}^{n} m_u^i + \sum_{u' \in \text{succ}_{\mathcal{M} \otimes \mathcal{A}}(u,\alpha)} P(u, \alpha, u') \cdot p_{u'} \tag{G.1l}$$

• backward reachability of $M = \bigcup_{i=1}^{n} M_i$ within the subsystem:

$$\forall u = (s,q) \in U \ \forall \alpha \in Act \ \forall u' = (s',q') \in \text{succ}_{\mathcal{M} \otimes \mathcal{A}}\big(u, \alpha\big): \quad t_{u,u'}^M \leq x_{s'} + (1 - \sigma_{u,\alpha}) \tag{G.1m}$$

$$\forall u \in U \ \forall \alpha \in Act \ \forall u' \in \text{succ}_{\mathcal{M} \otimes \mathcal{A}}(u, \alpha): \quad r_u^M < r_{u'}^M + (1 - t_{u,u'}^M) + (1 - \sigma_{u,\alpha}) \tag{G.1n}$$

$$\forall u = (s,q) \in U \ \forall \alpha \in Act: \quad (1 - \sigma_{u,\alpha}) + \sum_{i=1}^{n} m_u^i + \sum_{u' \in \text{succ}_{\mathcal{M} \otimes \mathcal{A}}\left(u,\alpha\right)} t_{u,u'}^M \geq x_s \ . \tag{G.1o}$$

Lemma 19 *The MILP formulation* (G.1a)–(G.1o) *is sound.*

Proof. Assume a satisfying assignment ν of the MILP (G.1a)–(G.1o). We define $\mathcal{M}' = (S', s_{\text{init}}, Act, P', L')$ with $S' = \{s \in S \mid \nu(x_s) = 1\}$,

$$P'(s, \alpha, s') = \begin{cases} P(s, \alpha, s'), & \text{if } \exists q \in Q : \nu(\sigma_{(s,q),\alpha}) = 1, \\ 0, & \text{otherwise,} \end{cases}$$

and $L'(s) = L(s)$ for all $s, s' \in S'$ and $\alpha \in Act$. We show that $\mathcal{M}'$ is an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq \lambda}(\mathcal{L})$, with a maximal probability to satisfy $\mathcal{L}$ under all MCSs.

1. We show that $\mathcal{M}'$ is a subsystem of $\mathcal{M}$. From (G.1i) we imply $\nu(p_{(s,q)_{\text{init}}}) > \lambda \geq 0$. Using (G.1b) and (G.1k) we get that $\nu(x_{s_{\text{init}}}) = 1$, i. e., $s_{\text{init}} \in S'$. The other conditions for $\mathcal{M}'$ being a subsystem of $\mathcal{M}$ are straightforward by the definition of $\mathcal{M}$.
2. Using (G.1b) we observe that $\big|\{\alpha \in Act \mid \nu(\sigma_{u,\alpha}) = 1\}\big| \leq 1$ for all $u \in U$. Therefore the deterministic memoryless scheduler σ for $\mathcal{M}' \otimes \mathcal{A}$ with $\sigma(u)(\alpha) = \nu(\sigma_{u,\alpha})$ for all $u \in S' \times Q$ is well-defined, and it induces a DTMC $\mathcal{D}' = (\mathcal{M}' \otimes \mathcal{A})^\sigma$ with state set $U' = S' \times Q$. In the following we use the notation $\text{def}(\sigma) = \{u \in U' \mid \exists \alpha \in Act : \sigma(u)(\alpha) = 1\}$. We show that $\Pr^u_{\mathcal{D}'}(\Diamond\, accept) = \nu(p_u)$ for all states in $u \in U'$ (where all states in all accepting end components of $\mathcal{D}'$ are labeled with *accept*).
   For $i = 1, \ldots, n$ let $M_i = \{u \in U \mid \nu(m_u^i) = 1\}$ and $M = \bigcup_{i=1}^{n} M_i$. Using

$$\nu(m_u^i) \overset{\text{(G.1j)}}{\leq} \nu(p_u) \overset{\text{(G.1k)}}{\leq} \sum_{\alpha \in Act} \nu(\sigma_{u,\alpha}) \overset{\text{(G.1b)}}{\leq} \nu(x_s) \tag{G.2}$$

   for all $u = (s,q) \in U$ and $i = 1, \ldots, n$, we have that $M \subseteq U'$.

