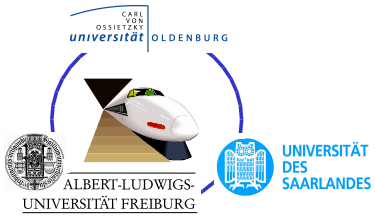


AVACS – Automatic Verification and Analysis of Complex Systems

REPORTS

of SFB/TR 14 AVACS

Editors: Board of SFB/TR 14 AVACS

Minimal Counterexamples for Refuting ω-Regular Properties of Markov Decision Processes

– Extended Version –

by

Ralf Wimmer Nils Jansen Erika Ábrahám
Joost-Pieter Katoen Bernd Becker

AVACS Technical Report No. 88
September 2012
ISSN: 1860-9821

Publisher: Sonderforschungsbereich/Transregio 14 AVACS
(Automatic Verification and Analysis of Complex Systems)
Editors: Bernd Becker, Werner Damm, Martin Fränzle, Ernst-Rüdiger Olderog,
Andreas Podelski, Reinhard Wilhelm
ATRs (AVACS Technical Reports) are freely downloadable from `www.avacs.org`

Author(s) contact: Ralf Wimmer (`wimmer@informatik.uni-freiburg.de`).

Minimal Counterexamples for Refuting ω-Regular Properties of Markov Decision Processes

– Extended Version –

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

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

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., as 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]:

☆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", and the DFG project "CEBug – Counterexample Generation for Stochastic Systems using Bounded Model Checking".

*Corresponding author:
Ralf Wimmer
Lehrstuhl für Rechnerarchitektur
Albert-Ludwigs-Universität Freiburg
Georges-Köhler-Allee 51
79110 Freiburg im Breisgau, Germany
Phone: +49 761 203 8179
Fax: +49 761 203 8142

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)

September 7, 2012

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.

Other uses of counterexamples include automated refinement of system abstractions as used in the successful CEGAR (counterexample-guided abstraction refinement) framework [2–4].

Research on counterexample generation in model checking is abundant [5–9]. 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 [7] 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 very 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., [10]). 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 [11] 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 [12] and LiQuoR [13]; DTMC model checking is supported by, e. g., MRMC [14] and FMurphi [15]. The PRISM set of case studies [16] 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 [17–20] 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 [21, 22].

For DTMCs, it was shown in [17] 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-path 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 [17], rendering the counterexample practically unusable for debugging purposes. Different proposals have been made to alleviate this problem. To mention a few, [17] represents the path set as a (weighted) regular expression, [18] detects loops on paths, and [19] 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 [23, 24] and of *critical subsystems* have been proposed in [25, 26]. 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 [25] apply best first search to identify a critical subsystem, while Jansen *et al.* [26] 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` [27] and `COMICS` [28], 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 [29], 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, provide a minimal sub-MDP whose maximal probability to satisfy the property exceeds λ. This problem has been proven to be NP-complete [4]. We first consider DTMCs and provide two formulations to this MCS problem: A SAT-modulo theories (SMT) formulation and a mixed integer linear program (MILP). 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 benchmark case studies are provided, which show the effectiveness of our approach and our optimizations. We show that our MILP approach yields often considerably more compact counterexamples than the heuristic methods [25, 26]. 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. 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 [30] and [31] 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 appendix.

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 finite or 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 $\text{Distr}(S)$.

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 finite or 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*.*

If $s \in S$ is the current state of an MDP $\mathcal{M}$, 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

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

We sometimes skip the index $\mathcal{M}$ when it is clear from the context.

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

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

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* $\sigma : \text{Paths}^{\text{fin}}_{\mathcal{M}}(s_{\text{init}}) \to \text{Distr}(Act)$*. We denote the set of schedulers on* $\mathcal{M}$ *by* $\text{Sched}_{\mathcal{M}}$*.*

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 DTMCs. 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 $\pi = s_0 s_1 \dots s_n \in \text{Paths}^{\text{fin}}_{\mathcal{D}}(s_0)$ is the set $\text{cyl}(\pi) = \{\pi' \in \text{Paths}^{\text{inf}}_{\mathcal{D}}(s_0) \mid \pi \text{ is a prefix of } \pi'\}$ of all infinite extensions of π. For the DTMC $\mathcal{D}$ and a state $s \in S$, a *probability space* $(\Omega, \mathcal{F}, \Pr^s_{\mathcal{D}})$ can be defined as follows:

[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 $L(s_\perp) = \emptyset$.

The *sample space* $\Omega = \text{Paths}_{\mathcal{D}}^{\inf}(s)$ is the set of all infinite paths starting in s. 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 $\text{Paths}_{\mathcal{D}}^{\text{fin}}(s)$, i. e., it is the closure of the cylinder sets under complement and countable union containing Ω. The *probability measure* $\Pr_{\mathcal{D}}^{s} : \mathcal{F} \to [0,1] \subseteq \mathbb{R}$ is the unique measure extending $\Pr_{\mathcal{D}}^{s}\big(\text{cyl}(s_0 s_1 \dots s_n)\big) = \prod_{i=0}^{n-1} P(s_i, s_{i+1})$ to the whole σ-algebra [32]. A set Π of paths is *measurable* iff $\Pi \in \mathcal{F}$.

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

In the following, we do not need schedulers in their full generality, 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 [33, Lemma 10.102]. The distribution assigned to a finite path by a memoryless scheduler depends only on the last state of the path. Note that for finite MDPs this yields a finite DTMC. 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 $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP. A scheduler σ for $\mathcal{M}$ is* memoryless *iff for all $\pi, \pi' \in \text{Paths}_{\mathcal{M}}^{\text{fin}}$ with $\text{last}(\pi) = \text{last}(\pi')$ we have that $\sigma(\pi) = \sigma(\pi')$. A scheduler σ for $\mathcal{M}$ is* deterministic *iff for all $\pi \in \text{Paths}_{\mathcal{M}}^{\text{fin}}$ there is an $\alpha \in Act$ such that $\sigma(\pi)(\alpha) = 1$.*

Memoryless deterministic schedulers can be regarded as functions $\sigma : S \to Act$. The induced DTMC of a memoryless scheduler σ is bisimilar to $\mathcal{M}^{\sigma,\text{md}} = (S, s_{\text{init}}, P', L)$ with $P'(s, s') = P(s, \sigma(s), s')$. If not stated differently, in the following we always refer to $\mathcal{M}^{\sigma,\text{md}}$ (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 $\gamma_i \subseteq \text{AP}$ 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 $a \in \text{AP}$ is the linear-time property:*

$$\Diamond\, a = \{\gamma_1\gamma_2\dots \in (2^{\text{AP}})^{\omega} \mid \exists i \geq 0 : a \in \gamma_i\}.$$

A path π of a DTMC $\mathcal{D}$ satisfies a reachability property $\Diamond\, a$ with $a \in \text{AP}$, written $\pi \models \Diamond\, a$, if $\text{trace}(\pi) \in \Diamond\, a$. We are interested in the total probability $\Pr_{\mathcal{D}}^{s_{\text{init}}}(\Diamond\, a)$ 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 $\Pr_{\mathcal{D}}^{s_{\text{init}}}(\Diamond\, a)$ is less than or equal to 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 $\Pr_{\mathcal{D}}^{s}(\Diamond\, a)$ 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 [34].

2.2.2. Model Checking Reachability Properties

Prior to checking $\mathcal{D} \models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$, the states that can neither reach an a-state nor can be reached from the initial state s_{init} can be safely removed from the DTMC $\mathcal{D}$. Let $S_{\mathcal{D}}^{\text{rel}(a)}$ denote the set of remaining states, referred to as *relevant* states of $\mathcal{D}$ for proposition $a \in \text{AP}$. The set $S_{\mathcal{D}}^{\text{rel}(a)}$ can be determined in linear time by a backward reachability analysis from the set of a-states and a forward reachability analysis from s_{init}. After this pre-processing, property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$ for DTMC $\mathcal{D}$ is checked by computing $\Pr_{\mathcal{D}}^{s}(\Diamond\, a)$ for all states $s \in S$ and comparing this probability for the initial state with the bound λ. The probabilities $p_s = \Pr_{\mathcal{D}}^{s}(\Diamond\, a)$ are obtained as the unique solution of the following linear equation system [33, p. 760]:

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

For MDPs, the procedure is similar. As a first step for reachability properties on MDPs, irrelevant states are removed.

Definition 6 (Relevant states of MDPs) *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP and $a \in AP$. Then*

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

is the set of relevant *for proposition a. If $s \notin S_{\mathcal{M}}^{rel(a)}$, then s is called* irrelevant *for a.*

The set of irrelevant states can be computed in linear time by a backward reachability analysis on $\mathcal{M}$ [33, Algorithm 46]. The removal of irrelevant states does not affect the reachability probabilities. 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^*, say, that maximizes the reachability probability for $\Diamond\, a$ and check whether $\Pr_{\mathcal{M}^{\sigma^*}}^{s_{\text{init}}}(\Diamond\, a) \leq \lambda$ [33, 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:

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

This equation system can be transformed into a linear optimization problem that yields the maximal reachability probability together with an optimal scheduler [33, 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 such properties.

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 [33, 35–37]. We use deterministic Rabin automata.

Definition 7 (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.*

A *run* r of $\mathcal{A}$ is a state sequence $q_0q_1q_2\ldots \in Q^\omega$ 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 $F = \{(R_i, A_i) \mid i = 1, \ldots, n\}$, 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 8 (ω-Regular property, Safra [38]) *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[3] $\Pr_{\mathcal{D}}^{s_{\text{init}}}(\mathcal{L})$ 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 9 (Product automaton) *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 $F = \{(R_i, A_i) \mid i = 1, \ldots, n\}$. 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*

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

For checking ω-regular properties, we first concentrate on DTMCs and show afterwards how to handle arbitrary MDPs. 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 10 (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 $s_0s_1\ldots s_n \in \text{Paths}_{\mathcal{D}}^{\text{fin}}$ with $s_0 = s$, $s_n = s'$ and $s_i \in S'$ for all $i = 1, \ldots, n$.*
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. *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. *The set of* input states *of S' is defined as $\text{In}(S') = \{s \in S' \mid \text{pred}_{\mathcal{D}}(s) \cap (S \setminus S') \neq \emptyset\}$.*
5. *The set of* output states *of S' is defined as $\text{Out}(S') = \{s \in S \setminus S' \mid \text{pred}_{\mathcal{D}}(s) \cap S' \neq \emptyset\}$.*

[3] Again, in $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ we overload $\mathcal{L}$ to denote the set $\{\pi \in \text{Paths}_{\mathcal{D}}^{\text{inf}}(s) \mid \text{trace}(\pi) \in \mathcal{L}\}$ 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 [34].

The SCC structure of a directed graph can be determined by Tarjan's algorithm in linear time [39]. The input states of an SCC S' are those states in S' through which paths enter S'. Analogously, the output states of an SCC S' are those states outside S' through which paths exit S'. As the set of output states of a BSCC is empty, the probability to visit each state in a BSCC infinitely often is one.

Definition 11 (Accepting BSCC) *Let DTMC $\mathcal{D} = (S, s_{\text{init}}, P, L)$, DRA $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ and $\mathcal{D} \otimes \mathcal{A} = (S \times Q, (s,q)_{\text{init}}, Act, P', L')$ their product. BSCC $B \subseteq S \times Q$ of $\mathcal{D} \otimes \mathcal{A}$ is called* accepting *iff there are some $(R_i, A_i) \in F$ such that $A_i \in L'(s,q)$ for some $(s,q) \in B$ and $R_i \notin L'(s',q')$ for all $(s',q') \in B$.*

We introduce the proposition *accept* and extend the labelling 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 ([35]) *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:*

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

For MDPs, the corresponding notion to a BSCC is a so-called *end component*.

