A Counterexample Guided Abstraction-Refinement Framework for Markov Decision Processes

ROHIT CHADHA and MAHESH VISWANATHAN

Dept. of Computer Science, University of Illinois at Urbana-Champaign

The main challenge in using abstractions effectively, is to construct a suitable abstraction for the system being verified. One approach that tries to address this problem is that of *counterexample guided abstraction-refinement (CEGAR)*, wherein one starts with a coarse abstraction of the system, and progressively refines it, based on invalid counterexamples seen in prior model checking runs, until either an abstraction proves the correctness of the system or a valid counterexample is generated. While CEGAR has been successfully used in verifying non-probabilistic systems automatically, CEGAR has only recently been investigated in the context of probabilistic systems. The main issues that need to be tackled in order to extend the approach to probabilistic systems is a suitable notion of "counterexample", algorithms to generate counterexamples, check their validity, and then automatically refine an abstraction based on an invalid counterexample. In this paper, we address these issues, and present a CEGAR framework for Markov Decision Processes.

Categories and Subject Descriptors: D.2.4 [**Software Engineering**]: Program Verification

General Terms: Verification

1. INTRODUCTION

Abstraction is an important technique to combat *state space explosion*, wherein a smaller, abstract model that conservatively approximates the behaviors of the original (concrete) system is verified/model checked. The main challenge in applying this technique in practice, is in constructing such an abstract model. *Counterexample guided abstraction-refinement (CEGAR)* [Clarke et al. 2000] addresses this problem by constructing abstractions *automatically* by starting with a coarse abstraction of the system, and progressively refining it, based on invalid counterexamples seen in prior model checking runs, until either an abstraction proves the correctness of the system or a valid counterexample is generated.

While CEGAR has been successfully used in verifying non-probabilistic systems automatically, until recently, CEGAR has not been applied in the context of systems that exhibit probabilistic behavior. In order to extend this approach to probabilistic systems, one needs to identify a family of abstract models, develop a suitable notion of counterexamples, and design algorithms to produce counterexamples from

This work was partially supported by NSF grants CCF 0429639, CCF 0448178, and CNS 0509321.

erroneous abstractions, check their validity in the original system, and (if needed) automatically refine an abstraction based on an invalid counterexample. In this paper we address these issues, and develop a CEGAR framework for systems described as *Markov Decision Processes (MDP).*

Abstractions have been extensively studied in the context of probabilistic systems with definitions for good abstractions and specific families of abstractions being identified (see Section 6). In this paper, like [Jonsson and Larsen 1991; D'Argenio et al. 2001; 2002], we use Markov decision processes to abstract other Markov decision processes. The abstraction will be defined by an equivalence relation (of finite index) on the states of the concrete system. The states of the abstract model will be the equivalence classes of this relation, and each abstract state will have transitions corresponding to the transitions of each of the concrete states in the equivalence class.

Crucial to extending the CEGAR approach to probabilistic systems is to come up with an appropriate notion of counterexamples. [Clarke et al. 2002] have identified a clear set of metrics by which to evaluate any proposal for counterexamples. Counterexamples must satisfy three criteria: (a) counterexamples should serve as an "explanation" of why the (abstract) model violates the property, (b) must be rich enough to explain the violation of a large class of properties, and (c) must be simple and specific enough to identify bugs, and be amenable to efficient generation and analysis.

With regards to probabilistic systems there are three compelling proposals for counterexamples to consider. The first, originally proposed in [Han and Katoen 2007a] for DTMCs, is to consider counterexamples to be a multi-set of executions. This has been extended to CTMCs [Han and Katoen 2007b], and MDPs [Aljazzar and Leue 2007]. The second is to take counterexamples to be MDPs with a *tree-like* graph structure, a notion proposed by [Clarke et al. 2002] for non-probabilistic systems and branching-time logics. The third and final notion, suggested in [Chatterjee et al. 2005; Hermanns et al. 2008], is to view general DTMCs (i.e., models without non-determinism) as counterexamples. We show that all these proposals are expressively inadequate for our purposes. More precisely, we show that there are systems and properties that do not admit any counterexamples of the above special forms.

Having demonstrated the absence of counterexamples with special structure, we take the notion of counterexamples to simply be "small" MDPs that violate the property and are simulated by the abstract model. Formally, a counterexample for a system $\mathcal{M}$ and property ψ_S will be a pair $(\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ is a MDP violating the property ψ_S that is *simulated* by $\mathcal{M}$ via the relation $\mathcal{R}$. The simulation relation has rarely been thought of as being formally part of the counterexample; requiring this addition does not change the asymptotic complexity of counterexample generation, since the simulation relation can be computed efficiently [Baier et al. 2000], and for the specific context of CEGAR, they are merely simple "injection functions". However, as we shall point out, defining counterexamples formally in this manner makes the technical development of counterexample guided refinement cleaner (and is, in fact, implicitly assumed to be part of the counterexample, in the case of non-probabilistic systems).

One crucial property that counterexamples must exhibit is that they be amenable to efficient generation and analysis [Clarke et al. 2002]. We show that generating the *smallest* counterexample is NP-complete. Moreover it is unlikely to be efficiently approximable. However, in spite of these negative results, we show that there is a very simple polynomial time algorithm that generates a *minimal* counterexample; a minimal counterexample is a pair $(\mathcal{E}, \mathcal{R})$ such that if any state/transition of $\mathcal{E}$ is removed, the resulting MDP no longer violates the property.

Intuitively, a counterexample is valid if the original system can exhibit the "behavior" captured by the counterexample. For non-probabilistic systems [Clarke et al. 2000; Clarke et al. 2002], a valid counterexamples is not simply one that is simulated by the original system, even though simulation is the formal concept that expresses the notion of a system exhibiting a behavior. One requires that the original system simulate the counterexample, "in the same manner as the abstract system". More precisely, if $\mathcal{R}$ is the simulation relation that witnesses $\mathcal{E}$ being simulated by the abstract system, then $(\mathcal{E}, \mathcal{R})$ is valid if the original system simulates $\mathcal{E}$ through a simulation relation that is "contained within" $\mathcal{R}$. This is one technical reason why we consider the simulation relation to be part of the concept of a counterexample. Thus the algorithm for checking validity is the same as the algorithm for checking simulations between MDPs [Baier et al. 2000; Zhang et al. 2007] except that we have to ensure that the witnessing simulation be "contained within R". However, because of the special nature of counterexamples, better bounds on the running time of the algorithm can be obtained.

Finally we outline how the abstraction can be automatically refined. Once again the algorithm is a natural generalization of the refinement algorithm in the non-probabilistic case, though it is different from the refinement algorithms proposed in [Chatterjee et al. 2005; Hermanns et al. 2008]; detailed comparison can be found in Section 6. We also state and prove precisely what the refinement algorithm achieves.

1.1 Our Contributions

We now detail our main technical contributions, roughly in the order in which they appear in the paper.

(1) For MDPs, we identify safety and liveness fragments of PCTL. Our fragment is syntactically different than that presented in [Desharnais 1999b; Baier et al. 2005] for DTMCs. Though the two presentations are semantically the same for DTMCs, they behave differently for MDPs.

(2) We demonstrate the expressive inadequacy of all relevant proposals for counterexamples for probabilistic systems, thus demonstrating that counterexamples with special graph structures are unlikely to be rich enough for the safety fragment of PCTL.

(3) We present formal definitions of counterexamples, their validity and consistency, and the notion of good counterexample-guided refinements. We distill a precise statement of what the CEGAR-approach achieves in a single abstraction-refinement step. Thus, we generalize concepts that have been hither-to only defined for "path-like" structures [Clarke et al. 2000; Clarke et al. 2002; Han and Katoen 2007a; 2007b; Aljazzar and Leue 2007; Hermanns et al. 2008] to general graph-like

structures [1], and for the first time formally articulate, what is accomplished in a single abstraction-refinement step.

(4) We present algorithmic solutions to all the computational problems that arise in the CEGAR loop: we give lower bounds as well as upper bounds for counterexample generation, and algorithms to check validity and to refine an abstraction.

(5) A sub-logic of our safe-PCTL, which we call weak safety, does indeed admit counterexamples that have a tree-like structure. For this case, we present an on-the-fly algorithm to unroll the minimal counterexample that we generate and check validity. This algorithm may perform better than the algorithm based on checking simulation for some examples in practice.

Though our primary contributions are to clarify the definitions and concepts needed to carry out CEGAR in the context of probabilistic systems, our effort also sheds light on implicit assumptions made by the CEGAR approach for non-probabilistic systems.

1.2 Outline of the Paper

The rest of the paper is organized as follows. We recall some definitions and notations in Section 2. We also present safety and liveness fragments of PCTL for MDP's in Section 2. We discuss various proposals of counterexamples for MDP's in Section 3, and also present our definition of counterexamples along with algorithmic aspects of counterexample generation. We recall the definition of abstractions based on equivalences in Section 4. We present the definitions of validity and consistency of abstract counterexamples and good counterexample-guided refinement, as well as the algorithms to check validity and refine abstractions in Section 5. Finally, related work is discussed in Section 6.

2. PRELIMINARIES

The paper assumes familiarity with basic probability theory, discrete time Markov chains, Markov decision processes, and the model checking of these models against specifications written in PCTL; the background material can be found in [Rutten et al. 2004]. This section is primarily intended to introduce notation, and to introduce and remind the reader of results that the paper will rely on.

2.1 Relations and Functions.

We assume that the reader is familiar with the basic definitions of relation and functions. We will primarily be interested in binary relations. We will use $\mathcal{R}, S, T, \ldots$ to range over relations and $f, g, h, \ldots$ to range over functions. We introduce here some notations that will be useful.

Given a set A, we will denote its power-set by 2^A. For a finite set A, the number of elements in A will be denoted by $|A|$.

The identity function on a set A will be often denoted by id_A. Given a function $f : A \to B$ and set $A' \subseteq A$, the restriction of f to A' will be denoted by $f|_{A'}$.

[1] Even when a counterexample is not formally a path, as in [Clarke et al. 2002] and [Hermanns et al. 2008], it is viewed as a collection of paths and simple cycles, and all concepts are defined for the case when the cycles have been unrolled a finite number of times.

For a binary relation $\mathcal{R} \subseteq A \times B$ we will often write $a \, \mathcal{R} \, b$ to mean $(a, b) \in \mathcal{R}$. Also, given $a \in A$ we will denote the set $\{b \in B \mid a \, \mathcal{R} \, b\}$ by $\mathcal{R}(a)$. Please note $\mathcal{R}$ is uniquely determined by the collection $\{\mathcal{R}(a) \mid a \in A\}$. A binary relation $\mathcal{R}_1 \subseteq A \times B$ is said to be *finer* than $\mathcal{R}_2 \subseteq A \times B$ if $\mathcal{R}_1 \subseteq \mathcal{R}_2$. The composition of two binary relations $\mathcal{R}_1 \subseteq A \times B$ and $\mathcal{R}_2 \subseteq B \times C$, denoted by $\mathcal{R}_2 \circ \mathcal{R}_1$, is the relation $\{(a, c) \mid \exists b \in B.\ a\mathcal{R}_1 b \text{ and } b\mathcal{R}_2 c\} \subseteq A \times C$.

We say that a binary relation $\mathcal{R} \subseteq A \times B$ is *total* if for all $a \in A$ there is a $b \in B$ such that $a \, \mathcal{R} \, b$. We say that a binary relation $\mathcal{R} \subseteq A \times B$ is *functional* if for all $a \in A$ there is at most one $b \in B$ such that $a \, \mathcal{R} \, b$. There is a close correspondence between functions and total, functional relations: for any function $f : A \to B$, the relation $\{(a, f(a)) \mid a \in A\}$ is a total and functional binary relation. Vice-versa, one can construct a unique function from a given total and functional binary relation. We will denote the total and functional relation given by a function f by rel_f.

A *preorder* on a set A is a binary relation that is reflexive and transitive. An equivalence relation on a set A is a preorder which is also symmetric. The *equivalence class* of an element $a \in A$ with respect to an equivalence relation $\equiv$, will be denoted by $[a]_\equiv$; when the equivalence relation $\equiv$ is clear from the context we will drop the subscript $\equiv$.

2.2 DTMC and MDP

Sequences. For a set X, we will denote the set of non-empty finite sequences of elements of X by X^+. The set X^ω will be the set of all countably infinite sequences of elements of X. For a finite sequence $\eta \in X^+$ and an infinite sequence $\alpha \in X^\omega$ we denote by $\eta\alpha \in X^\omega$ the sequence obtained by concatenating the sequences η and α in order.

Basic Probability Theory. Let X be a set and E be a set of subsets of X. We say that E is a σ-algebra on X if E contains the empty set, is closed under complementation and also under countable unions. Let F be a set of subsets of X. A σ-algebra generated by F is the smallest σ-algebra that contains F. A (sub)-probability space is a triple (X, E, μ) where E is a σ-algebra on X and μ is a (sub)-probability function [Papoulis and Pillai 2002] on E.

For (finite or countable) set X with σ-field 2^X, the collection all sub-probability measures (i.e., where measure of $X \leq 1$) will be denoted by $\mathrm{Prob}_{\leq 1}(X)$. For $\mu \in \mathrm{Prob}_{\leq 1}(X)$, and $A \subseteq X$, $\mu(A)$ denotes the measure of set A.

Kripke structures. A *Kripke structure* over a set of propositions AP, is formally a tuple $\mathcal{K} = (\mathsf{Q}, q_\mathcal{I}, \to, \mathsf{L})$ where Q is a set of states, $q_\mathcal{I} \in \mathsf{Q}$ is the initial state, $\to \subseteq \mathsf{Q} \times \mathsf{Q}$ is the transition function, and $\mathsf{L} : \mathsf{Q} \to 2^{\mathsf{AP}}$ is a labeling function that labels each state with the set of propositions true in it. DTMC and MDP are generalizations of Kripke structures where transitions are replaced by probabilistic transitions.

Discrete Time Markov Chains. A *discrete time Markov chain* (DTMC) over a set of propositions AP, is formally a tuple $\mathcal{M} = (\mathsf{Q}, q_\mathcal{I}, \delta, \mathsf{L})$ where Q is a (finite or countable) set of states, $q_\mathcal{I} \in \mathsf{Q}$ is the initial state, $\delta : \mathsf{Q} \to \mathrm{Prob}_{\leq 1}(\mathsf{Q})$ is the transition function, and $\mathsf{L} : \mathsf{Q} \to 2^{\mathsf{AP}}$ is a labeling function that labels each state with the set of propositions true in it. A DTMC is said to be *finite* if the set Q is finite. Unless otherwise explicitly stated, DTMC's in this paper will be assumed to be finite.

DTMCs generate a probability space (X, E, μ) as follows [Kemeny and Snell 1976; Rutten et al. 2004]. First of all, we pick a new state $\perp \notin \mathsf{Q}$. The set X is $(\mathsf{Q} \cup \{\perp\})^\omega$ and will henceforth be called the set of *paths*. Let E be the σ-algebra generated by the set $\{C_\eta \mid \eta \in (\mathsf{Q} \cup \{\perp\})^+\}$ where $C_\eta = \{\eta\alpha | \alpha \in (\mathsf{Q} \cup \{\perp\})^+\}$. The measure μ is the unique measure such that for each $\eta = q_0 q_1 \dots q_n \in Q^+$, $\mu(C_\eta)$ is

—0 if $q_0 \neq q_\mathcal{I}$

—1 if $n = 0$ and $q_0 = q_\mathcal{I}$, and

—$\prod_{0 \leq i < n} pr(q_i, q_{i+1})$ otherwise, where $pr(q_i, q_{i+1})$ is
 —$\delta(q_i)(q_{i+1})$ if $q_i, q_{i+1} \in \mathsf{Q}$,
 —$1 - \delta(q_i)(\mathsf{Q})$ if $q_i \in \mathsf{Q}$ and $q_{i+1} = \perp$,
 —1 if $q_i = q_{i+1} = \perp$, and
 —0 if $q_i = \perp$ and $q_{i+1} \in \mathsf{Q}$.