   - We first show $\Pr^u_{\mathcal{D}'}(\Diamond\, accept) = \nu(p_u) = 1$ for all $u \in M$.
     We show that all prerequisites of Lemma 1 on page 18 are satisfied for each M_i as state set and the A_i-states in M_i as target states. The constraints (G.1j)–(G.1k) assure $M_i \subseteq \text{def}(\sigma)$, i. e., that the scheduler σ selects an action for all M_i-states, for which by (G.1c) it holds that

$$\sum_{u' \in \text{succ}_{\mathcal{M} \otimes \mathcal{A}}(u, \sigma(u))} P'(u, \sigma(u), u') = 1 \, .$$

M_i is closed under successors w. r. t. the actions selected by σ because of (G.1d). Furthermore, M_i does not contain any R_i-states according to (G.1e). Given the assignment of $\sigma_{u,\alpha}$, constraints (G.1f)–(G.1h) are backward reachability constraints with the A_i-states as the target states. According to Lemma 13 on page 38, an assignment ν is satisfying these constraints iff from all states in M_i an A_i-state state is reachable inside M_i. Therefore by Lemma 1 it follows that $\Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept}) = 1$ for all states $u \in \bigcup_{i=1}^{n} M_i$, which coincides with $\nu(p_u)$ because of (G.1j). I. e., $\Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept}) = \nu(p_u) = 1$ for all $u \in M$.

- Now we show that $\Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept}) = \nu(p_u)$ for all $u \in U' \setminus M$.
  Constraints (G.1m)–(G.1o) assure that for all states $u \in U' \setminus M$ either $u \notin \mathrm{def}(\sigma)$ or a state in M is reachable from u (cf. Lemma 13).
  For states $u \in U \setminus \mathrm{def}(\sigma)$ without any action selected by σ, (G.1k) implies $\nu(p_u) = 0 = \Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept})$. Assume that these states and their connected (incoming) transitions are removed from $\mathcal{D}'$.
  Note that for non-selected states $u \in U \setminus U'$, the constraints (G.1b) and (G.1k) enforce $\nu(p_u) = 0$. Remember furthermore that $\nu(p_u) = 1$ for each $u \in M$. Therefore, for each $u \in (U' \setminus M) \cap \mathrm{def}(\sigma)$ and $\alpha \in \mathit{Act}$ with $\sigma(u) = \alpha$, according to (G.1l) it holds that

$$\nu(p_u) \leq \sum_{u' \in (U' \cap \mathrm{def}(\sigma)) \setminus M} P'(u, \alpha, u') \cdot \nu(p_{u'}) + \sum_{u' \in M} P'(u, \alpha, u')\,.$$

  Lemma 5 applied to the state set $U' \cap \mathrm{def}(\sigma)$ and target set M gives us $\nu(p_u) \leq \Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept})$. According to the objective function (G.1a), ν maximizes the probability $p_{(s,q)_{\mathrm{init}}}$. Lemma 2 states that the maximal solution satisfies $\Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept}) = \nu(p_u)$ for all $u \in (U' \cap \mathrm{def}(\sigma)) \setminus M$.
  We conclude that $\nu(p_u) = \Pr^u_{\mathcal{D}'}(\Diamond\, \mathit{accept})$ for all $u \in U'$.