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

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 $\text{succ}_{\mathcal{M}}(s, \alpha) \subseteq S'$ holds for all states $s \in S'$ and actions $\alpha \in A(s)$.*
2. *A sub-MDP S' with action set A is an* end component *of $\mathcal{M}$ if the directed graph $G = (S', V)$ with $V = \{(s,s') \in S' \times S' \mid \exists \alpha \in A(s) : s' \in \text{succ}_{\mathcal{M}}(s,\alpha)\}$ 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 with $F = \{(R_i, A_i) \mid i = 1, \ldots, n\}$ and $B \subseteq S \times Q$ an end component of $\mathcal{M} \otimes \mathcal{A}$. B is* accepting *if there is $i \in \{1, \ldots, n\}$ such that for all $(s,q) \in B : R_i \notin L'(s,q)$ and there is $(s,q) \in B : A_i \in L'(s,q)$.*

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 $\Pr_{\mathcal{M}^{\sigma^*}}^{s}(\mathcal{L}) = \max_{\sigma \in \text{Sched}_{\mathcal{M}}} \Pr_{\mathcal{M}^{\sigma}}^{s}(\mathcal{L})$ is at most λ.

Theorem 2 ([35]) *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:*

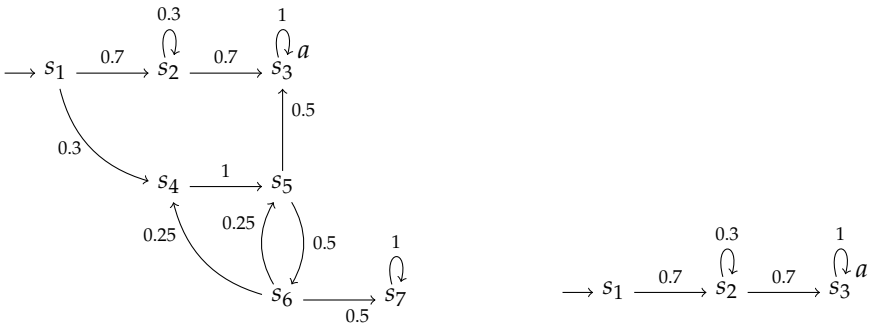
$$\Pr_{\mathcal{M}^{\sigma^*}}^{s_{\text{init}}}(\mathcal{L}) = \Pr_{\mathcal{M}^{\sigma^*} \otimes \mathcal{A}}^{(s,q)_{\text{init}}}(\Diamond \text{ accept}) \,.$$

2.4. Minimal Critical Subsystems

Let $\mathcal{M}$ be an MDP and consider $\mathcal{P}_{\leq \lambda}(\mathcal{L})$ for 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 13 (Minimal critical subsystem) *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP.*

1. *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. *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 minimal number of states among all critical subsystems.*



(a) Example DTMC

(b) Minimal Critical Subsystem

Figure 1: DTMC $\mathcal{D}$ and Minimal Critical Subsystem $\mathcal{D}'$ for $\mathcal{P}_{\leq 0.7}(\Diamond\, a)$.

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 *We illustrate the concept of an MCS by means of a DTMC. Consider the DTMC $\mathcal{D}$ in Figure 1(a) with initial state s_1 and the reachability property $\mathcal{P}_{\leq 0.7}(\Diamond\, a)$. State s_3 is the only target state. The overall probability of reaching s_3 is 0.9 which means the property is violated. An* MCS *for $\mathcal{D}$ and $\mathcal{P}_{\leq 0.7}(\Diamond\, a)$ is given in Figure 1(b).*

2.5. SAT-Modulo-Theories

SAT-modulo-theories (SMT) [40] refers to a generalization of the classical propositional satisfiability problem (SAT). Compared to SAT problems, in an SMT formula atomic propositions may 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 [41]. Several tools for solving SMT formulae over LRA are available, e. g., Z3 [42], CVC [43], and MathSAT [44].

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 [45, Problem MP1].

Definition 14 (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 efficiently solved. MILPs are widely used in operations research, hardware-software code design, and numerous other applications. Efficient open source as well as commercial implementations are available like SCIP [46] or CPLEX [47] by IBM. We refer the reader to, e. g., [48] 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 DTMC $\mathcal{D} = (S, s_{\text{init}}, P, L)$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$, we provide an SMT formula over LRA whose satisfying variable assignments correspond to the critical subsystems (of arbitrary size) of $\mathcal{D}$. Let $T_a = \{s \in S \mid a \in L(s)\}$ be the set of target states. We assume that DTMC $\mathcal{D}$ contains only relevant states for a. 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 $s \in S$ for keeping track of the reachability probability of a target state from s within the subsystem. The SMT formulation reads:

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

$$\text{such that}$$

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

$$\forall s \in S \setminus T_a: \quad (x_s = 0 \wedge p_s = 0) \oplus \Big(x_s = 1 \wedge p_s = \sum_{s' \in \text{succ}(s)} P(s,s') \cdot p_{s'}\Big) \tag{1c}$$

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

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 summand in line 1b). Target states that are contained in the subsystem have probability one (second summand in 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 linear in the size of $\mathcal{D}$. The correctness of the SMT formulation is stated as follows (and proven in Appendix A):

Theorem 3 *The SMT formulation* (1a)–(1d) *yields an MCS for DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

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 solution 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 solution for larger DTMCs is rather time-consuming. This is mainly due to the high number of disjunctions in the formula. This triggers relatively few implications, forcing the solver to attempt many different cases while searching for a solution.

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

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

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

$$\forall s \in S \setminus T_a: \quad p_s \leq \sum_{s' \in \text{succ}(s)} P(s,s') \cdot p_{s'} \tag{2d}$$

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

The probability p_s of a state $s \in T_a$ is 1 iff the state is contained in the MCS, i. e., iff $x_s = 1$ (cf. line (2b)). Analogously, for every state $s \in S \setminus T_a$ 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 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 objective function (2a).

Theorem 4 *The MILP formulation* (2a)–(2e) *yields an MCS for DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

A proof of this theorem can be found in Appendix B. As our experiments revealed that the MILP formulation yields substantial reductions to 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.

3.3. 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 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.

$$\forall s \in S \setminus T_a: \quad -x_s + \sum_{s' \in \text{succ}(s) \setminus \{s\}} x_{s'} \geq 0 \tag{3a}$$

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

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., $x_{s'} = 1$) yielding the sum over all successor/predecessor states to exceed one.

3.3.2. SCC Constraints

The forward/backward cuts do not encode the complete reachability of target states from the initial state: 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' (except the initial state s_{init}) have to be reached through one of its input states $\text{In}(S')$. Therefore we ensure that a state of an SCC can only be selected if at least one 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 do not contain a target state have to lead through one of the SCC's output states $\text{Out}(S')$. 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 solution contains only states on a path from the initial state to target states. This is still only ensured by minimizing the objective function.

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

$$\forall \text{SCC } S', S' \cap T_a = \emptyset : \quad \sum_{s \in S'} x_s \;\leq\; \left|S'\right| \cdot \sum_{s \in \text{Out}(S')} x_s \,. \tag{4b}$$

3.3.3. Reachability Constraints

If an SCC is selected which is connected to the input 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 ensures complete reachability. An assignment will only satisfy these additional constraints if all states are reachable from the initial state and a target state is reachable. 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 $r_s^{\rightarrow} \in [0,1] \subseteq \mathbb{R}$ for each state s. These variables define a partial order on the states. For each transition $(s, s') \in E_{\mathcal{D}}$ we introduce a characteristic integer variable $t_{s,s'}^{\rightarrow} \in [0,1] \subseteq \mathbb{Z}$. The constraints for forward reachability are as follows:

$$\forall s \in S \; \forall s' \in \text{succ}(s) : \quad 2t_{s,s'}^{\rightarrow} \leq x_s + x_{s'} \tag{5a}$$

$$\forall s \in S \; \forall s' \in \text{succ}(s) : \quad r_s^{\rightarrow} < r_{s'}^{\rightarrow} + (1 - t_{s,s'}^{\rightarrow}) \tag{5b}$$

$$\forall s \in S \setminus \{s_{\text{init}}\} : \quad (1 - x_s) + \sum_{s' \in \text{pred}(s)} t_{s',s}^{\rightarrow} \geq 1 \,. \tag{5c}$$

If $s \in S$ is selected and reachable from s_{init}, then there is a path $s_{\text{init}} = s_0 \ldots s_n = s$ such that $r_{s_i}^{\rightarrow} < r_{s_{i+1}}^{\rightarrow}$ 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 $t_{s,s'}^{\rightarrow} = 1$ connects selected states s and s' (line 5a). If $t_{s,s'}^{\rightarrow} = 1$, $r_s^{\rightarrow} < r_{s'}^{\rightarrow}$ 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, an incoming transition $t_{s',s}^{\rightarrow}$ 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 $r_s^{\leftarrow}$ for all states $s \in S$ and variables $t_{s,s'}^{\leftarrow}$ for all transitions $(s, s') \in E_{\mathcal{D}}$:

$$\forall s \in S \; \forall s' \in \text{succ}(s) : \quad 2t_{s,s'}^{\leftarrow} \leq x_s + x_{s'} \tag{6a}$$

$$\forall s \in S\ \forall s' \in \mathrm{succ}(s): \quad r_s^{\leftarrow} < r_{s'}^{\leftarrow} + (1 - t_{s,s'}^{\leftarrow}) \tag{6b}$$

$$\forall s \in S \setminus T_a: \quad (1 - x_s) + \sum_{s' \in \mathrm{succ}(s)} t_{s,s'}^{\leftarrow} \geq 1\,. \tag{6c}$$

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 fact that the above optimizations are correct follows from the following result, which is proven in Appendix C:

Theorem 5 *The SMT formulation* (1a)–(1d) *together with any (combination) of the three above optimizations yields an MCS for DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

3.4. ω-Regular Properties

We now generalize our MILP formulation to arbitrary ω-regular properties. Let $\mathcal{L}$ be such a property and DRA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. As before, we want to compute an MCS $\mathcal{D}'$ for which $\Pr_{\mathcal{D}'}^{s_{\text{init}}}(\mathcal{L}) > \lambda$ 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 $\mathcal{A}$ as in Definition 9 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 like to 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 probability for all states of the product automaton is needed, we use 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

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

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

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

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

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

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 correctness of the MILP formulation is proven in Appendix E, and reads as follows:

Theorem 6 *The MILP formulation* (7a)–(7f) *yields an MCS for DTMC $\mathcal{D}$ and ω-regular property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$.*

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 a bit 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[4], for MDPs it holds:

Theorem 7 ([4]) *Let $\mathcal{M}$ be an MDP with $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ and $k \in \mathbb{N}$. Then the problem to decide if 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 $T_a = \{a \in S \mid a \in L(s)\} \subseteq S$ 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 $s \in S \setminus T_a$ 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 $\sigma_{a,\alpha}$) 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_a to be reachable from s. However, for a relevant state s, T_a 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 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

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

be the set of *problematic* states in MDP $\mathcal{M}$ for proposition a. If $s \notin S_{\mathcal{M}}^{\text{probl}(a)}$ then s is called *unproblematic* for a. Our additional constraints prevent from obtaining a scheduler that chooses the "wrong" actions in problematic states (i. e., actions that yield the T_a 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 $Act_{\mathcal{M}}^{\text{probl}(a)} = \left\{(s,\alpha) \in S \times Act \mid \text{succ}_{\mathcal{M}}(s,\alpha) \subseteq S_{\mathcal{M}}^{\text{probl}(a)}\right\}$ 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 $s \in S_{\mathcal{M}}^{\text{probl}(a)}$ 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 $s' \in S_{\mathcal{M}}^{\text{probl}(a)}$ where $(s,\alpha) \in Act_{\mathcal{M}}^{\text{probl}(a)}$ for an action $\alpha \in Act$. We thus propose the following MILP formulation:

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

$$\text{such that}$$

[4]The problem of finding an MCS for a PCTL-formula on DTMCs is NP-complete [4]. This result however exploits nested PCTL-formulas.