We omit the details of the construction and refer the reader to [Rutten et al. 2004] for the complete construction.

Remark 2.1. We point out here that sometimes instead of the states, the labeling function labels the transitions [Segala and Lynch 1995; Desharnais et al. 2000].

Markov Decision Processes. A *finite Markov decision process* (MDP) over a set of propositions AP, is formally a tuple $\mathcal{M} = (\mathsf{Q}, q_\mathcal{I}, \delta, \mathsf{L})$ where $\mathsf{Q}, q_\mathcal{I}, \mathsf{L}$ are as in the case for finite DTMCs, and $\delta : q \to 2^{\mathrm{Prob}_{\leq 1}(\mathsf{Q})}$ maps each state to a *finite* non-empty collection of sub-probability measures. We will sometimes say that there is no transition out of $q \in \mathsf{Q}$ if $\delta(q)$ consists of exactly one sub-probability measure $\mathbf{0}$ which assigns 0 to all states in Q. For this paper, we will assume that for every q, q' and every $\mu \in \delta(q)$, $\mu(q')$ is a rational number. From now on, we will explicitly drop the qualifier "finite" for MDP's. Note that a finite DTMC can be viewed as a special case of a MDP, namely one in which the $\delta(q)$ consists of exactly one element for each state q.

We assume that the reader is familiar with the concept of *schedulers*. A scheduler resolves non-determinism. Formally, a scheduler is a function $\mathcal{S} : \mathsf{Q}^+ \to \mathrm{Prob}_{\leq 1}(\mathsf{Q})$ such that for any $q_0 q_1 \dots q_n \in Q^+$, $\mathcal{S}(q_0 q_1 \dots q_n) \in \delta(q_n)$. Given a scheduler and a MDP $\mathcal{M}$, we get a probability space $(X, E, \mu_\mathcal{S})$ as follows. As in the case of DTMCs, the set X is $(\mathsf{Q} \cup \{\perp\})^\omega$ where $\perp \notin \mathsf{Q}$. E is the σ-algebra generated by the set $\{C_\eta \mid \eta \in (\mathsf{Q} \cup \{\perp\})^+\}$ where $C_\eta = \{\eta\alpha | \alpha \in (\mathsf{Q} \cup \{\perp\})^+\}$. The measure $\mu_\mathcal{S}$ is the unique measure such that for each $\eta = q_0 q_1 \dots q_n \in Q^+$, $\mu_\mathcal{S}(C_\eta)$ is

—0 if $q_0 \neq q_\mathcal{I}$

—1 if $n = 0$ and $q_0 = q_\mathcal{I}$, and

—$\prod_{0 \leq i < n} pr_{(i,i+1)}$ otherwise, where $pr_{(i,i+1)}$ is
 —$\mu_\mathcal{S}(q_0 q_1 \dots q_i)(q_{i+1})$ if $q_i, q_{i+1} \in \mathsf{Q}$,
 —$1 - \mathcal{S}(q_0 q_1 \dots q_i)(\mathsf{Q})$ if $q_i \in \mathsf{Q}$ and $q_{i+1} = \perp$,
 —1 if $q_i = q_{i+1} = \perp$, and
 —0 if $q_i = \perp$ and $q_{i+1} \in \mathsf{Q}$.

A scheduler $\mathcal{S}$ is said to be *memoryless* if for each $\eta = q_0 q_1 \dots q_n$ and $\eta' = q'_0 q'_1 \dots q'_l$, $\mathcal{S}(\eta) = \mathcal{S}(\eta')$ whenever $q_n = q'_l$. So, a memoryless scheduler is completely specified by a function $tr_{\mathcal{S}} : \mathsf{Q} \to \mathrm{Prob}_{\leq 1}(\mathsf{Q})$, where $tr_{\mathcal{S}}(q)$ gives the transition $\mathcal{S}(q_0 q_1 \dots q_n)$ whenever $q_n = q$. We call $tr_{\mathcal{S}}(q)$, the *transition out of* q.

Remark 2.2. In the presence of scheduler $\mathcal{S}$ the MDP can also be thought of as a a (countable) DTMC, $\mathcal{M}^{\mathcal{S}} = \{\mathsf{Q}^+, q_{\mathcal{I}}, \delta^{\mathcal{S}}, L^{\mathcal{S}}\}$ where

—$L^{\mathcal{S}}(q_0 q_1 \dots q_n) = L(q_n)$ and

—$\delta^{\mathcal{S}}(q_0 q_1 \dots q_n)$ is the sub-probability measure $\mu^{\mathcal{S}} \in \mathrm{Prob}_{\leq 1}(\mathsf{Q}^+)$ such that for each $q'_0 q'_1 \dots q'_l \in Q^+$, $\mu^{\mathcal{S}}(q'_0 q'_1 \dots q'_l)$ is

  —$\mathcal{S}(q_0 q_1 \dots q_n)(q'_l)$ if $l = n+1$ and $q_i = q'_i$ for each $0 \leq i \leq n$, and

  —0 otherwise.

In the presence of a memoryless scheduler $\mathcal{S}$, the resulting DTMC $\mathcal{M}^{\mathcal{S}}$ is *bisimilar* to a finite DTMC which has the same set of states as $\mathcal{M}$, the same initial state and the same labeling function, while the transition out of a state q is the one given by the memoryless scheduler $\mathcal{S}$. We will henceforth confuse this finite DTMC with $\mathcal{M}^{\mathcal{S}}$.

Suppose there are at most k nondeterministic choices from any state in $\mathcal{M}$. For some ordering of the nondeterministic choices out of each states, the *labeled underlying graph* of a MDP is the directed graph $G = (\mathsf{Q}, \{E_i\}_{i=1}^{k})$, where $(q_1, q_2) \in E_i$ iff $\mu(q_2) > 0$, where μ is the ith choice out of q_1; we will denote the labeled underlying graph of $\mathcal{M}$ by $G_\ell(\mathcal{M})$. The *unlabeled underlying graph* will be $G' = (\mathsf{Q}, \cup_{i=1}^{k} E_i)$ and is denoted by $G(\mathcal{M})$. The following notation will be useful.

Notation 2.3. Given a MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$, a state $q \in \mathsf{Q}$ and a transition $\mu \in \delta(q)$, we say that $\mathsf{post}(\mu, q) = \{q' \in \mathsf{Q} \mid \mu(q') > 0.\}$

PCTL. We recall here the logic PCTL [Rutten et al. 2004] briefly. The formulas of the logic PCTL are generated over a set of propositions AP as follows–

$$\psi := \mathsf{tt} \; [] \; \mathsf{ff} \; [] \; P \; [] \; (\neg\psi) \; [] \; (\psi \wedge \psi) \; [] \; (\psi \vee \psi) \; [] \; \mathcal{P}_{\lhd p}(\mathsf{X}\,\psi) \; [] \; \mathcal{P}_{\lhd p}(\psi \; \mathcal{U} \; \psi)$$

where $P \in \mathsf{AP}$, $p \in [0,1]$ is a rational number and $\lhd \in \{<, \leq\}$. The logic is interpreted over DTMCs and MDPs and the details of the interpretation are given in an Electronic Appendix, for the sake of the flow of the paper. Given a MDP $\mathcal{M}$ and a state q of $\mathcal{M}$, we say $q \Vdash_{\mathcal{M}} \psi$ if q satisfies the formula ψ. We will drop $\mathcal{M}$ when clear from the context. We will say that $\mathcal{M} \Vdash \psi$ if the initial state of $\mathcal{M}$ satisfies the formula. The model checking problem for MDPs and PCTL is known to be in polynomial time [Bianco and de Alfaro 1995]. We also point out PCTL is also defined for MDP's in which the labels are on the transitions instead of the state, as in [Segala and Lynch 1994]. In [Segala and Lynch 1994], transition labels instead of propositions are used as formulas of PCTL , and a state q satisfies a transition label a if all the actions out of q a labeled a.

Unrolling of a MDP. Given a $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$, natural number $k \geq 0$, and $q \in \mathsf{Q}$ we will define a MDP $\mathcal{M}_k^q = (\mathsf{Q}_k^q, (q,k), \delta_k^q, \mathsf{L}_k^q)$ obtained by unrolling the underlying labeled graph of $\mathcal{M}$ up-to depth k. Formally, $\mathcal{M}_k^q = (\mathsf{Q}_k^q, (q,k), \delta_k^q, \mathsf{L}_k^q)$, the *k-th unrolling of* $\mathcal{M}$ *rooted at* q is defined by induction as follows.

—$\mathsf{Q}^q_k = \{(q,k)\} \cup (\mathsf{Q} \times \{j \in \mathbb{N} \mid 0 \leq j < k\})$.

—For all $(q',j) \in \mathsf{Q}^q_k$, $\mathsf{L}((q',j)) = \mathsf{L}(q')$.

—For all $(q',j) \in \mathsf{Q}^q_k$, $\delta^q_k((q',j)) = \{\mu^j \mid \mu \in \delta(q')\}$ where μ^j is defined as–

(1) $\mu^0(q'') = 0$ for all $q'' \in \mathsf{Q}^q_k$, and

(2) for $0 \leq j < k$, $\mu^{j+1}(q'') = \mu(q')$ if $q'' = (q',j)$ for some $q' \in \mathsf{Q}$ and 0 otherwise.

Please note that the underlying unlabeled graph of $\mathcal{M}^q_k$ is (directed) acyclic.

Direct Sum of MDP's. Given a MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$ over the set of propositions AP, let $\mathsf{Q}+\mathsf{Q}' = \mathsf{Q}\times\{0\} \cup \mathsf{Q}'\times\{1\}$ be the disjoint sum of Q and Q'. Now, define $\delta+\delta' : \mathsf{Q}+\mathsf{Q}' \to \mathrm{Prob}_{\leq 1}(\mathsf{Q}+\mathsf{Q}')$ and $\mathsf{L}+\mathsf{L}' : \mathsf{Q}+\mathsf{Q}' \to \mathsf{AP}$ as follows. For all $q \in \mathsf{Q}$ and $q' \in \mathsf{Q}'$,

—$(\delta+\delta')((q,0)) = \{\mu\times\{0\} \mid \mu \in \delta(q)\}$ and $(\delta+\delta')((q',1)) = \{\mu'\times\{1\} \mid \mu' \in \delta'(q')\}$ where $\mu\times\{0\}$ and $\mu'\times\{1\}$ are defined as follows.

—$\mu\times\{0\}(q_1,0) = \mu(q_1)$ and $\mu\times\{0\}(q'_1,1) = 0$ for all $q_1 \in \mathsf{Q}_1$ and $q'_1 \in \mathsf{Q}_2$.

—$\mu'\times\{1\}(q_1,0) = 0$ and $\mu'\times\{1\}(q'_1,1) = \mu'(q'_1)$ for all $q_1 \in \mathsf{Q}_1$ and $q'_1 \in \mathsf{Q}_2$.

—$(\mathsf{L}+\mathsf{L}')(q,0) = \mathsf{L}(q)$ and $(\mathsf{L}+\mathsf{L}')(q',1) = \mathsf{L}'(q')$.

Now given $q \in \mathsf{Q}+\mathsf{Q}'$, the MDP $(\mathcal{M}+\mathcal{M}')_q = (\mathsf{Q}+\mathsf{Q}', q, \delta+\delta', \mathsf{L}+\mathsf{L}')$ is said to be the *direct sum* of $\mathcal{M}$ and $\mathcal{M}'$ with q as the initial state.

Remark 2.4. MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$ are said to be *disjoint* if $\mathsf{Q}\cap\mathsf{Q}' = \emptyset$. If MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$ are disjoint, then $\mathsf{Q}+\mathsf{Q}'$ can be taken to be the union $\mathsf{Q}\cup\mathsf{Q}'$. In such cases, we will confuse $(q,0)$ with q, $(q',1)$ with q', $\mu\times\{0\}$ with μ and $\mu'\times\{1\}$ with μ' (with the understanding that $\mu \in \delta(q)$ takes value 0 on any $q' \in \mathsf{Q}'$ and $\mu \in \delta(q')$ takes value 0 on any $q \in \mathsf{Q}$).

2.3 Simulation

Given a binary relation $\mathcal{R}$ on the set of states Q , a set $\mathsf{A} \subseteq \mathsf{Q}$, is said to be $\mathcal{R}$-*closed* if the set $\mathcal{R}(\mathsf{A}) = \{t \mid \exists q \in \mathsf{A},\, q\mathcal{R}t\}$ is the same as A. For two sub-probability measures $\mu,\mu' \in \mathrm{Prob}_{\leq 1}(\mathsf{Q})$, we say μ' *simulates* μ *with respect to* a preorder $\mathcal{R}$ (denoted as $\mu \preceq_{\mathcal{R}} \mu'$) iff for every $\mathcal{R}$-closed set A, $\mu(\mathsf{A}) \leq \mu'(\mathsf{A})$. For a MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$, a preorder $\mathcal{R}$ on Q is said to be a *simulation relation*[2] if for every $q\,\mathcal{R}\,q'$, we have that $\mathsf{L}(q) = \mathsf{L}(q')$ and for every $\mu \in \delta(q)$ there is a $\mu' \in \delta(q')$ such that $\mu \preceq_{\mathcal{R}} \mu'$. We say that $q \preceq q'$ if there is a simulation relation $\mathcal{R}$ such that $q\,\mathcal{R}\,q'$.

Given an equivalence relation $\equiv$ on the set of states Q, and two sub-probability measures $\mu,\mu' \in \mathrm{Prob}_{\leq 1}(\mathsf{Q})$ we say that μ *is equivalent to* μ' with respect to $\equiv$ (denoted as $\mu \approx_{\equiv} \mu'$) iff for every $\equiv$-closed set A, $\mu(\mathsf{A}) = \mu'(\mathsf{A})$. For a MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$, an equivalence $\equiv$ on Q is said to be a *bisimulation* if for every $q \equiv q'$, we have that $\mathsf{L}(q) = \mathsf{L}(q')$ and for every $\mu \in \delta(q)$ there is a $\mu' \in \delta(q')$ such

[2] It is possible to require only that $\mathsf{L}(q) \subseteq \mathsf{L}(q')$ instead of $\mathsf{L}(q) = \mathsf{L}(q')$ in the definition of simulation. The results and proofs of the paper could be easily adapted for this definition. One has to modify the definition of safety and liveness fragments of PCTL appropriately.

that $\mu \approx_{\equiv} \mu'$. We say that $q \approx q'$ if there is a bisimulation relation $\equiv$ such that $q \equiv q'$.

Remark 2.5. The ordering on probability measures used in the definition of simulation presented in [Jonsson and Larsen 1991; Segala and Lynch 1994; Baier et al. 2005] is based on *weight functions*. However, the definition presented here, was originally proposed in [Desharnais 1999a] and shown to be equivalent [Desharnais 1999a; Segala 2006].

We say that MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ is simulated by $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$ (denoted by $\mathcal{M} \preceq \mathcal{M}'$) if there is a simulation relation $\mathcal{R}$ on the direct sum of $\mathcal{M}$ and $\mathcal{M}'$ (with any initial state) such that $(q_{\mathcal{I}}, 0)\ \mathcal{R}\ (q'_{\mathcal{I}}, 1)$. The MDP $\mathcal{M}$ is said to be bisimilar to $\mathcal{M}'$ (denoted by $\mathcal{M} \approx \mathcal{M}'$) if there is a bisimulation relation $\equiv$ on the direct sum of $\mathcal{M}$ and $\mathcal{M}'$ (with any initial state) such that $(q_{\mathcal{I}}, 0) \equiv (q'_{\mathcal{I}}, 1)$.

As an example of simulations, we have that every MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ simulates its k-th unrolling. Furthermore, we also have that if $k \leq k'$ then the k'-th unrolling simulates the k-th unrolling.