3. Above we have shown that $\nu(p_{(s,q)_{\mathrm{init}}}) = \Pr^{(s,q)_{\mathrm{init}}}_{\mathcal{D}'}(\Diamond\, \mathit{accept})$, and by (G.1i) we have that $\nu(p_{(s,q)_{\mathrm{init}}}) > \lambda$. Thus $\Pr^{(s,q)_{\mathrm{init}}}_{\mathcal{D}'}(\Diamond\, \mathit{accept}) > \lambda$. Using Theorem 2 we get that $\Pr^{s_{\mathrm{init}}}_{\mathcal{M}^\sigma}(\mathcal{L}) = \Pr^{(s,q)_{\mathrm{init}}}_{\mathcal{D}'}(\Diamond\, \mathit{accept}) > \lambda$, i. e., $\mathcal{M}'$ is critical.
4. We show that $\mathcal{M}'$ is minimal. Assume the opposite. Then there is an MCS $\mathcal{M}''$ for $\mathcal{M}$ and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with state set $S'' \subseteq S$ such that $|S''| < |S'|$. Since $\mathcal{M}''$ is an MCS, there is a deterministic memoryless scheduler σ for $\mathcal{M}'' \otimes \mathcal{A}$ such that $\Pr^{(s,q)_{\mathrm{init}}}_{(\mathcal{M}'' \otimes \mathcal{A})^\sigma}(\Diamond\, \mathit{accept}) > \lambda$.
   In all constraints (G.1a)–(G.1o) we syntactically replace (i) x_s by 1 if $s \in S''$ and by 0 otherwise, (ii) m^i_u by 1 if $u \in S'' \times Q$ is in an end component of $\mathcal{M}'' \otimes \mathcal{A}$ accepting for the ith accepting condition and by 0 otherwise, and $\sigma_{u,\alpha}$ by $\sigma(u)(\alpha) \in \{0, 1\}$.
   Lemma 2 applied to S'' implies that the constraint system resulting from the above substitution has a satisfying assignment; following the argumentation in item (2) above we get that this assignment maps $\Pr^{s_{\mathrm{init}}}_{\mathcal{D}_{S''}}(\mathcal{L}) > \lambda$ to $p_{(s,q)_{\mathrm{init}}}$, thus also satisfying (D.1b)). However, for this satisfying assignment the number of positive x_s variables is smaller than for ν, which contradicts our assumption that ν minimizes the objective function.
5. It remains to show that the probability to satisfy $\mathcal{L}$ from s_{init} in $\mathcal{M}'$ is maximal among all MCSs. This proof is analogous to the previous item. Assume the opposite. Then there is some MCS $\mathcal{M}''$ of $\mathcal{D}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ with state set $S'' \subseteq S$ such that the probability to satisfy $\mathcal{L}$ in the initial state is higher in $\mathcal{M}''$ as in $\mathcal{M}'$.
   We apply the same replacement as above to the constraint system (G.1a)–(G.1o) to get a satisfying assignment μ inducing $\mathcal{M}''$. Since $\mathcal{M}'$ and $\mathcal{M}''$ are both minimal, $\sum_{s \in S} \nu(x_s) = \sum_{s \in S} \mu(x_s)$, however $\nu(p_{(s,q)_{\mathrm{init}}}) < \mu(p_{(s,q)_{\mathrm{init}}})$, contradicting the optimality of ν. □

Lemma 20 *The MILP formulation* (G.1a)–(G.1o) *is complete.*

Proof. Let $\mathcal{M}' = (S', s_{\mathrm{init}}, \mathit{Act}, P', L')$ be an MCS of $\mathcal{M}$ for $\mathcal{P}_{\leq\lambda}(\mathcal{L})$, in which the probability to satisfy $\mathcal{L}$ in the initial state is the highest among all MCSs.

Since the subsystem $\mathcal{M}'$ is critical, there is a memoryless deterministic scheduler σ for $\mathcal{M}' \times \mathcal{A}$ such that $\Pr^{(s,q)_{\text{init}}}_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, \mathit{accept}) > \lambda$. Let B be the set of accepting BSCCs of $(\mathcal{M} \otimes \mathcal{A})^\sigma$ and $M_i = \bigcup\{C \in B \mid C \cap R_i = \emptyset \wedge C \cap A_i \neq \emptyset\}$ for $i = 1, \ldots, m$.
We define the following partial assignment ν:

- $\nu(x_s) = 1$ iff $s \in S'$
- $\nu(\sigma_{u,\alpha}) = 1$ iff $u = (s,q) \wedge s \in S' \wedge \sigma(u)(\alpha) = 1$, and
- $\nu(p_u) = \Pr^u_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond \mathit{accept})$,
- $\nu(m^i_u) = 1$ iff $u \in M_i$.