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

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

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

$$\forall s \in S \setminus T_a : \quad (1 - x_s) + \sum_{\alpha \in Act} \sigma_{s,\alpha} = 1 \tag{8e}$$

$$\forall s \in S \setminus T_a \; \forall \alpha \in Act : \quad p_s \leq (1 - \sigma_{s,\alpha}) + \sum_{s' \in \text{succ}_{\mathcal{M}}(s,\alpha)} P(s,\alpha,s') \cdot p_{s'} \tag{8f}$$

$$\forall (s,\alpha) \in Act_{\mathcal{M}}^{\text{probl}(a)} \; \forall s' \in \text{succ}_{\mathcal{M}}(s,\alpha) : \quad 2t_{s,s'}^{\leftarrow} \leq x_s + x_{s'} \tag{8g}$$

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

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

The constraints (8a)–(8d) are the same as for DTMCs. Equation (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 correctness of the MILP formulation is captured by the following theorem; its proof is provided in Appendix G.

Theorem 8 *The MILP formulation* (8a)–(8i) *yields an MCS for MDP $\mathcal{M}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

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.

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 go 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 MDP $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ and DRA $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ with $F = \{(R_i, A_i) \mid i = 1, \ldots, n\}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$ for ω-regular property $\mathcal{L}$. The property of interest is $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ and assume $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\mathcal{L})$. Furthermore, we assume 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 [49]. 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.[5]

To simplify notation we use $U = S \times Q$, $u = (s, q)$, and $u' = (s', q')$. Let $n_{u,\alpha} = |\text{succ}_{\mathcal{M}\otimes\mathcal{A}}(u, \alpha)|$ denote the number of successor states of u under action α. We have characteristic 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:

[5] 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).

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. $\forall u \in M_i : \sum_{u' \in \mathrm{succ}(u,\sigma(u)) \cap M_i} P'(u, \sigma(u), u') = 1$,
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 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 $m^i_u = 1$ indicates that 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 $u \in U$.

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:

$$\forall (s,q) \in U : \quad (1 - x_s) + \sum_{\alpha \in Act} \sigma_{(s,q),\alpha} \leq 1 \tag{9b}$$

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

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

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

$$\forall u \in U\ \forall \alpha \in Act : \quad n_{u,\alpha} \cdot (2 - \sigma_{u,\alpha} - m^i_u) + \sum_{u' \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} m^i_{u'} \geq n_{u,\alpha} \tag{9e}$$

$$\forall u \in S \times R_i : \quad m^i_u = 0 \tag{9f}$$

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

$$\forall u \in U\ \forall \alpha \in Act\ \forall u' \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha) : \quad 2t^i_{u,u'} \leq m^i_u + m^i_{u'} + 2(1 - \sigma_{u,\alpha}) \tag{9g}$$

$$\forall u \in U\ \forall \alpha \in Act\ \forall u' \in \mathrm{succ}_{\mathcal{M}}(u,\alpha) : \quad r^i_u < r^i_{u'} + (1 - t^i_{u,u'}) + (1 - \sigma_{u,\alpha}) \tag{9h}$$

$$\forall u \in S \times (Q \setminus A_i)\ \forall \alpha \in Act : \quad (1 - m^i_u) + (1 - \sigma_{u,\alpha}) + \sum_{u' \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} t^i_{u,u'} \geq 1 \tag{9i}$$

• probability computation:

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

$$\forall i = 1, \ldots, n\ \forall (s,q) \in U : \quad x_s \geq m^i_{(s,q)} \tag{9k}$$

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

$$\forall (s,q) \in U : \quad p_{(s,q)} \leq x_s \tag{9m}$$

$$\forall u \in U\ \forall \alpha \in Act: \tag{9n}$$

$$p_u \leq (1-\sigma_{u,\alpha}) + \sum_{i=1}^{n} m_u^i + \sum_{u' \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha)} P(u,\alpha,u') \cdot p_{u'} \tag{9o}$$

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

$$\forall (s,q) \in U\ \forall \alpha \in Act\ \forall (s',q') \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}\big((s,q),\alpha\big): \quad 2t^M_{(s,q),(s',q')} \leq x_s + x_{s'} + 2(1-\sigma_{(s,q),\alpha}) \tag{9p}$$

$$\forall u \in U\ \forall \alpha \in Act\ \forall u' \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}(u,\alpha): \quad r^M_u < r^M_{u'} + (1-t^M_{u,u'}) + (1-\sigma_{u,\alpha}) \tag{9q}$$

$$\forall (s,q) \in U\ \forall \alpha \in Act:$$

$$(1-x_s) + (1-\sigma_{(s,q),\alpha}) + \sum_{i=1}^{n} m^i_{(s,q)} + \sum_{(s',q') \in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}\left((s,q),\alpha\right)} t^M_{(s,q),(s',q')} \geq 1\,. \tag{9r}$$

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, (9c) 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 (9d) 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 (9l) sets the probability of M_i-states to one.

Second we ensure in (9e) the closure of M_i under successors. If state u belongs to M_i (i. e., $m^i_u = 1$) 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 $n_{u,\alpha}(2 - \sigma_{u,\alpha} - m^i_u)$ is zero iff α is selected in u and $u \in M_i$. In this case the sum over the corresponding variables $m^i_{u'}$ of the successors u' of u has to be at least the number of the successors of u.

Equation (9f) 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 (9g)–(9i). 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 (9g)–(9i) (see also Appendix C).

The remaining constraints are analogous to the MILP for reachability properties: Constraint (9j) ensures criticality of the subsystem. Constraints (9k) and (9l) force the states of the sets M_i (i. e., target states) to be included in the subsystem and to have probability 1. Constraint (9m) assigns probability 0 to all states not in the subsystem and (9o) 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 (9o) is also satisfied for target states. This is the case due to the expression $\sum_{i=1}^{n} m^i_u$ 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.

Theorem 9 *The MILP formulation* (9a)–(9r) *yields an MCS for MDP $\mathcal{M}$ and ω-regular property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$.*

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

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 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

To demonstrate the feasibility of our approaches, we implemented the algorithms described in the previous sections in C++ in a tool named `LTLSubsys`. It supports the generation of MCS for DTMCs and MDPs with LTL properties [50]. 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.

All experiments were carried out on a Dual-Core AMD Opteron™ Processor 2220 running at 2.8 GHz clock frequency with 16 GB of main memory under Kubuntu 12.04 Linux running in 64 bit mode. We use IBM `Cplex` 12.4.0.1 [47] as the MILP solver, and Microsoft `Z3` 4.0 [42] as the SMT solver for linear real arithmetic. We started `Cplex` with a single thread. To generate DRAs from LTL properties, we use the tool `ltl2dstar` [51] which first calls `ltl2ba` [52] to generate nondeterministic Büchi automata and afterwards determinizes them.

As benchmark models we use the following randomized protocols and algorithms, which are all available from the `PRISM` benchmark repository [16] at `http://www.prismmodelchecker.org/casestudies`.

The following case studies are DTMCs:

- `sleader`-*N*-*K* is a *synchronous leader election protocol* [53]. 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: $\mathcal{P}_{\leq\lambda}(\Diamond\, \text{elected})$
  – Property 2: $\mathcal{P}_{\leq\lambda}(\text{start} \wedge \bigcirc\Diamond\, (\text{start} \wedge \bigcirc\Diamond\, (\text{start} \wedge \bigcirc\Diamond\, \text{elected})))$

- `crowds`-*N*-*R* is a model of the *crowds protocol* [54], 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: $\mathcal{P}_{\leq\lambda}(\Diamond\, \text{identified})$ (identified once)
  – Property 2: $\mathcal{P}_{\leq\lambda}(\Box\Diamond\, \text{identified})$ (infinitely often identified)

- `nand`-*N*-*K*: This case study is about constructing reliable computation from unreliable components [55, 56]. 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 a reliable state is never reached.

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

- `brp`-*N*-*K* is the *bounded retransmission protocol* [57, 58]. 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: $\mathcal{P}_{\leq\lambda}(\Diamond$ sender is unsure$)$

We additionally used the following MDP case studies:

- `aleader`-*N* is the *asynchronous leader election protocol* [53]. 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. If 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: $\mathcal{P}_{\leq\lambda}(\Diamond$ one node is elected as leader$)$

- `consensus`-*N*-*K* is the *randomized consensus shared coin protocol* [59] 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: $\mathcal{P}_{\leq\lambda}(\Diamond$ 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 [60]. The protocol aims at the minimization of data collision 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: $\mathcal{P}_{\leq\lambda}(\Diamond$ all processes have delivered their message$)$

We compare the results of our tool `LTLSubsys` with the tools `COMICS` [28] and `DiPro` [27] that apply (different) heuristics to obtain small critical subsystems. For the former we use its global search algorithm on the non-abstracted DTMC, the latter applies an extended stochastic breadth-first search (XBF). `COMICS` supports only reachability properties on DTMCs. `DiPro` also only supports reachability properties, but besides DTMCs it can handle MDPs. All experiments that could not be carried out due to tool limitations are marked with "—". Unless otherwise stated, a time limit of 3600 seconds and a memory limit of 16 GB were set.

Table 1 lists the benchmark results for a series of instances of the DTMC and MDP case studies described above. The first block of columns contains the name of the benchmark, its number of states and transitions, the property we used and the value of probability bound λ. The next two blocks contain the results of `COMICS` and `DiPro`. For each tool we give the size of the computed critical subsystem and the running time in seconds. The last block of columns contains the results of our tool `LTLSubsys`. Its first column $|S_{\min}|$ lists the number of states in the MCS, the second column the running time of the tool without any optimizations, the third column the best running time we could achieve with the optimizations. If the tool did not terminate within one hour, the entry is marked with "TO". For benchmarks that could not be solved even with optimizations, we give the best solution that could be found up to this point (entries marked with "$*$"). Additionally, we give the best computed lower bound for the MCS in brackets ("TO (*value*)"). The last column contains the memory consumption if it exceeded 1 GB.

Regarding the size of the computed subsystem we can observe that none of the heuristic tools was able to find an MCS, `COMICS` being often slightly better, but sometimes slower than `DiPro`. In some cases the differences to the MCS are considerable, cf. aleader-4, for which `DiPro` returned 1312 states, while the MCS contains at most 295 states. The gap is even larger for crowds12-6: When we interrupted `DiPro` after 1 hour, it had already collected 18 665 states for a subsystem that was not yet critical, while `LTLSubsys` has found a critical subsystem with 270 states within this time.