PROPOSITION 2.6. *Given a MDP $\mathcal{M}$ with initial state $q_{\mathcal{I}}$ and natural numbers $k, k' \geq 0$ such that $k \leq k'$. Let $\mathcal{M}_k^{q_{\mathcal{I}}}$ and $\mathcal{M}_{k'}^{q_{\mathcal{I}}}$ be the k-th and k'-unrolling of $\mathcal{M}$ rooted at $q_{\mathcal{I}}$ respectively. Then $\mathcal{M}_k^{q_{\mathcal{I}}} \preceq \mathcal{M}$ and $\mathcal{M}_k^{q_{\mathcal{I}}} \preceq \mathcal{M}_{k'}^{q_{\mathcal{I}}}$.*

Simulation between disjoint MDP's. We will be especially interested in simulation between disjoint MDP's (in which case we can just take the union of state spaces of the MDP's as the state space of the direct sum). The simulations will also take a certain form which we will call *canonical form* for our purposes. In order to define this precisely, recall that for any set A, id_A is the identity function on A and that rel_{id_A} is the relation $\{(a, a) \mid a \in A\}$.

Definition 2.7. Given disjoint MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$, we say that a simulation relation $\mathcal{R} \subseteq (\mathsf{Q} + \mathsf{Q}') \times (\mathsf{Q} + \mathsf{Q}')$ on the direct sum of Q and Q' is in *canonical form* if there exists a relation $\mathcal{R}_1 \subseteq \mathsf{Q} \times \mathsf{Q}'$ such that $\mathcal{R} = \mathsf{rel}_{id_{\mathsf{Q}}} \cup \mathcal{R}_1 \cup \mathsf{rel}_{id_{\mathsf{Q}'}}$.

The following proposition states that any simulation contains a largest canonical simulation and hence canonical simulations are sufficient for reasoning about simulation between disjoint MDP's whose proof has been moved to an Electronic Appendix for the sake of the flow of the paper.

PROPOSITION 2.8. *Given disjoint MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$, let $\mathcal{R} \subseteq (\mathsf{Q} + \mathsf{Q}') \times (\mathsf{Q} + \mathsf{Q}')$ be a simulation relation on the direct sum of Q and Q'. Let $\mathcal{R}_1 = \mathcal{R} \cap (\mathsf{Q} \times \mathsf{Q}')$. Then the relation $\mathcal{R}_0 = \mathsf{rel}_{id_{\mathsf{Q}}} \cup \mathcal{R}_1 \cup \mathsf{rel}_{id_{\mathsf{Q}'}}$ is a simulation relation.*

Notation 2.9. In order to avoid clutter, we will often denote a simulation $\mathsf{rel}_{id_{\mathsf{Q}}} \cup \mathcal{R}_1 \cup \mathsf{rel}_{id_{\mathsf{Q}'}}$ in the canonical form by just $\mathcal{R}_1$ as in the following proposition. Further, if $\mathcal{R} \subseteq \mathsf{Q} \times \mathsf{Q}'$ is a canonical simulation, then we say that any set $A \subseteq \mathsf{Q} \cup \mathsf{Q}'$ is $\mathcal{R}$-closed iff it is $\mathsf{rel}_{id_{\mathsf{Q}}} \cup \mathcal{R} \cup \mathsf{rel}_{id_{\mathsf{Q}'}}$-closed.

PROPOSITION 2.10. *Given pairwise disjoint MDP's $\mathcal{M}_0 = (\mathsf{Q}_0, q_0, \delta_0, \mathsf{L}_0)$, $\mathcal{M}_1 = (\mathsf{Q}_1, q_1, \delta_1, \mathsf{L}_1)$ and $\mathcal{M}_2 = (\mathsf{Q}_2, q_2, \delta_2, \mathsf{L}_2)$, if $\mathcal{R}_{01} \subseteq \mathsf{Q}_0 \times \mathsf{Q}_1$ and $\mathcal{R}_{12} \subseteq \mathsf{Q}_1 \times \mathsf{Q}_2$ are* *canonical simulations then the relation* $\mathcal{R}_{02} = \mathcal{R}_{12} \circ \mathcal{R}_{01} \subseteq \mathsf{Q}_0 \times \mathsf{Q}_2$ *is a canonical simulation.*

2.4 PCTL-safety and PCTL-liveness.

We define a fragment of PCTL which we call the *safety fragment*. The *safety fragment* of PCTL (over a set of propositions AP) is defined in conjunction with the *liveness fragment* as follows.

$$\begin{aligned}
\psi_S &:= \mathsf{tt} \;[\!]\; \mathsf{ff} \;[\!]\; P \;[\!]\; (\neg P) \;[\!]\; (\psi_S \wedge \psi_S) \;[\!]\; (\psi_S \vee \psi_S) \;[\!]\; \mathcal{P}_{\lhd p}(\mathsf{X}\,\psi_L) \;[\!]\; \mathcal{P}_{\lhd p}(\psi_L \;\mathcal{U}\; \psi_L)\\
\psi_L &:= \mathsf{tt} \;[\!]\; \mathsf{ff} \;[\!]\; P \;[\!]\; (\neg P) \;[\!]\; (\psi_L \wedge \psi_L) \;[\!]\; (\psi_L \vee \psi_L) \;[\!]\; (\neg\mathcal{P}_{\lhd p}(\mathsf{X}\,\psi_L)) \;[\!]\; (\neg\mathcal{P}_{\lhd p}(\psi_L \;\mathcal{U}\; \psi_L))
\end{aligned}$$

where $P \in \mathsf{AP}$, $p \in [0,1]$ is a rational number and $\lhd \in \{<, \leq\}$.

Note that for any safety formula ψ_S there exists a liveness formula ψ_L such that for state q of a MDP $\mathcal{M}$, $q \Vdash_{\mathcal{M}} \psi_S$ iff $q \nVdash_{\mathcal{M}} \psi_L$. Restricting $\lhd$ to be $\leq$ in the above grammar, yields the strict liveness and weak safety fragments of PCTL. Finally recall that $\Diamond\psi$ is an abbreviation for $\mathsf{tt}\;\mathcal{U}\;\psi$.

There is a close correspondence between simulation and the liveness and safety fragments of PCTL— simulation preserves liveness and reflects safety whose proof has also been moved to an Electronic Appendix.

LEMMA 2.11. *Let* $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ *be a MDP. For any states* $q, q' \in \mathsf{Q}$, $q \preceq q'$ *implies that for every liveness formula* ψ_L, *if* $q \Vdash_{\mathcal{M}} \psi_L$ *then* $q' \Vdash_{\mathcal{M}} \psi_L$ *and that for every safety formula* ψ_S, *if* $q' \Vdash_{\mathcal{M}} \psi_S$ *then* $q \Vdash_{\mathcal{M}} \psi_S$.

Remark 2.12.

(1) The fragment presented here is syntactically different than the safety and liveness fragments presented in [Desharnais 1999b; Baier et al. 2005] for DTMCs, where the safety fragement and the liveness fragement takes the following form.

$$\begin{aligned}
\psi_S &:= \mathsf{tt} \;[\!]\; \mathsf{ff} \;[\!]\; P \;[\!]\; (\neg P) \;[\!]\; (\psi_S \wedge \psi_S) \;[\!]\; (\psi_S \vee \psi_S) \;[\!]\; \mathcal{P}_{\lhd p}(\mathsf{X}\,\psi_L) \;[\!]\; \mathcal{P}_{\lhd p}(\psi_L \;\mathcal{U}\; \psi_L)\\
\psi_L &:= \mathsf{tt} \;[\!]\; \mathsf{ff} \;[\!]\; P \;[\!]\; (\neg P) \;[\!]\; (\psi_L \wedge \psi_L) \;[\!]\; (\psi_L \vee \psi_L) \;[\!]\; \mathcal{P}_{\rhd p}(\mathsf{X}\,\psi_S) \;[\!]\; \mathcal{P}_{\rhd p}(\psi_S \;\mathcal{U}\; \psi_S)
\end{aligned}$$

where $P \in \mathsf{AP}$, $p \in [0,1]$ is a rational number and $\lhd \in \{<, \leq\}$ and $\rhd \in \{>, \geq\}$. The two presentations have the same semantics for DTMCs, but behave differently for MDPs (see Example 2.13 below). As far as we know, the safety fragment of PCTL for state-labeled MDPs has not been discussed previously in the literature.

(2) For MDPs in which the labels are on transitions and not on states, [Segala and Lynch 1994] also have a preservation result for the corresponding PCTL. However, their proof crucially relies on the fact that all the probabilistic transitions have the full measure 1 and their preservation result does not hold otherwise.

(3) Please note that, unlike the case of DTMCs [Desharnais 1999b; Baier et al. 2005], logical simulation does not characterize simulation for MDP's. One can recover the correspondence between logical simulation and simulation, if each nondeterministic choice is labeled uniquely and the logic allows one to refer to the label of transitions [Desharnais et al. 2000].

Example 2.13. This example highlights the motivation for the syntactic restriction of PCTL safety that we consider here. Consider the two MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$

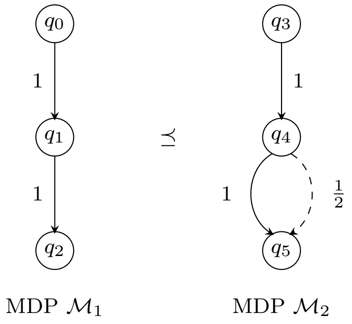

Fig. 1. MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ from Example 2.13.

shown in Figure 1. The initial state of $\mathcal{M}_1$ is q_0 and the initial state of $\mathcal{M}_2$ is q_3. The states q_2 and q_5 are labeled by proposition P. No other states are labeled by any proposition. Note that the $\mathcal{M}_1$ is simulated by $\mathcal{M}_2$. Consider the PCTL formula $\psi = \mathcal{P}_{\leq \frac{1}{2}}(\mathsf{X}(\mathcal{P}_{> \frac{1}{2}}(\mathsf{X}P)))$. Now $\mathcal{M}_1 \Vdash \psi$, but $\mathcal{M}_2 \nVdash \psi$. This is because $\mathcal{M}_2$ has a transition from the state q_4 in which the probability of transitioning to q_5 is exactly $\frac{1}{2}$. Therefore, ψ is not a safety formula (although if we were only considering DTMCs, ψ would be a safety formula; see remark above for the PCTL safety fragment for DTMCs presented in [Desharnais 1999b; Baier et al. 2005]). On the other hand, the formula $\psi_S = \mathcal{P}_{\leq \frac{1}{2}}(\mathsf{X}(\neg \mathcal{P}_{\leq \frac{1}{2}}(\mathsf{X}P)))$ is a safety formula and satisfied by both $\mathcal{M}_1$ and $\mathcal{M}_2$. Note that ψ_S and ψ are logically equivalent on DTMCs.

Example 2.14. We now give examples of safety and liveness properties in the context of PCTL. Consider a leader election protocol that elects a leader from among a collection of n (peer) processes. Such protocols are central to achieving a number goals in a distributed system, including Firewire Root Contention. The model of the protocol has probabilistic transitions because of randomization employed by the leader election algorithm, and non-determinism due the asynchrony present in a concurrent, distributed system. Let us assume that the proposition ElectedLeader labels exactly those global states where all the participants agree upon a common leader. Now one of the properties that one may want to verify about such a protocol is that a leader is *almost surely* elected. This can be written as $\mathcal{P}_{\geq 1}(\mathsf{tt}\ \mathcal{U}\ \text{ElectedLeader})$, and as can be seen it is a PCTL-liveness property as per our characterization. Consider now a stronger property, where want to ensure that a leader is elected with high probability within a certain time bound, where the deadline is either a real-time deadline or based on the number of steps taken by the protocol. Let us assume that the proposition DeadlinePassed labels exactly those global states where the time is greater than the deadline. Then our stronger correctness requirement can be written as $\mathcal{P}_{<0.05}(\mathsf{tt}\,\mathcal{U}\ (\text{DeadlinePassed} \wedge \neg \text{ElectedLeader}))$. This is a safety property by our classification.

3. COUNTEREXAMPLES

What is a counterexample? [Clarke et al. 2002] say that counterexamples must (a) serve as an "explanation" of why the (abstract) model violates the property, (b) must be rich enough to explain the violation of a large class of properties, and (c)

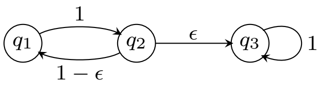

Fig. 2. DTMC with a large set of counterexample executions

must be simple and specific enough to identify bugs, and be amenable to efficient generation and analysis.

In this section, we discuss three relevant proposals for counterexamples. The first one is due to [Han and Katoen 2007a], who present a notion of counterexamples for DTMCs. This has been recently extended to MDP's by [Aljazzar and Leue 2007]. The second proposal for counterexamples was suggested in the context of non-probabilistic systems and branching time properties [Clarke et al. 2002]. Finally the third one has been recently suggested by [Chatterjee et al. 2005; Hermanns et al. 2008] for MDPs. We examine the all these proposals in order and identify why each one of them is inadequate for our purposes. We then present the definition of counterexamples that we consider in this paper.

3.1 Set of Traces as Counterexamples

The problem of defining a notion of counterexamples for probabilistic systems was first considered in [Han and Katoen 2007a]. Han and Katoen present a notion of counterexamples for DTMCs and define a counterexample to be a *finite* set of executions such that the measure of the set is greater than some threshold (they consider weak-safety formulas only). The problem to compute the smallest set of executions is intractable, and Han and Katoen present algorithms to generate such a set of executions. This definition has recently been extended for MDPs in [Aljazzar and Leue 2007].

The "set of executions as counterexample" proposal has a few drawbacks. Consider the DTMC $\mathcal{M}$ shown in Figure 2, where proposition P is true only in state q_3 and q_1 is the initial state. This DTMC violates the property $\psi_S = \mathcal{P}_{<1}(\Diamond P)$. However, there is no finite set of executions that witnesses this violation. Such properties are not considered in [Han and Katoen 2007a; Aljazzar and Leue 2007].

Even if the property being considered is a weak safety property, the number of paths serving as the counterexample can be large. For example, let $\psi_S = \mathcal{P}_{\leq 1-\delta}(\Diamond P)$. The Markov chain $\mathcal{M}$ violates property ψ_S for all values of $\delta > 0$. However, one can show that the smallest set of counterexamples is large due to the following observations.

—Any execution, starting from q_1, reaching q_3 is of the form $(q_1q_2)^kq_3$ with measure $(1-\epsilon)^{k-1}\epsilon$. Thus the measure of the set $Exec = \{(q_1q_2)^kq_3 \mid k \leq n\}$ is $1-(1-\epsilon)^n$, and the set $Exec$ has size $O(n^2)$.

—Thus the smallest set of examples that witnesses the violation of ψ_S has at least $r = \frac{\log \delta}{\log(1-\epsilon)}$ elements and the number of nodes in this set is $O(r^2)$.

r can be very large (for example, take $\epsilon = \frac{1}{2}$ and $\delta = \frac{1}{2^{2^n}}$). In such circumstances, it is unclear whether such a set of executions can serve as a comprehensible explanation for why the system violates the property ψ_S.

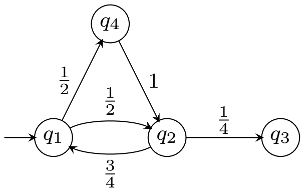

Fig. 3. DTMC $\mathcal{M}_{notree}$: No tree-like counterexamples

3.2 Tree-like Counterexamples

In the context of non-probabilistic systems and branching-time properties, [Clarke et al. 2002] suggest that counterexamples should be "tree-like". The reason to consider this proposal carefully is because probabilistic logics like PCTL are closely related to branching-time logics like CTL. *Tree-like counterexamples* for a Kripke structure $\mathcal{K}$ and property φ are defined to be a Kripke structure $\mathcal{E}$ such that (a) $\mathcal{E}$ violates the property φ, (b) $\mathcal{E}$ is simulated by $\mathcal{K}$, and (c) the underlying graph of $\mathcal{E}$ is *tree-like*, i.e., (i) every non-trivial maximal strongly connected component is a cycle, and (ii) the graph of maximal strongly connected components forms a tree. [Clarke et al. 2002] argue that this is the appropriate notion of counterexamples because tree-like counterexamples are easy to comprehend. Moreover, they show that for any Kripke structure $\mathcal{K}$ that violates an ACTL* formula φ, there is a tree-like counterexample $\mathcal{E}$.