Now we show that there is a total extension of this assignment that satisfies all constraints:

(G.1a) The defined assignment minimizes this function, since the MCS is minimal with maximal probability to satisfy $\mathcal{L}$ under all MCSs.

(G.1b) This constraint is satisfied since we do not select any action for states $u = (s,q)$ with $s \notin S'$ and σ selects exactly one action for each state $u = (s,q)$ with $s \in S'$.

(G.1c) Since all states of M_i are contained in a BSCC, and—for all states in a BSCC—the probability that a successor state is also in a BSCC is 1, this constraint is fulfilled.

(G.1d) For states u outside M_i and for actions not chosen by σ, the constraint is satisfied because in these cases $(2 - m^i_u - \sigma_{u,\alpha}) \geq 1$. For states $u = (s,q)$ with $s \in S'$ and action $\alpha = \sigma(u)$, $\nu(m^i_{u'}) = 1$ is required for all successor states u' of u. This is the case since M_i is a union of BSCCs.

(G.1e) In the definition of M_i we have required that $M_i \cap R_i = \emptyset$. Therefore this constraint is fulfilled.

(G.1f)–(G.1h) Each accepting BSCC in M_i contains by construction a state from A_i. Since in a BSCC each state is reachable from each state, we can apply Lemma 12 to obtain a satisfying assignment for these backward reachability constraints.

(G.1i) $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{(s,q)_{\text{init}}}_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, \mathit{accept}) > \lambda$ holds since the subsystem is critical.

(G.1j) For target states, which are the states in the accepting BSCCs, the reachability probability is one.

(G.1k) Since for each deadlocking state u without any outgoing transition $\nu(p_u) = \Pr^u_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, \mathit{accept}) = 0$ holds, the inequality is trivially satisfied. For non-deadlocking states, this inequation puts no constraints on the probability values, thus it holds also in that case.

(G.1l) For states from an M_i this constraint is fulfilled trivially, since the right-hand side evaluates at least to one. The case for $\sigma(u) \neq \alpha$ is similarly straightforward. The reachability probabilities for the remaining states which can reach the accepting BSCCs satisfy the equality

$$p_u = \sum_{u' \in \text{succ}_{\mathcal{M}' \otimes \mathcal{A}}(u,\alpha)} P'(u,\alpha,u') \cdot p_u$$

and therefore satisfy also this constraint. For the remaining states $\nu(p_u) = 0$ holds, also satisfying the constraint.

(G.1m)–(G.1o) These are the backward reachability constraints ensuring reachability of the accepting BSCCs. We distinguish different cases:

- $s \notin S'$: Set $\nu(t^M_{u,u'}) = 0$ for all $q \in Q$, $u = (s,q)$, $u' \in \text{succ}_{\mathcal{M}' \otimes \mathcal{A}}(u)$ and $\nu(r^M_u) = 0$. Then all three constraints are fulfilled.

- $s \in S'$, but from u no accepting BSCC can be reached. Choose $\nu(t^M_{u,u'}) = 0$ and $\nu(r^M_u) = 0$ as in the previous case. Since $\nu(\sigma_{u,\alpha}) = 0$ for all $\alpha \in Act$, the three constraints are satisfied.
- $s \in S'$ and from u a BSCC can be reached. According to Lemma 12 we can find a satisfying assignment for these backward reachability constraints.

We have shown that the constructed assignment ν satisfies all constraints of the MILP. □

Theorem 9 *The MILP formulation* (G.1a)–(G.1o) *is sound and complete.*

Proof. The MILP formulation is sound by Lemma 19 and complete by Lemma 20. □