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

Model	$\|S\|$	$\|E\|$	φ	$\Pr(\mathcal{L})$	λ	COMICS [28] $\|S_{\text{heur}}\|$	COMICS [28] Time	DiPro [27] $\|S_{\text{heur}}\|$	DiPro [27] Time	LTLSubsys $\|S_{\min}\|$	LTLSubsys Time (no cuts)	LTLSubsys Time (best)	LTLSubsys Mem. (GB)
DTMCs:													
crowds-5-4	3 515	6 035	1	0.235	0.1	143	0.28	118	3.93	81	16.92	9.14	< 1
			2	0.235	0.1	—	—	—	—	335	16.01	9.72	< 1
crowds-5-6	18 817	32 677	1	0.427	0.1	143	18.62	118	3.62	83	419.51	81.44	< 1
			2	0.427	0.1	—	—	—	—	415	272.13	193.52	< 1
crowds-5-8	68 740	120 220	1	0.591	0.1	143	224.75	118	3.82	83	1 684.52	343.41	< 1
			2	0.591	0.1	—	—	—	—	*1 034**	TO (*835*)		< 1
crowds-12-6	829 669	2 166 277	1	0.332	0.1	TO		TO		*270**	TO (*235*)		15.8
			2	0.332	0.1	—	—	—	—	*2 523**	TO (*1 519*)		2.2
sleader-4-4	782	1 037	1	1	0.5	401	0.05	564	6.16	392	0.91	0.76	< 1
			2	0.02441	0.01	—	—	—	—	394	TO	2.05	< 1
sleader-4-6	3 902	5 197	1	1	0.5	1 957	0.37	3 542	19.85	1 953	14.39	7.58	< 1
			2	0.005487	0.001	—	—	—	—	949	TO	3.89	< 1
sleader-4-8	12 302	16 397	1	1	0.5	6 157	4.59	6 222	53.19	6 150	24.69	22.35	< 1
			2	0.001846	0.0005	—	—	—	—	3 718	TO	33.20	< 1
sleader-8-4	458 847	524 382	1	1	0.5	TO		TO		*229 411**	TO (*229 390*)		3.8
			2	0.057478	0.01	—	—	—	—	*458 847**	TO (*7 989*)		10.2
nand-5-2	1 728	2 505	1	0.389	0.2	—	—	—	—	394	22.46	17.04	< 1
nand-5-3	2 526	3 639	1	0.384	0.2	—	—	—	—	614	87.81	63.33	< 1
nand-5-4	3 324	4 773	1	0.386	0.2	—	—	—	—	854	407.10	242.74	< 1
nand-25-2	347 828	541 775	1	0.435	0.1	—	—	—	—	*9 075**	TO (*2 816*)		3.5
brp-32-2	1 349	1 731	1	$2.61 \cdot 10^{-5}$	10^{-5}	235	0.04	990	7.54	218	3.3	0.5	< 1
brp-512-2	21 509	27 651	1	$2.61 \cdot 10^{-5}$	10^{-5}	9 140	13.83	15 875	153.81	*9 023**	TO (*4 311*)		< 1
MDPs:													
consensus-2-2	272	400	1	1	0.1	—	—	100	1.44	15	TO	733.26	< 1
consensus-2-4	528	784	1	1	0.1	—	—	228	1.69	*34**	TO (*14*)		< 1
csma-2-2	1 038	1 054	1	1	0.1	—	—	214	2.04	195	TO	2 866.13	< 1
csma-2-4	7 958	7 988	1	1	0.1	—	—	792	3.68	410	TO	1 415.97	< 1
csma-2-6	66 718	66 788	1	1	0.1	—	—	627	3.66	*415**	TO (*392*)		< 1
aleader-3	364	573	1	1	0.5	—	—	254	1.95	*66**	TO (*21*)		< 1
aleader-4	3 172	6 252	1	1	0.5	—	—	1 312	7.24	*295**	TO (*10*)		< 1

Table 1: Benchmark results for DTMCs and MDPs. All times are measured in seconds, memory consumption in GB. The time limit was set to 3600 seconds, the memory limit to 16 GB.

without any guarantee 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, sleader-8-4 with a solution of 229 411 states and a lower bound of 229 390 states. In other cases, the gap is much larger, e. g., for aleader-4 with 295 compared to 10.

Our optimizations, presented for DTMCs in Section 3.3, have a great impact on the solving times. Especially the forward-cuts and backward-cuts improved the feasibility of our approaches for all case-studies. 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 sleader-4-4 benchmark, where without optimizations a timeout was reached while the MILP together with the complete reachability cuts was solved to optimality within 2.05 seconds.

We also compare the MILP formulation against the SMT formulation. However, Z3 runs into a timeout for all instances in Table 1. We applied Z3 then to the smaller instance crowds-3-3 with $\lambda = 0.1$. It consists of

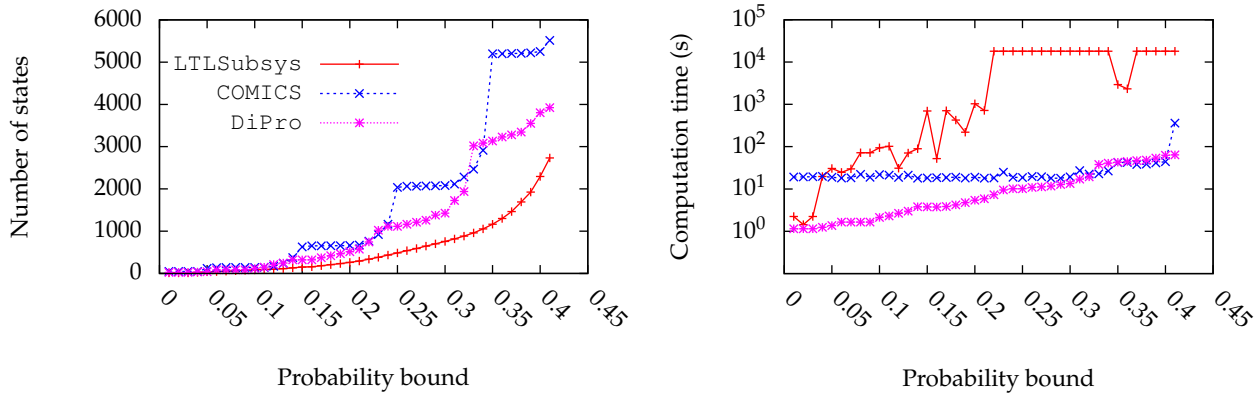


Figure 2: Size of the computed subsystem of crowds-5-6 and its computation time for different values of λ, comparing `COMICS` and `DiPro` with `LTLSubsys`. The time limit was set to 5 hours.

396 states and has an MCS with 39 states. Z3 needed for this instance 8 526.30 seconds, while `Cplex` solved it within 0.09 seconds.

In Figure 2 we study the evolution of the sizes of computed critical subsystems and the computation times. We computed a critical subsystem of crowds-5-6 for $\lambda \in [0, 0.41]$ 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 together on increasing λ. The computation time of `COMICS` stays more or less constant with the exception of the largest values of λ, while the time of `DiPro` increases linearly. `LTLSubsys` runs into a timeout for larger values of λ. In this case the best found solution is shown.

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 out 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] H. Hermanns, B. Wachter, L. Zhang, Probabilistic CEGAR, in: Proc. of CAV, Vol. 5123 of LNCS, Springer, 2008, pp. 162–175.

[4] R. Chadha, M. Viswanathan, A counterexample-guided abstraction-refinement framework for Markov decision processes, ACM Transactions on Computational Logic 12 (1) (2010) 1–45.

[5] P. Gastin, P. Moro, M. Zeitoun, Minimization of counterexamples in SPIN, in: Proc. of SPIN, Vol. 2989 of LNCS, Springer, 2004, pp. 92–108.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] M. Kwiatkowska, G. Norman, D. Parker, The PRISM benchmark suite, in: Proc. of QEST, IEEE CS Press, 2012, (to appear).

[17] T. Han, J.-P. Katoen, B. Damman, Counterexample generation in probabilistic model checking, IEEE Trans. on Software Engineering 35 (2) (2009) 241–257.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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, (to appear).

[29] 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.

[30] 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.

[31] 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.

[32] J. R. Norris, Markov Chains, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997.

[33] C. Baier, J.-P. Katoen, Principles of Model Checking, The MIT Press, 2008.

[34] M. Y. Vardi, Automatic verification of probabilistic concurrent finite-state programs, in: Proc. of FOCS, IEEE Computer Society, 1985, pp. 327–338.
[35] L. de Alfaro, Formal verification of probabilistic systems, Ph.D. thesis, Stanford University (1997).
[36] 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.
[37] 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.
[38] S. Safra, Complexity of automata on infinite objects, Ph.D. thesis, The Weizmann Institue of Science, Rehovot, Israel (1989).
[39] R. E. Tarjan, Depth-first search and linear graph algorithms, SIAM Journal on Computing 1 (2) (1972) 146–160.
[40] L. M. de Moura, N. Bjørner, Satisfiability modulo theories: introduction and applications, Communications of the ACM 54 (9) (2011) 69–77.
[41] 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.
[42] L. M. de Moura, N. Bjørner, Z3: An efficient SMT solver, in: Proc. of TACAS, Vol. 4963 of LNCS, Springer, 2008, pp. 337–340.
[43] C. Barrett, C. Tinelli, CVC3, in: Proc. of CAV, Vol. 4590 of LNCS, Springer, 2007, pp. 298–302.
[44] A. Griggio, A Practical Approach to Satisfiability Modulo Linear Integer Arithmetic, Journal on Satisfiability, Boolean Modeling and Computation 8 (2012) 1–27.
[45] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co Ltd, 1979.
[46] T. Achterberg, SCIP: Solving constraint integer programs, Mathematical Programming Computation 1 (1) (2009) 1–41.
[47] IBM CPLEX optimization studio, version 12.4, http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/ (2012).
[48] A. Schrijver, Theory of Linear and Integer Programming, Wiley, 1986.
[49] K. Chatterjee, M. Henzinger, Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification, in: Proc. of SODA, 2011, pp. 1318–1336.
[50] A. Pnueli, The temporal logic of programs, in: Proc. of FOCS, 1977, pp. 46–57. doi:10.1109/SFCS.1977.32.
[51] J. Klein, C. Baier, Experiments with deterministic ω-automata for formulas of linear temporal logic, Theoretical Computer Science 363 (2) (2006) 182–195.
[52] P. Gastin, D. Oddoux, Fast LTL to Büchi automata translation, in: Proc. of CAV, Vol. 2102 of LNCS, Springer, Paris, France, 2001, pp. 53–65.
[53] A. Itai, M. Rodeh, Symmetry breaking in distributed networks, Information and Computation 88 (1) (1990) 60–87.
[54] M. K. Reiter, A. D. Rubin, Crowds: Anonymity for web transactions, ACM Trans. on Information and System Security 1 (1) (1998) 66–92.
[55] J. von Neumann, Probabilistic logics and synthesis of reliable organisms from unreliable components, in: Automata Studies, Princeton University Press, 1956, pp. 43–98.
[56] 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.
[57] 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.
[58] 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.
[59] J. Aspnes, M. Herlihy, Fast randomized consensus using shared memory, Journal of Algorithms 15 (1) (1990) 441–460.
[60] 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. Each variable $v \in \text{Var}$ has a domain $\text{dom}(v)$. For real variables v, $\text{dom}(v) = [0,1] \subseteq \mathbb{R}$, for integer variables $\text{dom}(v) = \{0,1\} \subseteq \mathbb{Z}$. 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.

Lemma 2 *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC and $T \subseteq S$ a (possibly empty) set of target states. Assume that for all $s \in S$ holds: either a state in T is reachable from s or a state s^* with $\sum_{s' \in S} P(s^*, s') < 1$. Then the linear equation system*

$$p_s = \begin{cases} 1, & \text{if } s \in T, \\ \sum\limits_{s' \in S \setminus T} P(s,s') \cdot p_{s'} + \sum\limits_{s' \in T} P(s,s'), & \text{otherwise} \end{cases} \tag{A.1}$$

has a unique satisfying assignment.

PROOF. We follow the proof idea of [33, Theorem 10.19]. The solution of (A.1) is unique iff the solution of the corresponding homogeneous equation system is unique. Therefore consider

$$p_s = \begin{cases} 0, & \text{if } s \in T, \\ \sum\limits_{s' \in S \setminus T} P(s,s') \cdot p_{s'}, & \text{otherwise.} \end{cases} \tag{A.2}$$

The assignment $\nu(p_s) = 0$ for all $s \in S$ is a solution of (A.2).