The notion of tree-like counterexamples can be naturally extended to the case of MDPs. Formally, a *tree-like counterexample* for a MDP $\mathcal{M}$ and property ψ_S will be a (disjoint) MDP $\mathcal{E}$ such that the unlabeled underlying graph $G(\mathcal{E})$ is tree-like, $\mathcal{E}$ violates property ψ_S and is simulated by $\mathcal{M}$. However, surprisingly, unlike the case for Kripke structures and ACTL*, the family of tree-like counterexamples is not rich enough.

Example 3.1. Consider the DTMC $\mathcal{M}_{notree}$ shown in Figure 3 with initial state q_1, proposition P being true only in state q_3, and propositions P_1, P_2 and P_4 being true only is states q_1, q_2 and q_4 respectively. Consider the formula $\psi_S = \mathcal{P}_{<1}((P_1 \vee P_2 \vee P_4)\,\mathcal{U}\,P)$. Clearly, the DTMC $\mathcal{M}_{notree}$ violates ψ_S.

We will show that there is no tree-like counterexample for $\mathcal{M}_{notree}$ and formula ψ_S, defined in Example 3.1. This is done in two steps. First we prove that there is no tree-like DTMC counterexample for $\mathcal{M}_{notree}$ and ψ_S; this is the content of Proposition 3.2. Next, by using Proposition 3.2, we show in Lemma 3.3, that there is no tree-like MDP counterexample as well, which gives us our desired result.

The absence of tree-like DTMC counterexamples (Proposition 3.2) is proved as follows.

(1) First, assume that there is a tree-like DTMC $\mathcal{T}$ such that $\mathcal{T}$ is simulated by $\mathcal{M}_{notree}$ and $\mathcal{T} \not\Vdash \pi_S$. Without loss of generality, the initial state of $\mathcal{T}$ can be taken to be in the strongly connected component that is at the root of the component tree of $\mathcal{T}$.

(2) From the fact that $\mathcal{M}_{notree}$ simulates $\mathcal{T}$ and each state of $\mathcal{M}$ is labeled

by a unique proposition, we deduce that each state of $\mathcal{T}$ is simulated by exactly one state of $\mathcal{M}$. Thus the set of states of $\mathcal{T}$ can be partitioned into 4 sets $\mathsf{Q}_i = \{q \mid q$ is a state of $\mathcal{T}$ and q is simulated by $q_i\}$.

(3) Next we show that for each state $q \in \mathsf{Q}_i$, it is the case that the probability of transitioning from q to Q_j is the probability of transitioning from q_i to q_j in $\mathcal{M}_{notree}$ which allows to conclude that each $q \in \mathsf{Q}_i$ is in-fact bisimilar to q_i and not just simulated by q_i.

(4) Next, we observe that since there is no transition out of q_3, each state of Q_3 must be a leaf node of underlying unlabeled graph of $\mathcal{T}$. Consider the DTMC $\mathcal{T}_1$ obtained by deleting the states Q_3 from $\mathcal{T}$ and the DTMC $\mathcal{M}_1$ obtained by deleting the state q_3 from $\mathcal{M}_{notree}$. $\mathcal{T}_1$ is also tree-like and bisimlar to $\mathcal{M}_1$.

(5) Consider a strongly connected component $\mathcal{C}$ of $\mathcal{T}_1$ which is a leaf node of the component graph of $\mathcal{T}_1$. Now, this strongly connected component must be bisimilar to $\mathcal{M}_1$. However, the strongly connected component $\{q_1, q_2, q_4\}$ of $\mathcal{M}_1$ has two cycles $q_1 \to q_4 \to q_2 \to q_1$ and $q_1 \to q_2 \to q_1$. This implies that $\mathcal{C}$ must also contain two different cycles. Hence $\mathcal{C}$ cannot be a cycle which violates the tree-like structure of $\mathcal{T}_1$.

Using this observation, the absence of tree-like MDP counterexample (Lemma 3.3), is proved based on the following idea. If there is an MDP $\mathcal{M}$ such that $\mathcal{M}$ violates ψ_S and $\mathcal{M} \preceq \mathcal{M}_{notree}$, then there must be a memoryless scheduler $\mathcal{S}$ such that the DTMC $\mathcal{M}^{\mathcal{S}}$ violates ψ_S and is also simulated $\mathcal{M}_{notree}$. The result now follows from the observation that $\mathcal{M}^{\mathcal{S}}$ is tree-like if $\mathcal{M}$ is tree-like. We are now ready to present the formal details of these statements and their proofs.

Proposition 3.2. *Consider the DTMC $\mathcal{M}_{notree}$ and safety formula ψ_S defined in Example 3.1. If $\mathcal{T} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ is a DTMC (disjoint from $\mathcal{M}_{notree}$) such that $\mathcal{T} \preceq \mathcal{M}_{notree}$ and $\mathcal{T} \not\Vdash \psi_S$ then $\mathcal{T}$ is not tree-like.*

Proof. Assume, by way of contradiction, that $\mathcal{T}$ is tree-like. Let $\mathsf{Q}_0 = \{q_1, q_2, q_3, q_4\}$ and for each $1 \leq i \leq 4$, let μ_{q_i} denote the transition out of q_i in $\mathcal{M}_{notree}$.

For each $q \in \mathsf{Q}$, let μ_q denote the transition out of q in $\mathcal{T}$. Let $\mathcal{R} \subseteq \mathsf{Q} \times \mathsf{Q}_0$ be a canonical simulation that witnesses the fact that $\mathcal{T} \preceq \mathcal{M}_{notree}$. We have by definition, $q_{\mathcal{I}} \,\mathcal{R}\, q_1$. Please note that since $\mathcal{T}$ violates ψ_S the measure of all paths of $\mathcal{T}$ starting with $q_{\mathcal{I}}$ and satisfying $(P_1 \vee P_2 \vee P_4)\, \mathcal{U}\, P$ is 1.

As $\mathcal{T}$ is tree-like, any non-trivial strongly connected components of $G(\mathcal{T})$ is a cycle and G, the graph of the strongly connected components (trivial or non-trivial) of $G(\mathcal{T})$ form a tree. Without loss of generality, we can assume that the strongly connected component that forms the root of G contains $q_{\mathcal{I}}$ (otherwise we can just consider the DTMC restricted to the states reachable from $q_{\mathcal{I}}$).

Hence, we have that every state in Q is reachable from $q_{\mathcal{I}}$ with non-zero probability. From this and the fact $\mathcal{R}$ is a canonical simulation, we can show that for any state $q \in \mathsf{Q}$ there is a $q' \in \mathsf{Q}_0$ such that $q \,\mathcal{R}\, q'$. Also since each state in Q_0 is labeled by a unique proposition, it follows that for each $q \in \mathsf{Q}$ there is a unique $q' \in \mathsf{Q}_0$ such that $q \,\mathcal{R}\, q'$ (in other words, $\mathcal{R}$ is total and functional).

Now, for each $1 \leq i \leq 4$, let $\mathsf{Q}_i \subseteq \mathsf{Q}$ be the set $\{q \in \mathsf{Q} \mid q \,\mathcal{R}\, q_i.\}$ By the above observations, we have that Q_i's are pairwise disjoint; $\mathsf{Q} = \mathsf{Q}_1 \cup \mathsf{Q}_2 \cup \mathsf{Q}_3 \cup \mathsf{Q}_4$; and for each $1 \leq i \leq 4$, $\mathsf{Q}_i \cup \{q_i\}$ is a $\mathcal{R}$-closed set. Since $\mathcal{R}$ is a canonical simulation, whenever $q\mathcal{R}q'$, we have $\mu_q(Q_i) \leq \mu_{q'}(q_i)$, for each $1 \leq i \leq 4$. Moreover, we can, in fact, prove the following stronger claim.

Claim: $\mu_q(\mathsf{Q}_i) = \mu_{q'}(q_i)$ for each $q\mathcal{R}q'$ and $1 \leq i \leq 4$.

Proof of the claim: Consider some q, q' such that $q\ \mathcal{R}\ q'$. We proceed by contradiction. Assume that there is some i such $\mu_q(\mathsf{Q}_i) < \mu_{q'}(q_i)$. Please note that in this case $q' \neq q_3$ (as $\mu_{q_3}(q_i) = 0, \forall 1 \leq i \leq 4$).

There are several possible cases (depending q' and q_i). We just discuss the case when q' is q_4 and i is 2. The other cases are similar. For this case we have that $\mu_q(\mathsf{Q}_2) < 1$. Also note that for $j \neq 2$, $\mu_q(\mathsf{Q}_j) \leq \mu_{q_4}(q_j) = 0$. Hence, $\mu_q(\mathsf{Q}) < 1$. Now, pick two new states q_{new_2} and q_{new_3} not occurring in $\mathsf{Q} \cup \mathsf{Q}_0$. Construct a new tree-like DTMC $\mathcal{T}'$ extending $\mathcal{T}$ as follows. The states of $\mathcal{T}'$ are $\mathsf{Q} \cup \{q_{new_2}, q_{new_3}\}$. Only proposition P_2 is true in q_{new_2} and only proposition P is true in q_{new_3}. The labeling function for other states remains the same. We extend the probabilistic transition μ_q by letting $\mu_q(q_{new_2}) = 1 - \mu_q(\mathsf{Q})$ and $\mu_q(q_{new_3}) = 0$ (transition probabilities to other states do not get affected). The state q_{new_2} has a probabilistic transition $\mu_{q_{new_2}}$ such that $\mu_{q_{new_2}}(q_{new_3}) = \frac{1}{4}$ and $\mu_{q_{new_2}}(\bar{q}) = 0$ for any $\bar{q} \neq q_{new_3}$. The transition probability from q_{new_3} to any state is 0. For all other states the transitions remain the same.

Now, please note that there is a path π from $q_{\mathcal{I}}$ to q (in $\mathcal{T}$ and hence in $\mathcal{T}'$ also) with non-zero "measure" such that $P_1 \vee P_2 \vee P_4$ is true at each point in this path. Furthermore, at each point in this path, P is false. Consider the path π' in $\mathcal{T}'$ obtained by extending π by q_{new_2} followed by q_{new_3}. Now, by construction π' satisfies $(P_1 \vee P_2 \vee P_4)\,\mathcal{U}\,P$ and the "measure" of this path > 0 (as $1 - \mu_q(\mathsf{Q}) > 0$). Now the path π' is not present in $\mathcal{T}$ and hence the measure of all paths in $\mathcal{T}'$ that satisfy $(P_1 \vee P_2 \vee P_4)\,\mathcal{U}\,P$ is strictly greater than the measure of all paths in $\mathcal{T}$ that satisfy $(P_1 \vee P_2 \vee P_4)\,\mathcal{U}\,P$. The latter number is 1 and thus the measure of all paths in $\mathcal{T}'$ that satisfy $(P_1 \vee P_2 \vee P_4)\,\mathcal{U}\,P > 1$. Impossible! $\square$ (End proof of the claim)

We proceed with the proof of the main proposition. Let $\mathcal{R}_1 \subseteq (\mathsf{Q} \cup \mathsf{Q}_0) \times (\mathsf{Q} \cup \mathsf{Q}_0)$ be the reflexive, symmetric and transitive closure of $\mathcal{R}$ (in other words, the smallest equivalence that contains $\mathcal{R}$). It is easy to see that the equivalence classes of $\mathcal{R}_1$ are exactly $\mathsf{Q}_i \cup \{q_i\}, 1 \leq i \leq 4$. From this fact and our claim above, we can show that the $\mathcal{R}_1$ is a bisimulation.

Observe now that each element of $\mathsf{Q}_3 \subseteq \mathsf{Q}$ must be a leaf node of G, the graph of the strongly connected components of $G(\mathcal{T})$. Using this, one can easily show that if $\mathcal{T}_1$ is the DTMC obtained from $\mathcal{T}$ by restricting the state space to $\mathsf{Q}_1 \cup \mathsf{Q}_2 \cup \mathsf{Q}_4$, then $\mathcal{T}_1$ is tree-like. Let $\mathcal{M}_1$ be the DTMC obtained from $\mathcal{M}_{notree}$ by restricting the state space to $\mathsf{Q}_0 \setminus \{q_3\}$ and $\tilde{\mathsf{Q}} = (\mathsf{Q}_1 \cup \mathsf{Q}_2 \cup \mathsf{Q}_4) \cup (\mathsf{Q}_0 \setminus \{q_3\})$. It is easy to see that the the equivalence relation $\mathcal{R}_2 = \mathcal{R}_1 \cap (\tilde{\mathsf{Q}} \times \tilde{\mathsf{Q}})$ is also a bisimulation.

Now, let G_1 be the graph of strongly connected components of $G(\mathcal{T}_1)$. Now, fix a strongly connected component of $G(\mathcal{T}_1)$, say C, that is a leaf node of G_1. Fix a state q which is a node of C. Since G_1 is tree-like and C is a leaf node, it is easy to see that $\mathsf{post}(\mu_q, q)$ can contain at most 1 element. Also, we have that $q \in \mathsf{Q}_1 \cup \mathsf{Q}_2 \cup \mathsf{Q}_4$. Now if $q \in \mathsf{Q}_1$, we have that q (as a state of $\mathcal{T}_1$) is bisimilar to q_1 (as a state of $\mathcal{M}_1$). However, this implies that $\mathsf{post}(\mu_q, q)$ must be at least 2 as q_1 has a non-zero probability of transitioning to 2 states labeled by different propositions. Hence $q \notin \mathsf{Q}_1$. If $q \in \mathsf{Q}_2$, then please note that $\mathsf{post}(\mu_q, q)$ must contain an element in Q_1 which should also be in C. By the above observation this is not possible. Hence $q \notin \mathsf{Q}_2$. Similarly, we can show that $q \notin \mathsf{Q}_4$. Hence $q \notin \mathsf{Q}_1 \cup \mathsf{Q}_2 \cup \mathsf{Q}_4$. A contradiction. $\square$

We are ready to show that $\mathcal{M}_{notree}$ has no tree-like counterexamples.

LEMMA 3.3. *Consider the DTMC $\mathcal{M}_{notree}$ and formula ψ_S defined in Example 3.1. There is no tree-like counterexample witnessing the fact that $\mathcal{M}_{notree}$ violates ψ_S.*

PROOF. First, since $((P_1 \vee P_2 \vee P_3) \,\mathcal{U}\, P)$ is a simple reachability formula, if there is a MDP $\mathcal{E} \preceq \mathcal{M}_{notree}$ which violates ψ_S, then there is a memoryless scheduler S such that $\mathcal{E}^S$ violates the property [Bianco and de Alfaro 1995]. Now note that if we were to just consider the states of $\mathcal{E}^S$ reachable from the initial state of then $\mathcal{E}^S$ is also tree-like. In other words, there is a tree-like DTMC that is simulated by $\mathcal{M}_{notree}$ and which violates the property ψ_S. The result now follows from Proposition 3.2. $\square$

Tree-like graph structures are not rich enough for PCTL-safety. However, it can be shown that if we restrict our attention to weak-safety formulas, then we have tree counterexamples. However, such trees can be very big as they depend on the actual transition probabilities.

THEOREM 3.4. *If ψ_{WS} is a weak safety formula and $\mathcal{M} \nVdash \psi_{WS}$, then there is a $\mathcal{M}'$ such that $G(\mathcal{M}')$ is a tree, $\mathcal{M}' \preceq \mathcal{M}$, and $\mathcal{M}' \nVdash \psi_{WS}$.*

PROOF. The result follows from the following two observations.

- —If the underlying graph $G(\mathcal{M}_1)$ of a MDP $\mathcal{M}_1$ is acyclic then there is a MDP $\mathcal{M}_2$ such that $G(\mathcal{M}_2)$ is a tree and $\mathcal{M}_1 \approx \mathcal{M}_2$.
- —For any strict liveness formula ψ_{SL}, and a state $q \in \mathcal{M}$ if $q \Vdash_{\mathcal{M}} \psi_{SL}$ then there is a k such that $\mathcal{M}^q_k \Vdash \psi_{SL}$ where $\mathcal{M}^q_k$ is the k-th unrolling of $\mathcal{M}$ rooted at q.