Assume that there is a further satisfying assignment μ for (A.2) such that $\mu(p_s) \neq 0$ for some $s \in S \setminus T$. Since S is finite, the maximum of $|\mu(p_s)|$ exists. Let $\hat{\mu}$ be this maximum and $\hat{M} = \left\{ s \in S \,\middle|\, \hat{\mu} = |\mu(s)| \right\} \subseteq S \setminus T$. Since $\mu(p_s) \neq 0$ for some $s \in S \setminus T$, we have that $\hat{\mu} > 0$ and $\hat{M} \neq \emptyset$. Because of $P(s,s') \geq 0$ for all $s' \in S$ and $\sum_{s' \in S} P(s,s') \leq 1$ we have for $s \in \hat{M}$:

$$\hat{\mu} = |\mu(p_s)| \leq \sum_{s' \in \text{succ}_{\mathcal{D}'}(s) \setminus T} P(s,s') \cdot |\mu(p_{s'})| \leq \hat{\mu} \cdot \sum_{s' \in \text{succ}_{\mathcal{D}'}(s) \setminus T} P(s,s') \leq \hat{\mu} \,.$$

This implies that

$$\hat{\mu} = |\mu(p_s)| = \sum_{s' \in \text{succ}_{\mathcal{D}'}(s) \setminus T} P(s,s') \cdot |\mu(p_{s'})| = \hat{\mu} \cdot \sum_{s' \in S \setminus T} P(s,s') \,.$$

Since $\hat{\mu} > 0$ we have $\sum_{s' \in \text{succ}_{\mathcal{D}'}(s)} P(s,s') = 1$ and $\mu(p_{s'}) = \hat{\mu}$ for all $s' \in \text{succ}_{\mathcal{D}'}(s) \setminus T$. By induction it follows that $\sum_{s' \in \text{succ}_{\mathcal{D}'}(s'')} P(s'',s') = 1$ for all states s'' which are reachable from s within $\mathcal{D}'$. There is a path $s = s_0 s_1 \ldots s_n$ such that either $P(s_n, t) > 0$ for some $t \in T$ or $\sum_{s' \in \text{succ}(s_n)} P(s_n, s') < 1$. The latter case is a direct contradiction. In the former case, $P(s_n, t) > 0$ for some $t \in T$. Therefore $\sum_{s' \in \text{succ}_{\mathcal{D}}(s_n) \setminus T} < 1$, which is again a contradiction. Therefore our assumption that $\mu(p_s) \neq 0$ for some $s \in S$ was wrong. □

Now we can prove the correctness of the SMT-formulation for reachability properties of DTMCs.

$$\text{minimize} \quad \sum_{s \in S} x_s \tag{A.3a}$$

such that

$$\forall s \in T_a : \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \big(x_s = 1 \wedge p_s = 1\big) \tag{A.3b}$$

$$\forall s \in S \setminus T_a : \quad \big(x_s = 0 \wedge p_s = 0\big) \oplus \Big(x_s = 1 \wedge p_s = \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s,s') \cdot p_{s'}\Big) \tag{A.3c}$$

$$p_{s_{\text{init}}} > \lambda, \tag{A.3d}$$

Theorem 3 *The SMT formulation* (A.3a)–(A.3d) *yields an MCS for DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq \lambda}(\Diamond\, a)$.*

PROOF. Let $\mathcal{D}$ be a DTMC and $\mathcal{P}_{\leq\lambda}(\Diamond a)$ a reachability property that is violated by $\mathcal{D}$. We assume that $\mathcal{D}$ does not contain any irrelevant states for a and that $\mathcal{D}' = (S', s_{\text{init}}, P', L')$ is a critical subsystem of $\mathcal{D}$. We show that there is a satisfying assignment ν for the SMT formulation with $\nu(x_s) = 1$ iff $s \in S'$.

Since $\mathcal{D}'$ is a critical subsystem, model checking yields the following linear equation system with variables q_s for the probability to reach a T_a-state from s [33]:

$$q_s = \begin{cases} 1 & \text{if } s \in T_a, \\ 0 & \text{if } T_a \text{ is unreachable from } s \text{ within } \mathcal{D}', \text{ and} \\ \sum_{s' \in S'} P'(s,s') \cdot q_{s'}, & \text{otherwise.} \end{cases} \tag{A.4}$$

Let μ be a satisfying assignment of this equation system. Since $\mathcal{D}'$ is critical, we know that $\mu(q_{s_{\text{init}}}) > \lambda$. We define the following assignment ν:

$$\nu(x_s) = \begin{cases} 1, & \text{if } s \in S', \\ 0, & \text{otherwise} \end{cases} \qquad \text{and} \qquad \nu(p_s) = \begin{cases} \mu(q_s), & \text{if } s \in S', \\ 0, & \text{otherwise.} \end{cases}$$

We need to show that ν satisfies the SMT constraints. For constraints (A.3b) and (A.3d) this is obviously the case. So consider the remaining constraint (A.3c). If $s \notin S'$, according to the definition of ν, we have $\nu(x_s) = 0$ and $\nu(p_s) = 0$. This satisfies the first part of the constraint (and violates the second part). So the constraint is fulfilled because of the exclusive or between the two parts. If $s \in S'$, we have $\nu(x_s) = 1$ and distinguish two cases: First assume from s a state in T_a is reachable within $\mathcal{D}'$. Then:

$$\begin{aligned} \nu(p_s) = \mu(q_s) &= \sum_{s' \in S'} P'(s,s') \cdot \mu(q_{s'}) = \sum_{s' \in S'} P(s,s') \cdot \mu(q_{s'}) \\ &= \sum_{s' \in S' \cap \text{succ}_{\mathcal{D}}(s)} P(s,s') \cdot \mu(q_{s'}) = \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s,s') \cdot \nu(p_{s'}) \,. \end{aligned}$$

Therefore the SMT constraint is satisfied.

Otherwise, if from s the set T_a of target states is unreachable within $\mathcal{D}'$, we have $\nu(p_s) = \mu(q_s) = 0$. We need to show that in this case also the equation $p_s = \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s,s') \cdot p_{s'}$ of the SMT problem is fulfilled.

Let $S^0 = \{s \in S' \mid T_a \text{ is unreachable within } \mathcal{D}' \text{ from } s\}$. We can observe that $\text{succ}_{\mathcal{D}'}(s) \subseteq S^0$ for all $s \in S^0$. Now consider the linear equations of constraint (A.3c) that define the probabilities of the states $s \in S^0$. Since the probability of states outside of the subsystem is 0, we can restrict the sum to the states in $\mathcal{D}'$. We have the following linear equation system:

$$\forall s \in S^0 : \quad p_s = \sum_{s' \in \text{succ}_{\mathcal{D}'}(s)} P(s,s') \cdot p_{s'} \,. \tag{A.5}$$

This is a homogeneous equation system with $\nu(p_s) = 0$ as a solution. As $\mathcal{D}$ does not contain any irrelevant states, we can apply Lemma 2, which tells us that the solution of this equation system is unique. This then implies that the assignment ν that corresponds to the subsystem $\mathcal{D}'$ satisfies the SMT constraints.

We have herewith shown that for each critical subsystem there is an assignment that satisfies the SMT constraints. Now we show the opposite direction: Given a satisfying assignment for the SMT constraints, there is a corresponding critical subsystem.

Assume ν is a satisfying assignment for the SMT-constraints. We define the subsystem $\mathcal{D}' = (S', s_{\text{init}}, P', L')$ by $s \in S'$ iff $\nu(x_s) = 1$, $P'(s,s') = P(s,s')$ for $s, s' \in S'$, and $L'(s) = L(s)$ for $s \in S'$. We have to show that the subsystem is critical. Consider the linear equation system (A.4) from above. For states $S' \cap T_a$, the equation system is obviously satisfied. Assume $s \in S' \setminus T_a$ such that from s a T_a-state is reachable. Then

$$\nu(p_s) = \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s,s') \cdot \nu(p_{s'}) \qquad \text{due to the SMT constraints}$$

$$= \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s) \cap S'} P(s,s') \cdot \nu(p_{s'}) \qquad \text{since } \nu(p_{s''}) = 0 \text{ for all } s'' \notin S'$$

$$= \sum_{s' \in S'} P(s,s') \cdot \nu(p_{s'}) \qquad \text{since } P(s,s') = 0 \text{ for } s' \notin \mathrm{succ}_{\mathcal{D}}(s')$$

$$= \sum_{s' \in S'} P'(s,s') \cdot \nu(p_{s'}) \qquad \text{since } P'(s,s') = P(s,s') \text{ for } s,s' \in S'.$$

Hence, ν satisfies equation system (A.4).

The only remaining case is that there is no path from s to a T_a-state within S'. We have to show that the SMT constraints yield $\nu(p_s) = 0$. Since we assumed that $\mathcal{D}$ contains no irrelevant states, we can apply Lemma 2 to show that $\nu(s) = 0$, which satisfies the model checking equations. Hence the solution of (A.4) coincides with ν and we have that the subsystem is critical, since $\nu(p_{s_{\text{init}}}) > \lambda$.

As we have shown, there is a one-to-one correspondence between the satisfying assignments ν of the SMT constraints and the critical subsystems of $\mathcal{D}$. Since $\sum_{s \in S} \nu(x_s)$ is exactly the number of states in the subsystem, we obtain an MCS by minimizing this sum. □

Appendix B. MILP-Formulation for Reachability Properties of DTMCs

Lemma 3 *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC without irrelevant states, $S' \subseteq S$ with $s_{\text{init}} \in S'$, and $T \subseteq S$ a set of target states. If*

$$p_s \leq \sum_{s \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \setminus T} P(s,s') \cdot p_{s'} + \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \cap T} P(s,s') \tag{B.1}$$

has a satisfying assignment μ with $\mu(p_{s_{\text{init}}}) > \lambda$, then the satisfying assignment ν of

$$p_s = \sum_{s \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \setminus T} P(s,s') \cdot p_{s'} + \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \cap T} P(s,s') \tag{B.2}$$

also satisfies $\nu(p_{s_{\text{init}}}) > \lambda$.

PROOF. First, the solution of (B.2) is unique according to Lemma 2.

Let μ_{opt} be an assignment for (B.1) such that $\mu_{\text{opt}}(p_{s_{\text{init}}}) = \max\{\mu(p_{s_{\text{init}}}) \mid \mu \text{ satisfies (B.1)}\}$ is maximal among all satisfying assignments of (B.1). Since the domains of all variables are bounded, this maximum exists and is finite.

We claim that for all states that are reachable from s_{init} the inequalities (B.1) are satisfied by μ_{opt} with equality. Assume the converse is true, i. e., there is a state $s \in S' \setminus T$ that is reachable from s_{init} such that

$$0 < \varepsilon := \left(\sum_{s \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \setminus T} P(s,s') \cdot \mu_{\text{opt}}(p_{s'}) + \sum_{s' \in \mathrm{succ}_{\mathcal{D}}(s) \cap S' \cap T} P(s,s') \right) - \mu_{\text{opt}}(p_s) \, .$$

Let $s_{\text{init}} = s_0 s_1 \ldots s_n = s$ be a path vom s_{init} to s. We can increase the value $\mu_{\text{opt}}(p_{s_n})$ by at least $\varepsilon_n = \varepsilon$ (more, if p_{s_n} also appears on the right-hand side; note that $0 \leq P(s_n, s_n) < 1$ holds). This does not violate any inequality, since in the inequalities for the other states p_{s_n} appears with a non-negative coefficient. Assume, for some $i \leq n$, we have increased the value of s_i by ε_i. Then the right-hand side of the inequality for s_{i-1} increases by at least $P(s_{i-1}, s_i) \cdot \varepsilon_i > 0$. Therefore we can also increase the value of $p_{s_{i-1}}$ by $P(s_{i-1}, s_i) \cdot \varepsilon_i$. This can be continued along the path back to $s_{\text{init}} = s_0$, whose value may be increased by $\varepsilon_0 = \varepsilon \cdot \prod_{i=0}^{n-1} P(s_i, s_{i+1}) > 0$. Therefore $\mu_{\text{opt}}(p_{s_{\text{init}}})$ was not optimal, contradicting our assumption. This means, the inequalities of all states that are reachable from s_{init} are satisfied with equality for the optimal solution. We do not have to take the values of unreachable states into account, because they do not influence the value of s_{init}. □

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

such that

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

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

$$\forall s \in S \setminus T_a : \quad p_s \leq \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s, s') \cdot p_{s'} \tag{B.3d}$$

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

Theorem 4 *The MILP formulation* (B.3a)–(B.3e) *yields an MCS for $\mathcal{D}$ and the property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

PROOF. We show that each satisfying assignment of the SMT constraints (A.3b)–(A.3d) has a corresponding satisfying assignment of the MILP constraints and vice versa.

Let ν be a satisfying assignment of the SMT constraints. For target states $s \in S$ we have either $\nu(x_s) = 0$ and $\nu(p_s) = 0$ or $\nu(x_s) = 1$ and $\nu(p_s) = 1$, i. e., $\nu(x_s) \in \{0, 1\}$ and $\nu(p_s) = \nu(x_s)$. Consequently constraint (B.3b) is fulfilled.

If $s \in S \setminus T_a$, the SMT constraints require that either $\nu(x_s) = 0$ and $\nu(p_s) = 0$ or $\nu(x_s) = 1$ and $\nu(p_s) = \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s, s') \cdot \nu(p_{s'})$. In the former case $0 = \nu(p_s) \leq \nu(x_s) = 0$ holds and in the latter case $[0, 1] \ni \nu(p_s) \leq \nu(x_s) = 1$. So ν satisfies constraint (B.3c).

Now consider constraint (B.3d). For states s in the subsystem, i. e., $\nu(x_s) = 1$, this constraint is satisfied with equality by ν. For states not in the subsystem, i. e., $\nu(x_s) = 0$, we have $\nu(p_s) = 0$. Since $P(s, s') \geq 0$ for all $s, s' \in S$ and $\nu(p_{s'}) \geq 0$ for all $s' \in S$, the right-hand side of the inequality is non-negative. Therefore this constraint holds for all states.

Finally constraint (B.3e) is the same as constraint (A.3d) and holds therefore, too. This means, each assignment that satisfies the SMT constraints also satisfies the MILP.

Now assume that ν is a satisfying assignment of the MILP constraints. We show that there is a satisfying assignment ν' for the SMT-constraints such that $\nu'(x_s) = \nu(x_s)$ for all $s \in S$. We set $S' = \{s \in S \mid \nu(x_s) = 1\}$. Let $\nu'(p_s) = 1$ for all $s \in T_a \cap S'$ and $\nu'(p_s) = 0$ for $s \notin S'$. We have to determine $\nu'(p_s)$ for $s \in S' \setminus T_a$ such that the constraint (A.3c) is satisfied for all $s \in S \setminus T_a$. This is trivially the case for states $s \in S \setminus S'$. For $s \in S' \setminus T_a$ we have to satisfy the linear equation system:

$$p_s = \sum_{s \in \text{succ}_{\mathcal{D}}(s) \cap S' \setminus T_a} P(s, s') \cdot p_{s'} + \sum_{s' \in \text{succ}_{\mathcal{D}}(s) \cap S' \cap T_a} P(s, s'). \tag{B.4}$$

According to Lemma 2, it has a unique satisfying assignment by which we extend our partial assignment ν'. We have to show that the such defined assignment ν' corresponds to a critical subsystem, i. e., that $\nu'(p_{s_{\text{init}}}) > \lambda$.

Now return to the MILP formulation. For states $s \in S' \setminus T_a$, the value $\nu(p_s)$ is restricted by the inequality

$$p_s \leq \sum_{s' \in \text{succ}_{\mathcal{D}}(s)} P(s, s') \cdot p_{s'}.$$

Because of $\nu(p_s) = 1$ for $s \in T_a \cap S'$ and $\nu(p_s) = 0$ for $s \notin S'$, this simplifies to

$$p_s \leq \sum_{s \in \text{succ}_{\mathcal{D}}(s) \cap S' \setminus T_a} P(s, s') \cdot p_{s'} + \sum_{s' \in \text{succ}_{\mathcal{D}}(s) \cap S' \cap T_a} P(s, s'). \tag{B.5}$$

Applying Lemma 3 tells us that $\nu'(p_{s_{\text{init}}}) > \lambda$. This means the constructed assignment ν' represents a critical subsystem of $\mathcal{D}$. □

Appendix C. Correctness of Optimizations

Appendix C.1. Forward/Backward Cuts

$$\forall s \in S \setminus T_a: \quad -x_s + \sum_{s' \in \mathrm{succ}(s) \setminus \{s\}} x_{s'} \geq 0 \tag{C.1a}$$

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

Lemma 4 *The forward and backward cuts are satisfied for each MCS of DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

PROOF. Let $\mathcal{D}' = (S', s_{\mathrm{init}}, P', L')$ be an MCS of $\mathcal{D} = (S, s_{\mathrm{init}}, P, L)$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. Assume (C.1a) is violated for state $s \in S'$, i. e., for the corresending assigment ν, we have $\nu(x_s) = 1$, 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 $\nu(x_{s'}) = 0$. The only solution is $\nu(p_s) = 0$. Therefore state s is irrelevant and can be removed from the MCS without altering the probability of the initial state. This contradicts the minimality of $\mathcal{D}'$.

Assume now that (C.1b) is violated for state $s \in S'$, i. e., for the corresponding assignment ν we have $\nu(x_s) = 1$, but $\nu(x_{s'}) = 0$ for all $s' \in \mathrm{pred}_{\mathcal{D}}(s) \setminus \{s\}$. Then in the equation system which determines the probability of s_{init}, p_s does not appear. Therefore removing s does not change the probability of the initial state and the criticality, which is a contraction to $\mathcal{D}'$ being an MCS. □

Appendix C.2. SCC Cuts

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

$$\forall \mathrm{SCC}\ C, C \cap T_a = \emptyset: \quad \sum_{s \in C} x_s \;\leq\; \big|C\big| \cdot \sum_{s \in \mathrm{Out}(C)} x_s\,. \tag{C.2b}$$

Lemma 5 *The input and output SCC cuts are satisfied for each MCS of DTMC $\mathcal{D}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$.*

PROOF. Let $\mathcal{D}' = (S', s_{\mathrm{init}}, P', L')$ be an MCS of $\mathcal{D} = (S, s_{\mathrm{init}}, P, L)$ and property $\mathcal{P}_{\leq\lambda}(\Diamond\, a)$. Let $C \subseteq S \setminus \{s_{\mathrm{init}}\}$ be an SCC which violates (C.2a). All paths in $\mathcal{D}$ from s_{init} to T_a passing through C contain a state in $\mathrm{In}(C)$. Since $S' \cap \mathrm{In}(C) = \emptyset$, there is no path in $\mathcal{D}'$ from s_{init} to T_a containing a state from C. Therefore all states in $C \cap S' \neq \emptyset$ are irrelevant and can be removed from $\mathcal{D}'$ without alternating the probability of s_{init}, which contradicts the minimality of $\mathcal{D}'$.

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. □

Appendix C.3. Forward Reachability Constraints

Let S be a finite set and $I \subseteq S$ such that $\mathrm{pred}(s) \neq \emptyset$ for all $s \in S \setminus I$. 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 $s, s' \in S$:

$$\forall s' \in S \setminus I\ \forall s \in \mathrm{pred}(s): \quad 2t^{\leftarrow}_{s,s'} \leq x_s + x_{s'} \tag{C.3}$$

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

$$\forall s' \in S \setminus I: \quad (1 - x_{s'}) + \sum_{s \in \mathrm{pred}(s')} t^{\leftarrow}_{s,s'} \geq 1\,. \tag{C.5}$$

Lemma 6 *Let ν be a satisfying assignment of* (C.3)–(C.5). *Then $\nu(x_{s'}) = 1$ implies that there is a path $s_0 s_1 \dots s_n = s'$ from a state $s_0 \in I$ to s' with $\nu(x_{s_i}) = 1$ for all $0 \leq i \leq n$.*

PROOF. Constraint (C.5) enforces that each state $s' \in S \setminus I$ with $\nu(x_{s'}) = 1$ has a predecessor state $s \in \mathrm{pred}(s')$ with $t^{\rightarrow}_{s,s'} = 1$. Constraint (C.3) ensures that for this predecessor state $\nu(x_s) = 1$ holds. Constraint (C.4) finally ensures that $\nu(r^{\rightarrow}_s) < \nu(r^{\rightarrow}_{s'})$.

Assume there is a state $u_0 \in S \setminus I$ 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 $u_{i+1} \in \mathrm{pred}(u_i)$, $\nu(u_i) = 1$, $\nu(t^{\rightarrow}_{u_{i+1},u_i}) = 1$, 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 $\nu(r^{\leftarrow}_{u_k}) < \nu(r^{\leftarrow}_{u_i})$. Contradiction. Therefore our assumption was wrong and the lemma is valid. □

Lemma 7 *Let $S' \subseteq S$ such that each state $s \in S'$ is reachable from a state $t \in I$. Then there is a satisfying assignment ν of* (C.3)–(C.5) *with $\nu(x_s) = 1$ iff $s \in S'$.*

PROOF. Reverse all edges between states in S'. The shortest paths from the states in S' to I in the reversed graph form a forrest rooted at the I-states. Let n be the length of the longest such path. If a node s has distance k from the I-states, assign $\nu(r^{\rightarrow}_s) := k/n$. Assign $\nu(t^{\rightarrow}_{s,s'}) := 1$ iff (s', s) is contained in the shortest path forrest. One can easily check that the three constraints are satisfied by ν. □

Appendix C.4. Backward Reachability Constraints

Let S be a finite set and $T \subseteq S$ such that $\mathrm{succ}(s) \neq \emptyset$ for all $s \in S \setminus T$. Consider the following 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 $s, s' \in S$:

$$\forall s \in S \setminus T\ \forall s' \in \mathrm{succ}(s): \quad 2t^{\leftarrow}_{s,s'} \leq x_s + x'_s \tag{C.6}$$

$$\forall s \in S \setminus T\ \forall s' \in \mathrm{succ}(s): \quad r^{\leftarrow}_s < r^{\leftarrow}_{s'} + (1 - t^{\leftarrow}_{s,s'}) \tag{C.7}$$

$$\forall s \in S \setminus T: \quad (1 - x_s) + \sum_{s' \in \mathrm{succ}(s)} t^{\leftarrow}_{s,s'} \geq 1\,. \tag{C.8}$$

Lemma 8 *Let ν be a satisfying assignment of* (C.6)–(C.8). *Then $\nu(x_s) = 1$ 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.8) 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.6) ensures that for this successor state $\nu(x_{s'}) = 1$ holds. Constraint (C.7) finally ensures that $\nu(r^{\leftarrow}_s) < \nu(r^{\leftarrow}_{s'})$.

Assume 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 $u_{i+1} \in \mathrm{succ}(u_i)$, $\nu(u_i) = 1$, $\nu(t^{\leftarrow}_{u_i,u_{i+1}}) = 1$, 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 $\nu(r^{\leftarrow}_{u_i}) < \nu(r^{\leftarrow}_{u_k})$. Contradiction. Therefore our assumption was wrong and the lemma is valid. □

Lemma 9 *Let $S' \subseteq S$ such that from each state $s \in S'$ a state $t \in T$ is reachable. Then there is a satisfying assignment ν of* (C.6)–(C.8) *with $\nu(x_s) = 1$ iff $s \in S'$.*

PROOF. The shortest paths from the states in S' to T form a forrest rooted at the T-states. Let n be the length of the longest such path. If a node s has distance k from the T-states, assign $\nu(r^{\rightarrow}_s) := k/n$. Assign $\nu(t^{\rightarrow}_{s,s'}) := 1$ iff (s', s) is contained in the shortest path forrest. One can easily check that the three constraints are satisfied by ν. □

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

Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ an ω-regular property which is violated by $\mathcal{D}$. $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ is a DRA such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. Let $\{T_1, \ldots, T_n\}$ be the set of accepting BSCCs of $\mathcal{D} \otimes \mathcal{A}$ and $T = \bigcup_{i=1}^{n} T_i$. As always we assume that the product automaton $\mathcal{D} \otimes \mathcal{A} = (S \times Q, (s,q)_{\text{init}}, P', L')$ does not contain any irrelevant states.