The latter observation can be proved by a straightforward induction on the structure of strict liveness formulas. We consider the case when ψ_{SL} is $(\neg\mathcal{P}_{\leq p}(\psi_{SL_1} \,\mathcal{U}\, \psi_{SL_2}))$.

Now if $q \Vdash \neg\mathcal{P}_{\leq p}(\psi_{SL_1} \,\mathcal{U}\, \psi_{SL_2})$ then note that there is a memoryless scheduler $\mathcal{S}$ [Bianco and de Alfaro 1995] such that $q \Vdash_{\mathcal{M}^{\mathcal{S}}} \neg\mathcal{P}_{\leq p}(\psi_{SL_1} \,\mathcal{U}\, \psi_{SL_2})$. This implies that there is a finite set of finite paths of $\mathcal{M}^{\mathcal{S}}$ starting from q such that each path satisfies $\psi_{SL_1} \,\mathcal{U}\, \psi_{SL_2}$ and the measure of these paths $> p$ [Han and Katoen 2007a]. Now, these paths can be arranged in a tree T nodes of which are labeled by the corresponding state of $\mathcal{M}$. If the state q' labels a leaf node then we have $q' \Vdash_{\mathcal{M}} \psi_{SL_2}$; otherwise $q' \Vdash_{\mathcal{M}} \psi_{SL_1}$.

For any state $q' \in \mathsf{Q}$ labeling a node in T define $\mathsf{maxdepth}(q') = \max\{\mathsf{depth}(t) \mid t \text{ is a node of } \mathsf{T} \text{ and } t \text{ is labeled by } q'\}$. If q' labels a node of T such that $q' \Vdash_{\mathcal{M}} \psi_{SL_1}$ but $q' \nVdash_{\mathcal{M}} \psi_{SL_2}$ then fix $k_{q'}$ such that $\mathcal{M}^{q'}_{k_{q'}} \Vdash \psi_{SL_1}$ ($k_{q'}$ exists by induction hypothesis). If q' labels a node of T such that $q' \Vdash_{\mathcal{M}} \psi_{SL_2}$ but $q' \nVdash_{\mathcal{M}} \psi_{SL_1}$ then fix $k_{q'}$ such that $\mathcal{M}^{q'}_{k_{q'}} \Vdash \psi_{SL_2}$. If q' labels a node of T such that $q' \Vdash_{\mathcal{M}} \psi_{SL_1}$ and $q' \Vdash_{\mathcal{M}} \psi_{SL_1}$ then fix $k_{q'}$ such that $\mathcal{M}^{q'}_{k_{q'}} \Vdash \psi_{SL_1}$ and $\mathcal{M}^{q'}_{k_{q'}} \Vdash \psi_{SL_2}$. Now, let $k = \max\{\mathsf{maxdepth}(q') + k_{q'} \mid q' \text{ labels a node of } \mathsf{T}.\}$ It can now be shown easily that $\mathcal{M}^q_k \Vdash \psi_{SL}$. $\square$

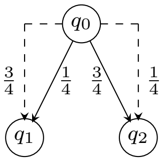

Fig. 4. MDP $\mathcal{M}$ used in the proof of Proposition 3.5. Transition μ_1 is shown as dashed edges, and transition μ_2 is shown as solid edges.

3.3 DTMCs as counterexamples

We now consider the third and final proposal for a notion of counterexamples that is relevant for MDPs. [Chatterjee et al. 2005] use the idea of abstraction-refinement to synthesize winning strategies for stochastic 2-player games. They abstract the game graph, construct winning strategies for the abstracted game, and check the validity of those strategies for the original game. They observe that for discounted reward objectives and average reward objectives, the winning strategies are *memoryless*, and so "counterexamples" can be thought of as finite-state models without nondeterminism (which is resolved by the strategies constructed).

This idea also used in [Hermanns et al. 2008]. They observe that for weak-safety formulas of the form $\mathcal{P}_{\leq p}(\psi_1 \, \mathcal{U} \, \psi_2)$ where ψ_1 and ψ_2 are propositions (or boolean combinations of propositions), if a MDP $\mathcal{M}$ violates the property then there is a memoryless scheduler $\mathcal{S}$ such that the DTMC $\mathcal{M}^{\mathcal{S}}$ violates $\mathcal{P}_{\leq p}(\psi_1 \, \mathcal{U} \, \psi_2)$ (see [Bianco and de Alfaro 1995]). Therefore, they take the pair $(\mathcal{S}, \mathcal{M}^{\mathcal{S}})$ to be the counterexample.

Motivated by these proposals and our evidence of the inadequacies of sets of executions and tree-like systems as counterexamples, we ask whether DTMCs (or rather purely probabilistic models) could serve as an appropriate notion for counterexamples of MDPs. We answer this question in the negative. The reason why DTMCs fail to suffice is the following: since each probabilistic operator can refer to different resolutions of the nondeterminism in the MDP, if a formula contains more than one probabilistic operator then in a counterexample more than one resolution of the nondeterminism may be required.

Proposition 3.5. *There is a MDP $\mathcal{M}$ and a safety formula ψ_S such that $\mathcal{M} \not\Vdash \psi_S$ but there is no DTMC $\mathcal{M}'$ that violates ψ_S and is simulated by $\mathcal{M}$.*

Proof. The MDP $\mathcal{M}$ will have three states q_0, q_1, q_2 with q_0 as the initial state, and is shown in Figure 4. The transition probability from q_1 and q_2 to any other state is 0. There will be two transitions out of q_0, μ_1 and μ_2, where $\mu_1(q_0) = 0, \mu_1(q_1) = \frac{3}{4}, \mu_1(q_2) = \frac{1}{4}$ and $\mu_2(q_0) = 0, \mu_2(q_1) = \frac{1}{4}, \mu_2(q_2) = \frac{3}{4}$. For the labeling function, we pick two distinct propositions P_1 and P_2 and let $\mathsf{L}(q_0) = \emptyset$, $\mathsf{L}(q_1) = \{P_1\}$ and $\mathsf{L}(q_2) = \{P_2\}$. Consider the safety formula $\psi_S = \mathcal{P}_{<\frac{3}{4}}(\mathsf{X}(P_1 \wedge \neg P_2)) \vee \mathcal{P}_{<\frac{3}{4}}(\mathsf{X}(\neg P_1 \wedge P_2))$. Now $\mathcal{M}$ violates ψ_S.

Suppose that $\mathcal{M}' = (\mathsf{Q}', q_{\mathcal{I}}, \delta', \mathsf{L}')$ is a counterexample for $\mathcal{M}$ and ψ_S. Then we must have $q_{\mathcal{I}} \Vdash \neg\mathcal{P}_{<\frac{3}{4}}(\mathsf{X}(P_1 \wedge \neg P_2)) \wedge \neg\mathcal{P}_{<\frac{3}{4}}(\mathsf{X}(\neg P_1 \wedge P_2))$. Now, if $\mathcal{M}'$ is a DTMC, $\delta'(q_{\mathcal{I}})$ must contain exactly one element $\mu_{q_{\mathcal{I}}}$. Also since $q_{\mathcal{I}} \Vdash \neg\mathcal{P}_{<\frac{3}{4}}(\mathsf{X}(P_1 \wedge$

$\neg P_2$)) there must be a state q'_1 such that $P_1 \in \mathsf{L}'(q'_1), P_2 \notin \mathsf{L}'(q'_1)$ and $\mu_{q_\mathcal{I}}(q'_1) \geq \frac{3}{4}$. Similarly, there must also be a state q'_2 such that $P_2 \in \mathsf{L}'(q'_2), P_1 \notin \mathsf{L}'(q'_2)$ and $\mu_{q_\mathcal{I}}(q'_2) \geq \frac{3}{4}$. Now, clearly $q'_1 \neq q'_2$ (as they do not satisfy the same set of propositions). However, we have that $\mu_{q_\mathcal{I}}(Q') \geq \frac{3}{4} + \frac{3}{4} > 1$. A contradiction. □

Remark 3.6. Please note that in the proof of Proposition 3.5, we have used a safety formula which is a disjunction of safety formulas. We could also construct examples in which the safety formula consists of nested reachability formulas, *ie*, a formula of the form $\mathcal{P}_{\leq p}(\phi \, \mathcal{U} \, \phi')$ where ϕ and/or ϕ' are not boolean combinations of propositional symbols. For example, consider the MDP $\mathcal{M}$ in the proof of Proposition 3.5 and let ψ be safety formula $\mathcal{P}_{<\frac{3}{4}}((\neg \mathcal{P}_{<\frac{3}{4}}(P_2)) \, \mathcal{U} \, P_1)$. Then $\mathcal{M} \not\Vdash \psi$, and it can be shown along the lines of the proof of Proposition 3.5 that there is no DTMC counterexample for $\mathcal{M}$ and ψ.

3.4 Our Proposal: MDPs as Counterexamples

Counterexamples for MDPs with respect to safe PCTL formulas cannot have any special structure. We showed that there are examples of MDPs and properties that do not admit any tree-like counterexample (Section 3.2). We also showed that there are examples that do not admit collections of executions, or general DTMCs (i.e., models without nondeterminism) as counterexamples (Sections 3.1, 3.3). Therefore in our definition, counterexamples will simply be general MDPs. We will further require that counterexamples carry a "proof" that they are counterexamples in terms of a canonical simulation which witnesses the fact the given MDP simulates the counterexample. Although we do not really need to have this simulation in the definition for discussing counterexamples (one can always compute a simulation), this slight extension will prove handy while discussing counterexample guided refinement. Formally,

Definition 3.7. For a MDP $\mathcal{M} = (\mathsf{Q}, q_\mathcal{I}, \delta, \mathsf{L})$ and safety property ψ_S such that $\mathcal{M} \not\Vdash \psi_S$, a counterexample is pair $(\mathcal{E}, \mathcal{R})$ such that $\mathcal{E} = \{\mathsf{Q}_\mathcal{E}, q_\mathcal{E}, \delta_\mathcal{E}, \mathsf{L}_\mathcal{E}\}$ is a MDP disjoint from $\mathcal{M}$, $\mathcal{E} \not\Vdash \psi_S$ and $\mathcal{R} \subseteq \mathsf{Q}_\mathcal{E} \times \mathsf{Q}$ is a canonical simulation.

For the counterexample to be useful we will require that it be "small". Our definition of what it means for a counterexample to be "small" will be driven by another requirement outlined by in [Clarke et al. 2002], namely, that it should efficiently generatable. These issues will be considered next.

3.5 Computing Counterexamples

Since we want the counterexample to be small, one possibility would be to consider the smallest counterexample. The size of a counterexample $(\mathcal{E}, \mathcal{R})$ can be taken to be the sum of sizes of the underlying labeled graph of $\mathcal{E}$, the size of the numbers used as probabilities in $\mathcal{E}$ and the cardinality of the set $\mathcal{R}$; the smallest counterexample is then the one that has the smallest size. However, it turns out that computing the smallest counterexample is a computationally hard problem. This is the formal content of our next result. For this section, we assume the standard definition of the size of a PCTL formula.

Notation 3.8. Given a safety formula, ψ_S, we denote the size of ψ_S by $|\psi_S|$.

We now formally define the size of the counterexample.

Definition 3.9. Let $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ be a MDP. The size of $\mathcal{M}$, denoted as $|\mathcal{M}|$, is the sum of the size (vertices+edges) of the labeled underlying graph $G_\ell(\mathcal{M})$ and the sum $\sum_{q\in\mathsf{Q},q'\in\mathsf{Q},\mu\in\delta(q),\mu(q')>0} size(\mu(q'))$ The size of a counterexample $(\mathcal{E}, \mathcal{R})$, denoted as $|(\mathcal{E}, \mathcal{R})|$, is the sum of the size of $\mathcal{E}$ and the cardinality (number of elements) of the relation $\mathcal{R}$.

Please note that any MDP $\mathcal{M}$ of size n has a counterexample of size $\leq 2n$ (just take an isomorphic copy of $\mathcal{M}$ as the counterexample MDP and take the obvious “injection” as the canonical simulation relation). the next theorem shows that the problem of finding the smallest counter-example is NP-complete. The proof of the theorem has been moved to an Electronic Appendix so as to not disturb the flow of the paper.

THEOREM 3.10. *Given a MDP $\mathcal{M}$, a safety formula ψ_S such that $\mathcal{M} \not\Vdash \psi_S$, and a number $k \leq 2|\mathcal{M}|$, deciding whether there is a counterexample $(\mathcal{E}, \mathcal{R})$ of size $\leq k$ is* NP*-complete.*

Not only is the problem of finding the smallest counterexample NP-hard, it also unlikely to be efficiently approximable. This is the content of the next theorem whose proof has also been moved to an Electronic Appendix.

THEOREM 3.11. *Given a MDP $\mathcal{M}$, a safety formula ψ_S and $n = |\mathcal{M}|+|\psi_S|$ such that $\mathcal{M} \not\Vdash \psi_S$. The smallest counterexample for $\mathcal{M}$ and ψ_S cannot be approximated in polynomial time to within $O(2^{\log^{1-\epsilon} n})$ unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{poly\log(n)})$.*

Remark 3.12. A few points about our hardness and inapproximability results are in order.

(1) Please note that we did not take the size of the labeling function into account. One can easily modify the proofs to take this into account.

(2) The same reduction also shows a lower bound for the safety fragment of ACTL^* properties as the reduction does not rely on any important features of quantitative properties.

Since finding the smallest counterexamples is computationally hard, we consider the problem of finding *minimal counterexamples*. Intuitively, a minimal counterexample has the property that removing any edge from the labeled underlying graph of the counterexample, results in a MDP that is not a counterexample. In order to be able to define this formally, we need the notion of when one MDP is contained in the other.

Definition 3.13. We say that a MDP $\mathcal{M}' = (\mathsf{Q}', q_I, \delta', L')$ is contained in an MDP $\mathcal{M} = (\mathsf{Q}, q_I, \delta, L)$ if $\mathsf{Q}' \subseteq \mathsf{Q}$, $L'(q') = L(q')$ for all $q' \in \mathsf{Q}'$, and there is a 1-to-1 function $f : \delta' \to \delta$ with the following property: For each $q', q'' \in \mathsf{Q}'$ and $\mu' \in \delta'(q')$, $f(\mu') \in \delta(q')$, either $\mu'(q'') = f(\mu')(q'')$ or $\mu'(q'') = 0$. We denote this by $\mathcal{M}' \subseteq \mathcal{M}$.

Observe that if $\mathcal{M}' \subseteq \mathcal{M}$ then $\mathcal{M}' \preceq \mathcal{M}$. We present the definition of minimal counterexamples obtained by lexicographic ordering on pairs $(\mathcal{E}, \mathcal{R})$.

Definition 3.14. For a MDP $\mathcal{M}$ and a safety property ψ_S, $(\mathcal{E}, \mathcal{R})$ is a minimal counterexample iff

```
Initially $M_{curr} = \bar{\mathcal{M}}$
For each edge $(\bar{q}, \bar{q_1}) \in E_i$ in $G_\ell(M_{curr})$
    Let $M'$ be the MDP obtained from $M_{curr}$ by setting $\mu(\bar{q_1}) = 0$, where $\mu$ is the
        $i$th choice out of $\bar{q}$, in $M_{curr}$
    If $M' \not\Vdash \psi_S$ then $M_{curr} = M'$
od ← End of For loop
Let $\mathcal{E}$ be the MDP obtained from $M_{curr}$ by removing the set of states from $M_{curr}$
    which are not reachable in the underlying unlabeled graph of $M_{curr}$
If $\mathsf{Q}_\mathcal{E}$ is the set of states of $\mathcal{E}$, then let $\mathsf{rel_{inj}} = \{(\bar{q}, q) \mid \bar{q} \in \mathsf{Q}_\mathcal{E}\}$
return($\mathcal{E}, \mathsf{rel_{inj}}$)
```

Fig. 5. Algorithm for computing the minimal counterexample

—$(\mathcal{E}, \mathcal{R})$ is a counterexample for $\mathcal{M}$ and ψ_S and
—If $(\mathcal{E}_1, \mathcal{R}_1)$ is also a counterexample for $\mathcal{M}$ and ψ_S, then
  —$\mathcal{E}_1 \subseteq \mathcal{E}$ implies that $\mathcal{E}_1 = \mathcal{E}$; and
  —if $\mathcal{E}_1 = \mathcal{E}$ then $\mathcal{R}_1 \subseteq \mathcal{R}_2$ implies that $\mathcal{R}_1 = \mathcal{R}_2$.