The SMT formulation for MCSs of ω-regular properties is as follows:

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

such that

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

$$\forall i = 1, \ldots, n \; \forall (s,q) \in T_i : (x_{T_i} = 0 \wedge p_{(s,q)} = 0) \oplus (x_{T_i} = 1 \wedge p_{(s,q)} = 1 \wedge x_s = 1) \tag{D.1c}$$

$$\forall (s,q) \in (S \times Q) \setminus T : \big(x_s = 0 \wedge p_{(s,q)} = 0\big) \oplus \big(x_s = 1 \wedge p_{(s,q)} = \sum_{(s',q') \in \text{succ}_{\mathcal{D}\otimes\mathcal{A}}(s)} P'\big((s,q),(s',q')\big) \cdot p_{(s',q')}\big). \tag{D.1d}$$

Lemma 10 *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC, $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ a violated ω-regular property and $S' \subseteq S$ a critical subsystem of $\mathcal{D}$. Then there is a satisfying assignment ν of the SMT constraints* (D.1b)–(D.1d) *such that $\nu(x_s) = 1$ iff $s \in S'$.*

PROOF. Let $S' \subseteq S$ be a critical subsystem of $\mathcal{D}$ and $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ a DRA with $F = \{(R_i, A_i) \mid i = 1, \ldots, n\}$ and $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. $\Pi = \{\pi \in \text{Paths}^{\text{inf}}_{\mathcal{D}}(s_{\text{init}}) \mid \pi \vDash \mathcal{L} \wedge \forall i \geq 0 : \pi^i \in S'\}$ denotes the set of infinite paths within the subsystem that satisfy $\mathcal{L}$. Since 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 $\Pr\{\pi \in \text{Paths}^{\text{inf}}_{\mathcal{D}\otimes\mathcal{A}}((s,q)_{\text{init}}) \mid \inf(\pi) \in \text{BSCC}(\mathcal{D} \otimes \mathcal{A})\} = 1$ [33, 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
- $\forall c \in C \; \exists i \in \{1, \ldots, n\} : (\forall (s,q) \in C : R_i \notin L'(s,q)) \wedge (\exists (s,q) \in c : A_i \in L'(s,q))$, 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 SMT constraints above:

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

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

This lemma also implies that the subsystems are independent of the actual DRA used for the property.

Lemma 11 *Let $\mathcal{D} = (S, s_{\text{init}}, P, L)$ be a DTMC, $\mathcal{L}$ an ω-regular property, and $\mathcal{A}$ a DRA with $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. For each satisfying assignment ν of the SMT-constraints there is a critical subsystem of $\mathcal{D}$ with state space $S' = \{s \in S \mid \nu(x_s) = 1\}$.*

PROOF. Let ν be a satisfying assignment of the SMT constraints (D.1b)–(D.1d). We define a subsystem $\mathcal{D}' = (S', s_{\text{init}}, P', L')$ by $S' = \{s \in S \mid \nu(x_s) = 1\}$, $P'(s,s') = P(s,s')$ for all $s, s' \in S'$, and $L'(s) = L(s)$ for $s \in S'$. We have to show that $\mathcal{D}'$ is critical.

We construct the product automaton $\mathcal{D}' \otimes \mathcal{A}$. It consists of all states $(s,q) \in S \times Q$ with $\nu(x_s) = 1$. Let $B = \{T_i \mid \nu(x_{T_i}) = 1\}$. It holds $\bigcup B \subseteq S' \times Q$ since $\nu(x_{T_i}) = 1$ implies $\nu(x_s) = 1$ for all $s \in T_i$. Therefore the set of accepting BSCCs of $\mathcal{D}' \otimes \mathcal{A}$ is a superset of B. Therefore the probability to reach an accepting BSCC is at least the probability to reach a state in $\bigcup B$. Now consider the linear equation system which determines the reachability probabilities for $(s,q) \in S \times Q \setminus \bigcup B$:

$$p_{(s,q)} = \begin{cases} 0, & \text{if } \bigcup B \text{ is not reachable from } s, \\ \sum\limits_{(s',q') \in \text{succ}_{\mathcal{D}'\otimes\mathcal{A}}(s,q)} P((s,q),(s',q')) \cdot p_{(s',q')}, & \text{otherwise.} \end{cases}$$

In the SMT-formulation we have the equation $p_s = \sum\limits_{(s',q') \in \text{succ}_{\mathcal{D}'\otimes\mathcal{A}}(s,q)} P((s,q),(s',q')) \cdot p_{(s',q')}$ for all states $s \in S \times Q \setminus T$. Since $\mathcal{D} \otimes \mathcal{A}$ does not contain any irrelevant states, we can use Lemma 2 to show both formulations are equivalent. Then $\Pr^{(s,q)_{\text{init}}}_{\mathcal{D}'\times\mathcal{A}}(\Diamond\, \mathit{accept}) \geq \Pr^{(s,q)_{\text{init}}}_{\mathcal{D}'\times\mathcal{A}}(\Diamond \bigcup B) = \nu(p_{(s,q)_{\text{init}}}) > \lambda$. Hence the constructed subsystem is critical. □

Now we can use both lemmas to prove the following theorem:

Theorem 10 *The SMT formulation* (D.1a)–(D.1d) *yield an MCS for DTMC $\mathcal{D}$ and ω-regular property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$.*

PROOF. According to Lemma 10, there is a satisfying assignment for each critical subsystem. Lemma 11 states that also the converse holds. Consequently there is a one-to-one mapping between the critical subsystems of $\mathcal{D}$ and the satisfying assignments of the SMT constraints. Since $\sum_{s\in S} \nu(x_s)$ is equal to the number of states in the subsystem, minimizing this sum yields an MCS. □

Appendix E. 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 $\mathcal{L}(\mathcal{A}) = \mathcal{L}$. The MILP-formulation for an MCS for $\mathcal{D}$ and $\mathcal{L}$ is as follows:

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

such that

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

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

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

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

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

Theorem 6 *The MILP formulation* (E.1a)–(E.1f) *yields an MCS for DTMC $\mathcal{D}$ and ω-regular property $\mathcal{P}_{\leq\lambda}(\mathcal{L})$.*

PROOF. We show the theorem by proving the equivalence of the MILP and the SMT formulation given in Appendix D.

SMT $\rightarrow$ MILP: It is easy to see that each satisfying assignment of the SMT constraints also satisfy the MILP constraints.

MILP $\rightarrow$ SMT: Let ν be a satisfying assignment of the MILP constraints. We construct a satisfying assignment μ of the SMT constraints:

- $\mu(x_s) = \nu(x_s$ for all $s \in S$,
- $\mu(x_{T_i}) = \nu(x_{T_i})$ for $i = 1,\ldots,n$,
- $\mu(p_{(s,q)}) = 0$ if $\nu(x_s) = 0$, and
- $\mu(p_{(s,q)}) = 1$ if $\nu(x_s) = 1$ and $(s,q) \in T$.

For the remaining states $(s,q) \in S \times Q \setminus T$ we have to satisfy the following equation system:

$$p_{(s,q)} = \sum_{(s',q')\in \mathrm{succ}_{\mathcal{D}}(s,q)} P((s,q),(s',q'))\cdot p_{(s',q')}.$$

According to Lemma 2 its solution is unique. Let $\mu(p_{(s,q)})$ be this solution. We have to show that $\mu(p_{(s,q)_{\text{init}}}) > \lambda$. This, however, directly follows from Lemma 3. □

Appendix F. Complexity of MCSs for MDPs

Theorem 7 ([4]) *Let MDP $\mathcal{M}$ with $\mathcal{M} \not\models \mathcal{P}_{\leq\lambda}(\Diamond\, a)$ and $k \in \mathbb{N}$. Then: the problem to decide if 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 [4].) 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 [45, 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 $\alpha' \in Act$, and finally

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:

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

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

"$\Rightarrow$" Let $\mathcal{M}'$ 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 $c \in \cap C$ 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 G. MILP-Formulation for Reachability Properties of MDPs

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

such that

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

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

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

$$\forall s \in S \setminus T_a: \quad (1 - x_s) + \sum_{\alpha \in Act} \sigma_{s,\alpha} = 1 \tag{G.1e}$$

$$\forall s \in S \setminus T_a \; \forall \alpha \in Act: \quad p_s \leq (1 - \sigma_{s,\alpha}) + \sum_{s' \in \text{succ}_{\mathcal{M}}(s,\alpha)} P(s, \alpha, s') \cdot p_{s'} \tag{G.1f}$$

$$\forall (s, \alpha) \in Act_{\mathcal{M}}^{\text{probl}(a)} \; \forall s' \in \text{succ}_{\mathcal{M}}(s, \alpha): \quad 2t^{\leftarrow}_{s,s'} \leq x_s + x_{s'} \tag{G.1g}$$

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

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

Lemma 12 *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP and $\mathcal{P}_{\leq\lambda}(\Diamond a)$ a violated reachability property. Let ν be a satisfying assignment of the MILP constraints* (G.1b)–(G.1i)*. Then the subsystem $\mathcal{M}' = (S', s_{\text{init}}, P', L')$ with $S' = \{s \in S \mid \nu(x_s) = 1\}$, $P(s, s') = P(s, \sigma(s), s')$ and $L'(s) = L(s)$ for all $s, s' \in S$ with $\sigma(s) = \alpha$ iff $\nu(\sigma_{s,\alpha}) = 1$ is critical.*

PROOF. The scheduler σ is well-defined, since constraint (G.1e) ensures that for each $s \in S'$ there is one and only one action $\alpha \in Act$ with $\sigma_{s,\alpha} = 1$. We can observe that constraints (G.1f) and (G.1i) are satisfied for all $\alpha \neq \sigma(s)$.

Now consider the DTMC $\mathcal{M}'$ induced by σ and the following linear equation system:

$$p_s = \begin{cases} 1 & \text{if } s \in T_a, \\ \sum_{s' \in \text{succ}_{\mathcal{M}'}(s)} P'(s, s') \cdot p_{s'} & \text{otherwise.} \end{cases}$$

For $s \in S' \setminus T_a$, we obtain

$$p_s = \sum_{s' \in \text{succ}_{\mathcal{M}^\sigma}(s) \setminus T_a} P'(s, s') \cdot p_{s'} + \sum_{s' \in \text{succ}_{\mathcal{M}^\sigma}(s) \cap T_a} P'(s, s') \,. \tag{G.2}$$

From all unproblematic states $s \in S' \setminus S^{\text{probl}(a)}_{\mathcal{M}}$ there is a path in $\mathcal{M}^{\sigma'}$ to a target state for all schedulers σ'. Due to the backward reachability constraints (G.1g)–(G.1i), from all problematic states an unproblematic state and therefore in $\mathcal{M}^\sigma$ also a target state is reachable. Since $\mathcal{M}$ does not contain irrelevant states, we can apply Lemma 2. It states that this equation system has a unique solution, which is the reachability probabilities for T_a.