Though finding the smallest counterexample is NP-complete and is unlikely to be efficiently approximable, there is a very simple polynomial time algorithm to compute a minimal counterexample. In fact the counterexample computed by our algorithm is going to be contained in the original MDP (upto "renaming" of states). Before we proceed, we fix some notation for the rest of the paper.

Notation 3.15. Given a MDP $\mathcal{M} = (\mathsf{Q}, q_\mathcal{I}, \delta, \mathsf{L})$, for each $q \in \mathsf{Q}$ fix a unique element $\bar{q}$ not occurring in Q. Define an *isomorphic* MDP $\bar{\mathcal{M}} = (\bar{\mathsf{Q}}, \bar{q_\mathcal{I}}, \bar{\delta}, \bar{\mathsf{L}})$ as follows.

—$\bar{\mathsf{Q}} = \{\bar{q} \mid q \in \mathsf{Q}\}$.
—$\bar{\delta}(\bar{q}) = \{\bar{\mu} \mid \mu \in \delta(q)\}$ where
  —for each $\bar{q} \in \bar{Q}$, $\bar{\mu}(\bar{q}) = \mu(q)$.
—$\bar{\mathsf{L}}(\bar{q}) = \mathsf{L}(q)$.

We are ready to give the counterexample generation algorithm. The algorithm shown in Figure 5 clearly computes a minimal counterexample contained in the original MDP upto "renaming" of states (note that the minimality of $\mathsf{rel_{inj}}$ is a direct consequence of the fact that every state in $\mathcal{E}$ is reachable from initial state and hence must be simulated by some state in $\mathcal{M}$). Its running time is polynomial because model checking problem for MDPs is in P [Bianco and de Alfaro 1995].

Theorem 3.16. *Given a MDP $\mathcal{M}$ and a safety formula ψ_S such that $\mathcal{M} \not\Vdash \psi_S$, the algorithm in Figure 5 computes a minimal counterexample and runs in time polynomial in the size of $\mathcal{M}$ and ψ_S.*

Please note that for safety properties of the form $\mathcal{P}_{\leq p}(\psi_S \mathcal{U} \psi'_S)$ and $\mathcal{P}_{<p}(\psi_S \mathcal{U} \psi'_S)$ where ψ_S, ψ'_S are boolean combinations of propositions, if $(\mathcal{E}, \mathsf{rel_{inj}})$ is the counterexample generated by Figure 5 then $\mathcal{E}$ must be a DTMC. This is because if $\mathcal{E}$ violates such a property then there is a memoryless scheduler $\mathcal{S}$ such that $\mathcal{E}^\mathcal{S}$ violates the same property (see [Bianco and de Alfaro 1995]). For such properties;

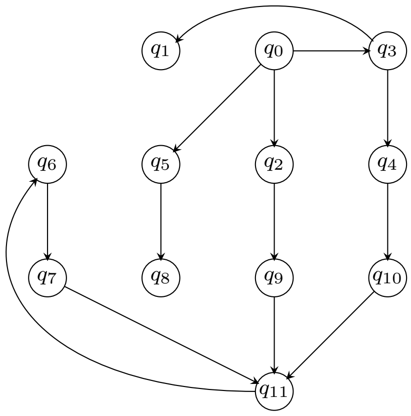

Fig. 6. Kripke structure $\mathcal{K}_{\mathsf{ex}}$

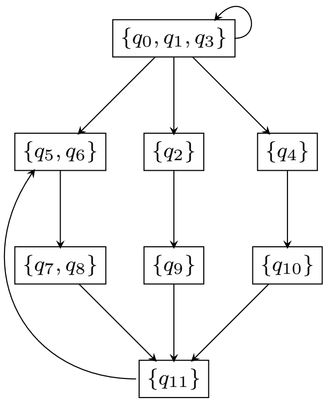

Fig. 7. Its abstraction $\mathcal{K}_{\mathsf{ab}}$

model-checking algorithm also computes the memoryless scheduler witnessing the violation. Thus for such properties, one could initialize M_{curr} to be $\bar{\mathcal{M}}^{\mathcal{S}_1}$ where $\mathcal{S}_1$ is the memoryless scheduler generated when $\bar{\mathcal{M}}$ is model-checked for violation of the given safety property.

There are a couple of important research questions to be investigated in connection with making the algorithm in Figure 5 scalable. First, the algorithm makes multiple calls to the model checker which can be expensive. It is therefore imperative that techniques to reuse the results of previous model checking runs be leveraged. Second, the counterexample returned by the algorithm, clearly depends on the order in which edges of $G_\ell(\bar{\mathcal{M}})$ are considered. Discovering heuristics for this ordering, based on the property and $\mathcal{M}$, is an important research question.

4. ABSTRACTIONS

Usually, in counterexample guided abstraction refinement framework, the abstract model is defined with the help of an equivalence relation on the states of the system [Clarke et al. 2000]. Informally, the construction for non-probabilistic systems proceeds as follows. Given a Kripke structure $\mathcal{K} = (\mathsf{Q}, q_{\mathcal{I}}, \rightarrow, \mathsf{L})$ and equivalence relation $\equiv$ on Q such that $\mathsf{L}(q) = \mathsf{L}(q')$ for $q \equiv q'$; the abstract Kripke structure for $\mathcal{K}$ and $\equiv$ is defined as the Kripke structure $\mathcal{K}_{\mathcal{A}} = (\mathsf{Q}_{\mathcal{A}}, q_{\mathcal{A}}, \rightarrow_{\mathcal{A}}, \mathsf{L}_{\mathcal{A}})$ where

—$\mathsf{Q}_{\mathcal{A}} = \{[q]_\equiv \mid q \in \mathsf{Q}\}$ is the set of equivalence classes under $\equiv$,

—$q_{\mathcal{A}} = [q_{\mathcal{I}}]_\equiv$,

—$[q]_\equiv \rightarrow_{\mathcal{A}} [q']_\equiv$ if there is some $q_1 \in [q]_\equiv$ and $q_1' \in [q']_\equiv$ such that $q_1 \rightarrow q_1'$, and

—$\mathsf{L}_{\mathcal{A}}([q]_\equiv) = \mathsf{L}(q)$.

Example 4.1. Consider the Kripke $\mathcal{K}_{\mathsf{ex}}$ structure given in Figure 6 where q_0 is the initial state and the state q_{11} is labeled by proposition P (no other state is labeled by any proposition). Consider the equivalence relation $\equiv$ which partitions the set $\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$ into the equivalence classes $\{q_0, q_1, q_3\}$, $\{q_2\}$, $\{q_4\}$, $\{q_5, q_6\}$, $\{q_7, q_8\}$, $\{q_9\}$, $\{q_{10}\}$ and q_{11}. Then the abstract Kripke structure, $\mathcal{K}_{\mathsf{ab}}$ for $\mathcal{K}_{\mathsf{ex}}$ and $\equiv$ is given by the Kripke structure in Figure 7. Here $\{q_0, q_1, q_3\}$ is the initial state and $\{q_{11}\}$ is labeled by proposition P.

This construction is generalized for MDP's in [Jonsson and Larsen 1991; Huth 2005; D'Argenio et al. 2001]. To describe this generalized construction formally, we first need to lift distributions on a set with an equivalence relation $\equiv$ to a distribution on the equivalence classes of $\equiv$ [3].

Definition 4.2. Given $\mu \in \mathrm{Prob}_{\leq 1}(\mathsf{Q})$ and an equivalence $\equiv$ on Q, the lifting of μ (denoted by $[\mu]_\equiv$) to the set of equivalence classes of Q under $\equiv$ is defined as $[\mu]_\equiv([q]_\equiv) = \mu(\{q' \in \mathsf{Q} \mid q' \equiv q\})$.

For a MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$, we will say a binary relation $\equiv$ is an *equivalence relation compatible with* $\mathcal{M}$, if $\equiv$ is an equivalence relation on Q such that $\mathsf{L}(q) = \mathsf{L}(q')$ for all $q \equiv q'$. The abstract models used in our framework are then formally defined as follows.

Definition 4.3. Given a set of propositions AP, let $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ be an AP labeled MDP. Let $\equiv$ be an equivalence relation compatible with $\mathcal{M}$. The *abstract MDP for* $\mathcal{M}$ *with respect to the equivalence relation* $\equiv$ is a MDP $\mathcal{M}_\equiv = (\mathsf{Q}_\equiv, q_\equiv, \delta_\equiv, \mathsf{L}_\equiv)$ where

(1) $\mathsf{Q}_\equiv = \{[q]_\equiv \mid q \in \mathsf{Q}\}$.
(2) $q_\equiv = [q_{\mathcal{I}}]_\equiv$.
(3) $\delta_\equiv([q]_\equiv) = \{\mu \mid \exists q' \in [q]_\equiv \text{ and } \mu_1 \in \delta(q') \text{ such that } \mu = [\mu_1]_\equiv\}$.
(4) $\mathsf{L}_\equiv([q]_\equiv) = \mathsf{L}(q)$.

The elements of Q will henceforth be called *concrete states* and the elements $\mathsf{Q}_\equiv$ will henceforth be called *abstract states.* The relation $\mathsf{rel}^\alpha_\equiv \subseteq \mathsf{Q} \times \mathsf{Q}_\equiv$ defined as $\mathsf{rel}^\alpha_\equiv = \{(q, [q]_\equiv) \mid q \in \mathsf{Q}\}$ will henceforth be called the *abstraction relation.* The relation $\mathsf{rel}^\gamma_\equiv \subseteq \mathsf{Q}_\equiv \times \mathsf{Q}$ defined as $\mathsf{rel}^\gamma_\equiv = \{([q]_\equiv, q') \mid [q]_\equiv \in \mathsf{Q}_\equiv, q \equiv q'\}$ will henceforth be called *concretization relation.*

Remark 4.4. The relation $\mathsf{rel}^\alpha_\equiv$ is total and functional and hence represents a function α which is often called the *abstraction map* in literature. Please note that one can define the equivalence $\equiv$ via the function α. The relation $\mathsf{rel}^\gamma_\equiv$ is total (not necessarily functional) and hence represents a map into the power-set 2^{Q}. The function $\gamma : \mathsf{Q}_\equiv \to 2^{\mathsf{Q}}$ defined as $\gamma(a) = \mathsf{rel}^\gamma_\equiv(a)$ is often called the *concretization map* in literature.

We conclude this section by making a couple of observations about the construction of the abstract MDP. First notice that the abstract MDP $\mathcal{M}_\equiv$ has been defined to ensure that it simulates $\mathcal{M}$ via the canonical simulation relation $\mathsf{rel}^\alpha_\equiv$. Next, we show that we can obtain a "refinement" of the abstract MDP $\mathcal{M}_\equiv$, by considering the abstraction of $\mathcal{M}$ with respect to another equivalence $\simeq$ that is finer than $\equiv$. This is stated next.

[3] It is possible to avoid lifting distributions if one assumes that each transition in the system is uniquely labeled, and has the property that the target sub-probability measure has non-zero measure for at most one state. This does not affect the expressive power of the model and is used in [Hermanns et al. 2008]. However the disadvantage is that the abstract model may be larger as fewer transitions will be collapsed.

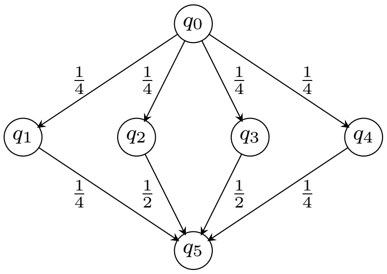

Fig. 8. DTMC $\mathcal{M}_{\mathsf{ex}}$

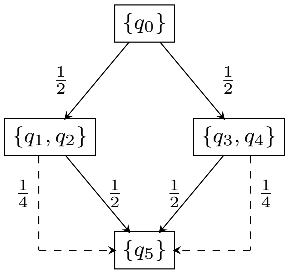

Fig. 9. MDP $\mathcal{M}_{\mathsf{ab}}$. The state $\{q_1, q_2\}$ (and $\{q_3, q_4\}$) has a nondeterministic choice between two transitions corresponding to q_1's (q_4's) transition and q_2's (q_3's) transition in $\mathcal{M}_{\mathsf{ex}}$. The former is shown with dashed edge and the latter as a solid edge.

Definition 4.5. Let $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ be a MDP over the set of atomic propositions AP. Further let $\equiv$ and $\simeq$ be two equivalence relations compatible with $\mathcal{M}$ such that $\simeq \subseteq \equiv$. The abstract MDP $\mathcal{M}_{\simeq}$ is said to be a *refinement* of $\mathcal{M}_{\equiv}$. The relation $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \subseteq \mathsf{Q}_{\simeq} \times \mathsf{Q}_{\equiv}$ defined as $\{([q']_{\simeq}, [q]_{\equiv}) \mid [q']_{\simeq} \subseteq [q]_{\equiv}\}$ is said to be a refinement relation for $(\mathcal{M}_{\simeq}, \mathcal{M}_{\equiv})$.

The following is an immediate consequence of the definition.

PROPOSITION 4.6. *Let $\simeq$ and $\equiv$ be two equivalence relations compatible with the MDP $\mathcal{M}$ such that $\simeq \subseteq \equiv$. Recall that the refinement relation for $(\mathcal{M}_{\simeq}, \mathcal{M}_{\equiv})$ is denoted by $\mathsf{rel}^{\alpha}_{\simeq,\equiv}$. Then $\mathsf{rel}^{\alpha}_{\simeq,\equiv}$ is a canonical simulation and $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathsf{rel}^{\alpha}_{\simeq} = \mathsf{rel}^{\alpha}_{\equiv}$.*

Example 4.7. Consider for example, the DTMC $\mathcal{M}_{\mathsf{ex}}$ from Figure 8 where q_0 is the initial state and proposition P labels q_5. Let $\equiv$ be the equivalence relation which partitions the set $\{q_0, q_1, q_2, q_3, q_4, q_5\}$ into the equivalence classes $\{q_0\}$, $\{q_1, q_2\}$, $\{q_3, q_4\}$ and $\{q_5\}$. The resulting MDP $\mathcal{M}_{\mathsf{ab}}$ is given in in Figure 9. Here the initial state is $\{q_0\}$ and P labels $\{q_5\}$.

5. COUNTEREXAMPLE GUIDED REFINEMENT

As described in Section 4, in our framework, a MDP $\mathcal{M}$ will be abstracted by another MDP $\mathcal{M}_{\equiv}$ defined on the basis of an equivalence relation $\equiv$ on the states of $\mathcal{M}$. Model checking $\mathcal{M}_{\equiv}$ against a safety property ψ_S will either tell us that ψ_S is satisfied by $\mathcal{M}_{\equiv}$ (in which case, it is also satisfied by $\mathcal{M}$ as shown in Lemma 2.11) or it is not. If $\mathcal{M}_{\equiv} \not\Vdash \psi_S$ then $\mathcal{M}_{\equiv}$ can be analyzed to obtain a minimal counterexample $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$, using the algorithm in Figure 5. The counterexample $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ must be analyzed to decide whether $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ proves that $\mathcal{M}$ fails to satisfy ψ_S, or the counterexample is *spurious* and the abstraction (or rather the equivalence relation $\equiv$) must be refined to “eliminate” it. In order to carry out these steps, we need to first identify what it means for a counterexample to be *valid and consistent* for $\mathcal{M}$, describe and analyze an algorithm to check validity, and then demonstrate how the abstraction can be refined if the counterexample is spurious. In this section,

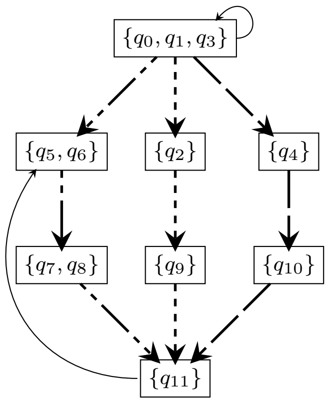

Fig. 10. The three counterexamples $\mathcal{K}_{\mathsf{cex}_1}$ (shown with short dashed edges), $\mathcal{K}_{\mathsf{cex}_2}$ (shown with long dashed edges), and $\mathcal{K}_{\mathsf{cex}_3}$ (shown with short and long dashed edges)

we will outline our proposal to carry out these steps. We will frequently recall how these steps are carried out in the non-probabilistic case through a running example to convince the reader that our definitions are a natural generalization to the probabilistic case.