Now consider again the MILP. Due to the constraint $p_s = x_s$ for T_a-states, ν satisfies the equation for target states. Consider the implications which are triggered by the values of x_s and $\sigma_{s,\alpha}$ for $s \in S' \setminus T_a$:

$$p_s \leq \sum_{s' \in \text{succ}_{\mathcal{M}}(s)} P(s, \alpha, s') \cdot p_{s'}$$

which simplifies to

$$p_s \leq \sum_{s' \in \text{succ}_{\mathcal{M}^\sigma}(s) \setminus T_a} P'(s, s') \cdot p_{s'} + \sum_{s' \in \text{succ}_{\mathcal{M}^\sigma}(s) \cap T_a} P'(s, s') \,. \tag{G.3}$$

We now apply Lemma 3, which tells us that the solution of (G.2) is not smaller than any solution of (G.3), which is larger than λ by assumption. Therefore the subsystem is critical. □

Lemma 13 *Let $S' \subseteq S$ be a critical subsystem of the MDP $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ under scheduler σ. We require that from any state $s \in S'$ a target state in T_a is reachable within the subsystem. Then there exists a satisfying assignment ν of the MILP constraints* (G.1a)–(G.1i) *with $\nu(x_s) = 1$ iff $s \in S'$ and $\sigma_{s,\alpha} = 1$ iff $\sigma(s) = \alpha$.*

PROOF. Let $\mathcal{M}' = (S', s_{\text{init}}, P', L')$ be such a subsystem and σ the corresponding scheduler. We construct an assignment ν as follows: $\nu(x_s) = 1$ iff $s \in S'$, $\nu(p_s) = 0$ if $s \notin S$, $\sigma_{s,\alpha} = 1$ iff $\alpha = \sigma(s)$ and $s \in S'$. Since from each state $s \in S'$ a target state can be reached, Lemma 9 ensures that the reachability constraints have a satisfying assignment. Let $\nu(r_s^{\rightarrow})$ and $\nu(t_{s,s'}^{\rightarrow})$ be the values of such an assignment. It remains to satisfy $p_s \leq \sum_{s' \in \text{succ}_{\mathcal{M}'}(s)} P(s, \sigma(s), s') \cdot p_{s'}$. This constraint is fulfilled with equality by the probabilities of reaching a T-state within $\mathcal{M}'$. If we let $\nu(p_s)$ be these reachability probabilities, all constraints are satisfied. □

Theorem 8 *The MILP formulation* (G.1a)–(G.1i) *yields an MCS for MDP $\mathcal{M}$ and property $\mathcal{P}_{\leq\lambda}(\Diamond a)$.*

PROOF. Every solution of the MILP corresponds to a critical subsystem. Every subsystem in which from each state a target state is reachable is a solution of the MILP. This is the case for every MCS. Since $\sum_{s \in S} \nu(x_s)$ is the number of states in the subsystem corresponding to the satisfying assignment ν, we obtain an MCS by minimizing this sum. □

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

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. $\forall u \in M_i : \sum_{u' \in \mathrm{succ}(u,\sigma(u)) \cap M_i} P'(u, \sigma(u), u') = 1$,
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 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. □

Lemma 14 *Let $\mathcal{M} = (S, s_{\text{init}}, Act, P, L)$ be an MDP, $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ a violated ω-regular property and (S', A) with $S' \subseteq S$ and $A : S' \to 2^{Act}$ a critical subsystem of $\mathcal{M}$. Then there is a satisfying assignment ν of the MILP* (9a)–(9r) *such that $\nu(x_s) = 1$ iff $s \in S'$.*

PROOF. Let $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, F)$ be a DRA with $F = \{(R_i, A_i) \mid i = 1, \ldots, m\}$ such that $\mathcal{L} = \mathcal{L}(\mathcal{A})$. Consider the MDP $\mathcal{M}' = (S', s_{\text{init}}, Act, P', L')$ with $P'(s, s') = P(s, s')$ and $L'(s) = L(s)$ for $s, s' \in S'$ and the product automaton $\mathcal{M}' \otimes \mathcal{A}$. Since the subsystem is critical, there is a memoryless deterministic scheduler σ such that (1) $\sigma(s, q) \in A(s)$ for all $(s, q) \in S' \times Q$ and (2) $\Pr^{(s,q)_{\text{init}}}_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, accept) > \lambda$. Let B be the set of accepting BSCCs of $(\mathcal{M} \otimes \mathcal{A})^\sigma$ and $M_i = \bigcup\{C \in B \mid C \cap R_i = \emptyset \wedge C \cap A_i \neq \emptyset\}$.

We define the following (partial) variable assignment:

- $\nu(x_s) = 1$ iff $s \in S'$
- $\nu(\sigma_{s,\alpha}) = 1$ iff $s \in S'$, $\sigma(s, q) = \alpha$ and an accepting BSCC is reachable from (s, q), and
- $\nu(p_{(s,q)}) = \Pr^{(s,q)}_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, accept)$.

Now we check all constraint for satisfaction:

(9b) This constraint is satisfied since we do not select any action for states (s, q) with $s \notin S'$ and σ selects exactly one action for each state (s, q) with $s \in S'$.

(9c) $\sum_{\alpha \in Act} \sigma_{(s,q),\alpha} = 0$ for $s \in S'$ iff no accepting BSCC is reachable from (s, q). Then $\Pr^{(s,q)}_{(\mathcal{M}' \otimes \mathcal{A})^\sigma}(\Diamond\, accept) = 0$ holds and the constraint is satisfied.

(9d) 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.

(9e) For states outside M_i and for actions not chosen by σ, the constraint is satisfied because in these cases $(2 - m^i_{(s,q)} - \sigma_{(s,q),\alpha}) \geq 1$. For states (s, q) with $s \in S'$ and action $\alpha = \sigma(s, q)$, $\nu(m^i_{(s',q')}) = 1$ is required for all successor states (s', q') of (s, q). This is the case since M_i is a union of BSCCs.

(9f) In the Definition of M_i we have required that $M_i \cap R_i = \emptyset$. Therefore this constraint is fulfilled.

(9g)–(9i) 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 9 to obtain a satisfying assignment for these backward reachability constraints.

(9j) $\nu(p_{(s,q)_{\text{init}}}) = \Pr^{(s,q)_{\text{init}}}_{(\mathcal{M}'\otimes\mathcal{A})^\sigma}(\Diamond\, accept) > \lambda$ holds since the subsystem is critical.

(9k) Since M_i contains only states in the subsystem, this inequality is satisfied.

(9l) For target states, which are the states in the accepting BSCCs, the reachability probability is one.

(9m) States not in the subsystem have probability zero. For states inside the subsystem, the inequality is trivially satisfied.

(9o) For states in an accepting BSCC this constraint is fulfilled trivially, since the right-hand side evaluates at least to one. The reachability probabilities for the remaining states which can reach the accepting BSCCs satisfy the equality

$$p_{(s,q)} = \sum_{(s',q')\in \text{succ}_{\mathcal{M}'\otimes\mathcal{A}}(s,q)} P'((s,q),(s',q')) \cdot p_{(s,q)}$$

and therefore also this constraint. For the remaining states $\nu(p_{(s,q)}) = 0$ holds, also satisfying the constraint.

(9p)–(9r) : These are backward reachability constraints with the accepting BSCCs as the target states. We distinguish different cases:

- $s \notin S'$: Set $\nu(t^M_{(s,q),(s',q')}) = 0$ for all $q \in Q$ and all $(s',q') \in \text{succ}_{\mathcal{M}'\otimes\mathcal{A}}(s,q)$ and $\nu(r^M_{(s,q)}) = 0$. Then all three constraints are fulfilled.
- $s \in S'$, but from (s,q) no accepting BSCC can be reached. Choose $\nu(t^M_{(s,q),(s',q')}) = 0$ and $\nu(r^M_{(s,q)}) = 0$ as in the previous case. Since $\nu(\sigma_{(s,q),\alpha}) = 0$ for all $\alpha \in Act$, the three constraints are satisfied.
- $s \in S'$ and from (s,q) a BSCC can be reached. According to Lemma 9 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. □

Lemma 15 *Let $\mathcal{M}$ be an MDP and $\mathcal{P}_{\leq\lambda}(\mathcal{L})$ a violated ω-regular property. Let furthermore $\mathcal{A} = (Q, q_{\text{init}}, 2^{\text{AP}}, \delta, f)$ be a DRA with $F = \{(R_i, A_i) \mid i = 1,\dots,n\}$. Assume that ν is a satisfying assignment of the MILP* (9b)–(9r)*. Then the subsystem (S', A) with $S' = \{s \in S \mid \nu(x_s) = 1\}$ and $A(s) = \{\alpha \in Act \mid \exists q \in Q : \nu(\sigma_{(s,q),\alpha}) = 1\}$ forms a critical subsystem of $\mathcal{M}$.*

PROOF. We define the following subsystem $\mathcal{M}' = (S', s_{\text{init}}, Act, P', L')$ with $S' = \{s \in S \mid \nu(x_s) = 1\}$, $A(s) = \{\alpha \in Act \mid \exists q \in Q : \nu(\sigma_{(s,q),\alpha}) = 1\}$,

$$P'(s,\alpha,s') = \begin{cases} P(s,\alpha,s') & \text{if } \alpha \in A(s), \\ 0 & \text{otherwise} \end{cases}$$

and $L'(s) = L(s)$ for all $s, s' \in S'$.

Consider the product automaton $\mathcal{M}' \otimes \mathcal{A}$. We define a memoryless deterministic scheduler on $\mathcal{M}' \otimes \mathcal{A}$ by $\sigma(s,q) = \alpha$ iff $\nu(\sigma_{(s,q),\alpha}) = 1$. States (s,q), for which σ is not defined that way, are removed from $\mathcal{M} \otimes \mathcal{A}$. For all states (s,q) which are not contained in the so constructed system, $\nu(p_{(s,q)}) = 0$ holds: Either $s \notin S'$, then (9m) ensures $\nu(p_{(s,q)}) = 0$, otherwise, if (s,q) was removed because $\nu(\sigma_{(s,q),\alpha}) = 0$ for all $\alpha \in Act$, this is ensured by constraint (9c). Note that removing states and transition cannot increase reachability probabilities. The scheduler is well-defined, since we requested in (9b) that $\nu(\sigma_{(s,q),\alpha}) = 1$ for at most one action $\alpha \in Act$.

For $i = 1,\dots,n$ set $M_i = \{(s,q) \in S \times Q \mid \nu(m^i_{(s,q)}) = 1\}$. We have that $M_i \subseteq S' \times Q$ since (9k) forces all states to be included in the subsystem if they are in M_i for some i.

M_i contains only states for which

$$\sum_{(s',q')\in \mathrm{succ}_{\mathcal{M}\otimes\mathcal{A}}((s,q),\sigma(s,q))} P'((s,q),\sigma(s,q),(s',q')) = 1$$

holds due to (9d). M_i is closed under successors w. r. t. the actions selected by σ because of (9e). Furthermore, M_i does not contain an R_i-state according to (9f). Given the assignment of $\sigma_{(s,q),\alpha}$, constraints (9g)–(9i) are backward reachability constraints with the A_i-states as the target states. According to Lemma 8, an assignment ν is satisfying these constraints iff from all states (s,q) with $\nu(m^i_{(s,q)}) = 1$ a target state is reachable. Therefore all prerequisites of Lemma 1 are fulfilled. It follows that $\Pr^{(s,q)}_{\mathcal{M}'\otimes\mathcal{A}}(\Diamond\, accept) = 1$ for all states $(s,q) \in \bigcup_{i=1}^{n} M_i$, which coincides with $\nu(p_{(s,q)})$ because of (9k).

We set $M = \bigcup_{i=1}^{n} M_i$. Now consider the linear equation system describing the probabilities of reaching an M-state:

$$p_{(s,q)} = \begin{cases} 1 & \text{if } (s,q) \in M, \\ 0 & \text{if } M \text{ is unreachable from } (s,q), \\ \sum\limits_{(s',q')\in \mathrm{succ}_{\mathcal{M}'\otimes\mathcal{A}}((s,q),\sigma(s,q))} P'((s,q),\sigma(s,q),(s',q')) \cdot p_{(s',q')} & \text{otherwise.} \end{cases}$$

For states $(s,q) \in M$, $\nu(p_{(s,q)})$ coincides with the solution of this equation system because of (9g). For states (s,q) from which M is unreachable, the backward reachability constraints (9p)–(9r) are only satisfiable iff $\nu(\sigma_{(s,q),\alpha}) = 0$ for all $\alpha \in Act$. Then $\nu(p_{(s,q)}) = 0$ (constraint (9c)). For the remaining states (s,q) and the action $\alpha = \sigma(s,q)$ the following constraint (9m) is satisfied:

$$p_{(s,q)} \leq \sum_{(s',q')\in \mathrm{succ}_{\mathcal{M}'\otimes\mathcal{A}}((s,q),\alpha)} P((s,q),\alpha,(s',q')) \cdot p_{(s',q')}\,.$$

According to Lemma 3, the solution of the equation system is not smaller than the solution $\nu(p_{(s,q)_{\mathrm{init}}})$ of the inequality, which satisfies $\nu(p_{(s,q)_{\mathrm{init}}}) > \lambda$. Therefore the subsystem is critical. □