5.1 Checking Counterexamples

Checking if a counterexample proves that the system $\mathcal{M}$ fails to meet its requirements ψ_S, intuitively, requires one to check if the "behavior" (or behaviors) captured by the counterexample are indeed exhibited by the system. The formal concept that expresses when a systems exhibits certain behaviors is *simulation*. Thus, one could potentially consider defining a valid counterexample to be one that is simulated by the MDP $\mathcal{M}$. However, as we illustrate in this section, the notion of valid counterexamples that is used in the context of non-probabilistic systems [Clarke et al. 2000; Clarke et al. 2002] is stronger. We, therefore, begin by motivating and formally defining when a counterexample is valid and consistent (Section 5.1.1), and then present and analyze the algorithm for checking validity (Section 5.1.2).

5.1.1 *Validity and Consistency of Counterexamples.* In the context of non probabilistic systems, a valid counterexample is not simply one that is simulated by the original system. This is illustrated by the following example; we use this to motivate our generalization to probabilistic systems.

Example 5.1. Recall the Kripke structure $\mathcal{K}_{\mathsf{ex}}$ given in Example 4.1 along with the abstraction $\mathcal{K}_{\mathsf{ab}}$ (these structures are given in Figures 6 and 7, respectively). The LTL safety-property $\Box(\neg P)$ is violated by $\mathcal{K}_{\mathsf{ab}}$. For such safety properties, counterexamples are just paths in $\mathcal{K}_{\mathsf{ab}}$ (which of course can be viewed as Kripke structures in their own right). The counterexample generation algorithms in [Clarke et al. 2000; Clarke et al. 2002] could possibly generate any one of three paths in $\mathcal{K}_{ab}$ shown in Fig 10: $\mathcal{K}_{\mathsf{cex}_1} = \{q_0, q_1, q_3\} \rightarrow \{q_2\} \rightarrow \{q_9\} \rightarrow \{q_{11}\}$, $\mathcal{K}_{\mathsf{cex}_2} = \{q_0, q_1, q_3\} \rightarrow \{q_4\} \rightarrow \{q_{10}\} \rightarrow \{q_{11}\}$ and $\mathcal{K}_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_5, q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$. Now each of the counter-examples $\mathcal{K}_{\mathsf{cex}_1}, \mathcal{K}_{\mathsf{cex}_2}$ and $\mathcal{K}_{\mathsf{cex}_3}$ is simulated by $\mathcal{K}_{\mathsf{ex}}$ because $\mathcal{K}_{\mathsf{ex}}$ has a path, starting from the initial state, having 4 states, where only the fourth state satisfies proposition P. However, the algorithm outlined in [Clarke et al. 2000; Clarke et al. 2002] only considers $\mathcal{K}_{\mathsf{cex}_1}$ to be valid. In order to see this, let us recall how the algorithm proceeds. The algorithm starts from the last state of the counterexample and proceeds backwards, checking at each point whether any of the concrete states corresponding to the abstract state in the counterexample can exhibit the counterexample from that point onwards. It also checks that the initial concrete state can simulate the counterexample. Thus, $\mathcal{K}_{\mathsf{cex}_2}$ is invalid because q_0 does not have a transition to q_4 and $\mathcal{K}_{\mathsf{cex}_3}$ is invalid because none of q_0, q_1, or q_3 have a transition to q_6 (the only state among q_5 and q_6 that can exhibit $\{q_5, q_6\} \to \{q_7, q_8\} \to \{q_{11}\}$).

The example above illustrates that to check validity, the algorithm searches for a simulation relation, wherein each (abstract) state of the counterexample is mapped to one of the concrete states that correspond to it, rather than an arbitrary simulation relation. Thus the "proof" for the validity of a counterexample in a concrete system, must be "contained" in the proof that demonstrates the validity of the counterexample in the abstract system. Based on this intuition we formalize the notion of when a counterexample is valid and consistent.

Definition 5.2. Let $\mathcal{M}$ be a MDP with set of states Q, and $\equiv$ be an equivalence relation that is compatible with $\mathcal{M}$. Let ψ_S be a PCTL-safety formula such that $\mathcal{M}_\equiv \not\Vdash \psi_S$ and let $(\mathcal{E}, \mathcal{R}_0)$ be a counterexample for $\mathcal{M}_\equiv$ and ψ_S with set of states $\mathsf{Q}_\mathcal{E}$. We say that the counterexample $(\mathcal{E}, \mathcal{R}_0)$ is *valid* and *consistent with* $(\mathcal{M}, \equiv)$ if there is a relation $\mathcal{R} \subseteq \mathsf{Q}_\mathcal{E} \times \mathsf{Q}$ such that

(1) $\mathcal{R}$ is a canonical simulation (validity); and
(2) $\mathsf{rel}_\equiv^\alpha \circ \mathcal{R} \subseteq \mathcal{R}_0$ (consistency).

The relation $\mathcal{R}$ is said to be a *validating simulation.* If no such $\mathcal{R}$ exists then $(\mathcal{E}, \mathcal{R}_0)$ is said to be *invalid for* $(\mathcal{M}, \equiv)$.

The above definition provides one technical reason for why it is convenient to view a counterexample as not just a MDP but rather as a MDP along with a simulation relation; we will see another justification for this when we discuss refinement.

Remark 5.3. When the counterexample $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ is generated as in Theorem 3.16, $\mathsf{Q}_\mathcal{E} \subseteq \{\bar{a} \mid a \in \mathsf{Q}_\equiv\}$ and the relation $\mathsf{rel}_{\mathsf{inj}} = \{(\bar{a}, a) \mid \bar{a} \in \mathsf{Q}_\mathcal{E}\}$. In this case, please note that consistency is equivalent to requiring that $\mathcal{R} \subseteq \{(\bar{a}, q) \mid q \in \mathsf{rel}_\equiv^\gamma(a)\}$. In other words, consistency is equivalent to requiring that $\mathcal{R} \subseteq \mathsf{rel}_\equiv^\gamma \circ \mathsf{rel}_{\mathsf{inj}}$.

We conclude this section by showing that for minimal counterexamples $(\mathcal{E}, \mathcal{R}_0)$ the containment in the consistency requirement can be taken to be equality.

PROPOSITION 5.4. *Let $\mathcal{M}$ be a MDP, $\equiv$ an equivalence relation compatible with $\mathcal{M}$ and ψ_S be a safety formula such that $\mathcal{M}_\equiv \not\Vdash \psi_S$. If $(\mathcal{E}, \mathcal{R}_0)$ is a minimal counterexample for $\mathcal{M}_\equiv$ and ψ_S and $(\mathcal{E}, \mathcal{R}_0)$ is consistent and valid for $(\mathcal{M}, \equiv)$ with validating simulation $\mathcal{R}$ then $\mathsf{rel}_\equiv^\alpha \circ \mathcal{R} = \mathcal{R}_0$.*

PROOF. We have that $\mathcal{R}$ is a simulation and $\mathsf{rel}_\equiv^\alpha \circ \mathcal{R} \subseteq \mathcal{R}_0$. Since $\mathsf{rel}_\equiv^\alpha$ and $\mathcal{R}$ are simulations, so is $\mathsf{rel}_\equiv^\alpha \circ \mathcal{R}$. Also since $\mathcal{E} \not\Vdash \psi_S$, we get that $(\mathcal{E}, \mathsf{rel}_\equiv^\alpha \circ \mathcal{R})$ is also

```
Initially $R = \{(\bar{a}, q) \mid q \in \mathsf{rel}^{\gamma}_{\equiv}(\bar{a})\}$ and $R_{old} = \emptyset$
while ($R_{old} \neq R$) do
    $R_{old} = R$
    For each state $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$ and each $\mu \in \delta_{\mathcal{E}}(\bar{a})$ do
        $R = \{(\bar{b}, q) \in R \,|\, b \neq a\} \cup \{(\bar{a}, q) \in R \,|\, \exists \mu' \in \delta(q).\ \mu \preceq_{R_{old}} \mu'\}$
        If $\mathcal{R}(\bar{a}) = \emptyset$ then return ("invalid",$\bar{a}$, $\mu$, $R_{old}$, $R$)
        If $\bar{a} = q_{\mathcal{E}}$ and $q_{\mathcal{I}} \notin R(\bar{a})$ then return ("invalid",$\bar{a}$, $\mu$, $R_{old}$, $R$)
     od ← end of For loop
od ← end of while loop
Return ("valid")
```

Fig. 11. Algorithm for checking validity and consistency of counterexamples

a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S. Thus, if $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R} \subsetneq \mathcal{R}_0$, then $(\mathcal{E}, \mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R})$ is not minimal. Hence $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R} = \mathcal{R}_0$. □

5.1.2 *Algorithm to check Validity of Counterexamples.* We now present the algorithm to check the validity and consistency of a counterexample. We will assume that the counterexample is a minimal one, generated by the algorithm in Theorem 3.16. Thus the counterexample is of the form $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$, where the set of states $\mathsf{Q}_{\mathcal{E}}$ is a subset of $\{\bar{a} \mid a \in \mathsf{Q}_{\equiv}\}$ and the relation $\mathsf{rel}_{\mathsf{inj}}$ is $\{(\bar{a}, a) \mid \bar{a} \in \mathsf{Q}_{\mathcal{E}}\}$. The algorithm for counterexample checking is then the standard simulation checking algorithm [Baier et al. 2000] that computes the validating simulation through progressive refinement, except that in our case we start with $R = \{(\bar{a}, q) \mid q \in \mathsf{rel}^{\gamma}_{\equiv}(a)\}$. The algorithm is shown in Figure 11. Please note that for the rest of the paper (and in the algorithm) by $\mu \preceq_{R_{old}} \mu'$ we mean $\mu \preceq_{id_{\mathsf{Q}} \cup R_{old} \cup id_{\mathsf{Q}_{\mathcal{E}}}} \mu'$.

We will now show that the algorithm in Figure 11 is correct. We start by showing that the algorithm terminates.

PROPOSITION 5.5. *The counterexample checking algorithm shown in Figure 11 terminates.*

PROOF. Let $\mathcal{R}^n_0$ and $\mathcal{R}^n_1$ be respectively the relations denoted by the variable R at the beginning and the end of the n-th iteration of the while loop. A simple inspection of the algorithm tells us that either $\mathcal{R}^n_1 = \mathcal{R}^n_0$ or $\mathcal{R}^n_1 \subsetneq \mathcal{R}^n_0$. If $\mathcal{R}^n_1 = \mathcal{R}^n_0$ then the while loop terminates (and returns "valid"). Otherwise the size of relation denoted by variable R decreases by at least one. Hence, if $\mathcal{R}^n_1$ is never equal to $\mathcal{R}^n_0$ then it must be case that $\mathcal{R}^n_1(\bar{a})$ becomes empty for some n and some $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$ and then the algorithm terminates. □

We now show that if the algorithm returns "valid" then the counterexample $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ is valid and consistent.

PROPOSITION 5.6. *If the algorithm in Figure 11 returns "valid" then the counterexample* $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ *is valid and consistent for* $(\mathcal{M}, \equiv)$.

PROOF. Please note that the algorithm returns "valid" only when the while loop terminates. At that point the variables R_{old} and R denote the same relation, which we will call $\mathcal{R}$ for the rest of the proof. Please note as $\mathcal{R} \subseteq \{(\bar{a}, q) \mid q \in \mathsf{rel}^{\gamma}_{\equiv}(\bar{a})\}$, $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R} \subseteq \mathsf{rel}_{\mathsf{inj}}$. Hence, the result will follow if we can show that $\mathcal{R}$ is a canonical simulation. Consider the last iteration of the while loop. Now, a simple inspection of the algorithm says that the variable R does not change its value in this iteration. The only way this is possible is if for each $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, $\mu \in \delta_{\mathcal{E}}(\bar{a})$ and each $(\bar{a}, q) \in \mathcal{R}$, there exists $\mu' \in \delta(q)$ such that $\mu \preceq_{\mathcal{R}} \mu'$. Also, $q_{\mathcal{I}} \in \mathcal{R}(q_{\mathcal{E}})$. Thus $\mathcal{R}$ is a canonical simulation. □

We now show that the if $(\mathsf{Q}_{\mathcal{E}}, \mathsf{rel}_{\mathsf{inj}})$ is valid and consistent, then the algorithm in Figure 11 must return "valid".

PROPOSITION 5.7. *If the counterexample* $(\mathsf{Q}_{\mathcal{E}}, \mathsf{rel}_{\mathsf{inj}})$ *is valid and consistent with* $(\mathcal{M}, \equiv)$ *then the algorithm in Figure 11 returns "valid".*

PROOF. Assume that the counterexample $(\mathsf{Q}_{\mathcal{E}}, \mathsf{rel}_{\mathsf{inj}})$ is valid and consistent. Thus there is a canonical simulation $\mathcal{R} \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}_{\equiv}$ such that $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R} = \mathsf{rel}_{\mathsf{inj}}$ (equality is a consequence of minimality; see Proposition 5.4). We make the following observations:

(1) As $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R} = \mathsf{rel}_{\mathsf{inj}}$, $\mathcal{R} \subseteq \{(\bar{a}, q) \mid q \in \mathsf{rel}^{\gamma}_{\equiv}(a)\}$.
(2) For each $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, $\mathcal{R}(\bar{a}) \neq \emptyset$ (otherwise $(\bar{a}, a)$ will be present in $\mathsf{rel}_{\mathsf{inj}}$ but not in $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R}$).
(3) $q_{\mathcal{E}} \mathcal{R} q_{\mathcal{I}}$ ($\mathcal{R}$ is a simulation).
(4) For each $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, each $\mu \in \delta(\bar{a})$ and each $\bar{a} \,\mathcal{R}\, q$ there exists a $\mu' \in \delta(q)$ such that $\mu \preceq_{\mathcal{R}} \mu'$ ($\mathcal{R}$ is a simulation).
(5) For any relation $\mathcal{R}_1 \subset \{(\bar{a}, q) \mid q \in \mathsf{rel}_{\gamma}(a)\}$ such that $\mathcal{R} \subset \mathcal{R}_1$, $\mu_a \in \delta(\bar{a})$, $\mu_q \in \delta(q)$ we have that $\mu_a \preceq_{\mathcal{R}} \mu_q$ implies that $\mu_a \preceq_{\mathcal{R}_1} \mu_q$.

Now, the first observation above implies that in the algorithm in Figure 11 initially $\mathcal{R}$ is contained within the relation denoted by the variable R. $\mathcal{R}$ is also contained in the relation denoted by the variable R_{old}, the first time the variable R_{old} takes a non-empty value. From this point on, we claim $\mathcal{R}$ is always contained in the relations R_{old} and R. This claim is a consequence of the fourth and the fifth observations which ensure that every time R is updated, $\mathcal{R}$ is contained in the relation denoted by R. Finally note that second and third observations ensure that the algorithm can never declare the counterexample to be invalid and hence by termination (Proposition 5.5), the algorithm must return "valid". □

A careful analysis of the special structure of the validating simulation yields better bounds than that reported in [Baier et al. 2000] for general simulation.

THEOREM 5.8. *Let* $\mathcal{M}$ *be a MDP,* $\equiv$ *an equivalence relation compatible with* $\mathcal{M}$*, and* ψ_S *a safety property such that* $\mathcal{M}_{\equiv} \not\Vdash \psi_S$*. Let* $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ *be a counterexample for* $\mathcal{M}_{\equiv}$ *and* ψ_S *generated using Theorem 3.16. Then the algorithm in Figure 11 returns "valid" iff* $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ *is valid and consistent with* $(\mathcal{M}, \equiv)$*. Let* $n_{\mathcal{M}}$ *and* $m_{\mathcal{M}}$ *be the number of vertices and edges, respectively, in the underlying labeled graph* $G_{\ell}(\mathcal{M})$*. The algorithm shown in Figure 11 runs in time* $O(n^2_{\mathcal{M}} m^2_{\mathcal{M}})$*.*

PROOF. Thanks to Propositions 5.6 and 5.7, the result will follow if we show that the running time of the algorithm is $O(n^2_{\mathcal{M}} m^2_{\mathcal{M}})$.

Observe that the outermost while loop runs for as long as R changes. If for each $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, we define $s_a = |\mathsf{rel}^{\gamma}_{\equiv}(a)| = |\mathsf{rel}^{\gamma}_{\equiv} \circ \mathsf{rel}_{\mathsf{inj}}(\bar{a})|$, a bound on the number of iterations of the outermost loop is $\sum_{\bar{a} \in \mathsf{Q}^{\mathcal{E}}} s_a = n_{\mathcal{M}}$, as each state of Q belongs to at most one $\mathsf{rel}^{\gamma}_{\equiv}(a)$. Next let us define d_a to be number of outgoing edges from the set $\mathsf{rel}^{\gamma}_{\equiv}(a)$ and d_q to be the out-degree of q in $G_\ell(\mathcal{M})$. Clearly for each state $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, the total number of tests of the form $\mu \preceq_{R_{old}} \mu'$ is bounded by $\sum_{q \in \mathsf{rel}^{\gamma}_{\equiv}(a)} d_a d_q = d_a^2$. Thus, the total number of tests $\mu \preceq_{R_{old}} \mu'$ in a single iteration of the outermost loop is bounded by $\sum_{\bar{a} \in \mathsf{Q}_{\mathcal{E}}} d_a^2 \leq m_{\mathcal{M}}^2$. Now, because each state of Q belongs to at most one $\mathsf{rel}^{\gamma}_{\equiv} \circ \mathsf{rel}_{\mathsf{inj}}(\bar{a})$, the test $\mu \preceq_{R_{old}} \mu'$ simply requires one to check that for each $\bar{b} \in \mathsf{Q}_{\mathcal{E}}$, $\mu(\bar{b}) \leq \sum_{q:(\bar{a},q) \in R_{old}} \mu'(q)$ and thus can be done in $O(n_{\mathcal{M}})$ time. Thus we don't need to compute flows in bipartite networks, as is required in the general case [Baier et al. 2000]. The total running time is, therefore, $O(n_{\mathcal{M}}^2 m_{\mathcal{M}}^2)$. □

We make some observations about the validity checking algorithm.

(1) For linear time properties and non-probabilistic systems, checking validity of a counterexample simply determines if the first state in the counterexample can be simulated by the initial state of the system by going backwards from the last state of the counterexample. The same idea can be exploited for probabilistic systems as well if the underlying unlabeled graph of the counterexample $\mathcal{E}$ is a tree (or more generally a DAG); the resulting algorithm will depend on the height of the counterexample and cut the running time of the algorithm by a factor of $n_{\mathcal{M}}$.

(2) Theorem 3.4 observes that for weak safety formulas, counterexamples whose underlying graph is a tree can be found by "unrolling" the minimal MDP counterexample. Instead of first explicitly unrolling the counterexample and then checking, one can unroll the counterexample "on the fly" while checking validity. This algorithm is presented in Section 5.3, after our discussion on refinement so as to not interrupt the flow. The crucial idea is to decide when to stop unrolling which is made by keeping track of the satisfaction of subformulas at various states. The running time of the algorithm will be $O(h \cdot n_{\mathcal{M}} m_{\mathcal{M}}^2)$, where h is the height of the unrolled tree. Thus depending on $n_{\mathcal{M}}$ and h, one could either compute the actual simulation relation, or simply check whether the tree of height h is simulated.

(3) One can construct the graph of maximal strongly connected components of $G_\ell(\mathcal{E})$, and compute the simulation relation on each maximal strongly connect component, in the order of their topological sort. While this new algorithm will not yield better asymptotic bounds, it may work better in practice.

To complete the description of the CEGAR approach, all we need to do is describe the refinement step. However, before we proceed, we describe a result and give some notations for the case when the counterexample generated by the algorithm in Figure 5 is declared by the counterexample checking algorithm to be invalid.

PROPOSITION 5.9. *If the algorithm in Figure 11 returns ("invalid", $\bar{a}$, μ, R_{old}, R), then for all $q \in R_{old}(\bar{a}) \setminus \mathcal{R}(\bar{a})$ and all $\mu_1 \in \delta(q)$, $\mu \not\preceq_{R_{old}} \mu_1$.*

PROOF. Immediate consequence of the algorithm. □

Notation 5.10. If the algorithm in Figure 11 returns ("invalid", $\bar{a}$, μ, R_{old}, R) then

—$\bar{a}$ is said to be an *invalidating abstract state*;

—$\mu \in \delta_{\mathsf{Q}_{\mathcal{E}}}(\bar{a})$ is said to be an *invalidating transition*; and

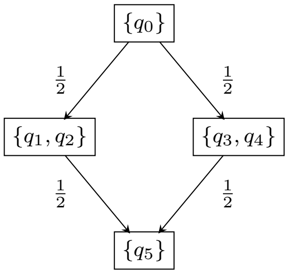

Fig. 12. Counterexample $\mathcal{E}$ for MDP $\mathcal{M}_{\mathsf{ab}}$ in Figure 9

—the pair (R_{old}, R) is said to be the *invalidating witness* for $\bar{a}$ and μ.

Example 5.11. Consider for example, the DTMC $\mathcal{M}_{\mathsf{ex}}$ and its abstraction $\mathcal{M}_{\mathsf{ab}}$ from Example 4.7, shown in Figures 8 and 9 respectively. Recall that q_0 is the q_0 is the initial state and proposition P labels q_5. Consider the safety formula $\psi_S = \mathcal{P}_{\leq \frac{2}{3}}(\mathsf{X}(\neg\mathcal{P}_{<\frac{1}{2}}(\mathsf{X}P)))$. The MDP $\mathcal{M}_{\mathsf{ab}}$ violates ψ_S and the counter-example generation algorithm will output the counter-example $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ where $\mathcal{E}$ shown in Figure 12 is a DTMC. The relation $\mathsf{rel}_{\mathsf{inj}}$ is $\{(\bar{a}, a) \mid a \text{ is a state of } \mathcal{M}_{\mathsf{ab}}\}$. Now, $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ is not valid for $\mathcal{M}_{\mathsf{ab}}$ and the counter-example checking algorithm will output "invalid" in the second iteration of the while loop in Figure 11. The invalidating abstract state is the initial state $\overline{\{q_0\}}$, the invalidating transition is the transition out of $\overline{\{q_0\}}$ and invalidating witness will be (R_{old}, R) where

—$R_{old} = \{(\overline{\{q_0\}}, \{q_0\}), (\overline{\{q_1, q_2\}}, \{q_2\}), (\overline{\{q_3, q_4\}}, \{q_3\}), (\overline{\{q_5\}}, \{q_5\})\}$, and
—$R = \{(\overline{\{q_1, q_2\}}, \{q_2\}), (\overline{\{q_3, q_4\}}, \{q_3\}), (\overline{\{q_5\}}, \{q_5\})\}$.

5.2 Refining Abstractions

The last step in the abstraction-refinement loop is to refine the abstraction in the case when the counterexample is invalid. The algorithm in Figure 11, concludes the invalidity of the counterexample, when it finds some abstract state a such that $\bar{a}$ is a state of the counterexample and $\bar{a}$ is not simulated by any concrete state in $\mathsf{rel}_{\gamma}(a)$, or when $\bar{a}$ is the initial state of the counterexample and it is not simulated by the initial state of $\mathcal{M}$. At this point, we will refine the abstraction by refining the equivalence $\equiv$ that was used to construct the abstract model in the first place. The goal of the refinement step is for it to be "counterexample guided". The ideal situation is one where the spurious counterexample is "eliminated" by the refinement step. However, as we remind the reader, this is not achieved in the CEGAR approach for non-probabilistic systems. We, therefore, begin (Section 5.2.1) by motivating and defining the notion of a "good refinement". We show that good refinements do indeed lead to progress in the CEGAR approach. After this, in Section 5.2.2, we present a refinement algorithm along with a proof that it results in good refinements.

5.2.1 *Good Refinements.* We begin by recalling the refinement step in the CEGAR approach for non-probabilistic systems through an example, to demonstrate

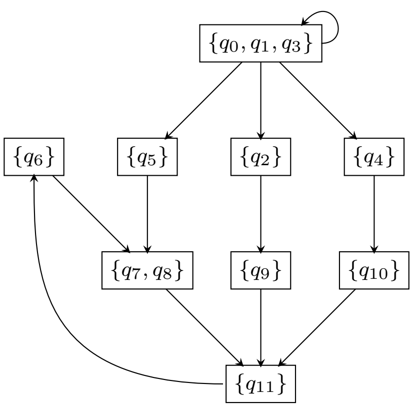

Fig. 13. The abstraction $\mathcal{K}'_{\mathsf{ab}}$

that refinement does not lead to the elimination of the counterexample. The example, however, motivates what the refinement step does indeed achieve, leading us to the notion of good refinements.

Example 5.12. As in Example 5.1, consider the Kripke structure $\mathcal{K}_{\mathsf{ex}}$ from Example 4.1 along with the abstraction $\mathcal{K}_{\mathsf{ab}}$ (these structures are also given in Figures 6 and 7). The LTL safety-property $\phi = \Box(\neg P)$ is violated by $\mathcal{K}_{\mathsf{ab}}$ and the counterexample generation algorithm may generate the spurious counterexample $\mathcal{K}_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_5, q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$. Now, the counterexample checking algorithm in [Clarke et al. 2000; Clarke et al. 2002] starts from the last state of the counterexample and proceeds backwards, checking at each point whether any of the concrete states corresponding to the abstract state in the counterexample can exhibit the counterexample from that point onwards. The algorithm finds that there is a path $q_6 \rightarrow q_7 \rightarrow q_{11}$ in $\mathcal{K}_{\mathsf{ex}}$ which simulates $\{q_5, q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$ but finds that there is no transition from $\{q_0, q_1, q_3\}$ to q_6. At this point, it declares the counterexample to be invalid. The refinement step then breaks the equivalence class $\mathsf{post}(\{q_0, q_1, q_3\}) = \{q_5, q_6\}$ into $\{q_6\}$ and $\{q_5\}$. The resulting abstraction $\mathcal{K}'_{\mathsf{ab}}$ shown in Figure 13 still has the "spurious" counterexample $\mathcal{K}'_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_5\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$ "contained" within $\mathcal{K}_{\mathsf{cex}_3}$.

Though, in Example 5.12, the counterexample is not eliminated by the refinement, progress is nonetheless made. Considering the example carefully, one notes that breaking $\{q_5, q_6\}$ could have yielded two possible new paths: $\mathcal{K}'_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_5\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$ and $\mathcal{K}''_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$. However, only one path $\mathcal{K}'_{\mathsf{cex}_3}$ is a simulated by $\mathcal{K}'_{\mathsf{ab}}$ while the other path $\mathcal{K}''_{\mathsf{cex}_3}$ is not simulated by $\mathcal{K}'_{\mathsf{ab}}$ and hence has been "eliminated". Thus, what is eliminated is at least one simulation relation that is "contained" in the original spurious counterexample. We capture this concept for MDPs as follows.

Definition 5.13. Let $\mathcal{M}$ be a MDP with states Q, $\simeq$ and $\equiv$ be equivalence relations on Q compatible with $\mathcal{M}$ such that $\simeq \subseteq \equiv$, and ψ_S be a safety property. Let $(\mathcal{E}, \mathcal{R})$ be a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S, where $\mathsf{Q}_{\mathcal{E}}$ is the set of states of $\mathcal{E}$ with initial state $q_{\mathcal{E}}$. Finally let $q_{\simeq}$ by the initial state of $\mathcal{M}_{\simeq}$. We say that $\simeq$ is a *good* $\equiv$*-refinement* for $(\mathcal{E}, \mathcal{R})$ if there is *some* $\mathcal{R}' \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}_{\simeq}$ such that $(q_{\mathcal{E}}, q_{\simeq}) \in \mathcal{R}'$, $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ but $\mathcal{R}'$ is not a canonical simulation (of $\mathcal{E}$ by $\mathcal{M}_{\simeq}$). If no such $\mathcal{R}'$ exists, we say that $\simeq$ is a *bad* $\equiv$*-refinement* for $(\mathcal{E}, \mathcal{R})$.

Intuitively, a good $\equiv$-refinement $\simeq$ ensures that $(\mathcal{E}, \mathcal{R}')$ is not a counterexample for $\mathcal{M}_{\simeq}$ and ψ_S. Observe that the conditions on $\mathcal{R}'$ ensure that $\mathcal{R}'$ is one of the possible proofs that $\mathcal{M}$ violates ψ_S "contained" within the counterexample $(\mathcal{E}, \mathcal{R})$. This presents yet another justification for formally treating the simulation relation (or proof) as part of the notion of a counterexample. In the absence of the simulation relation, it is difficult justify why refinement is "counterexample guided" given that the behavior (i.e., $\mathcal{E}$) itself may not be eliminated.

Remark 5.14. Before presenting the consequences of good refinements, we would like examine the formal definition more carefully, in order to highlight the subtle reasons why all the points in the definition are needed.

(1) Observe that $\mathcal{R}'$ satisfying $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ always exist: take $\mathcal{R}'$ to be any relation such that for each $q_0 \in \mathsf{Q}_{\mathcal{E}}$, $\mathcal{R}'(q_0) = \cup_{[q]_{\equiv} \in \mathcal{R}(q_0)} X_{[q]_{\equiv}}$ where $X_{[q]_{\equiv}}$ is any non-empty subset of $\{[q_1]_{\simeq} \in \mathsf{Q}_{\simeq} \mid [q_1]_{\simeq} \subseteq [q]_{\equiv}\}$. Further, any $\mathcal{R}'$ such that $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ must be of this form.

(2) We demand $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ rather than $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' \subseteq \mathcal{R}$. One can easily come up with $\mathcal{R}'$ which satisfies $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' \subseteq \mathcal{R}$ but is not a simulation (take, *e.g.* $\mathcal{R}' = \emptyset$). Note also that if $(\mathcal{E}, \mathcal{R})$ is a minimal counterexample then the set $\mathcal{R}(q_0)$ is non-empty for each $q_0 \in \mathsf{Q}_{\mathcal{E}}$. Thus, by taking $\simeq$ to be $\equiv$ and taking $X_{[q]_{\equiv}}$ to be empty for some $[q]_{\equiv}$ above, we will ensure that the resulting $\mathcal{R}'$ is not a simulation and $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' \subseteq \mathcal{R}$. Hence we would have declared $\equiv$ to be a good $\equiv$-refinement if we had not required the equality! Thus we may not be able to guarantee any progress in the CEGAR loop (see Proposition 5.16).

(3) We require that $(q_{\mathcal{E}}, q_{\simeq}) \in \mathcal{R}'$. If we had not required this condition then one could have achieved a "good" refinement by just breaking the initial state and taking $\mathcal{R}'$ to be any relation such that $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ but fails to be a simulation just because $\mathcal{R}'$ relates $q_{\mathcal{E}}$ to equivalence classes that does not contain the initial state of $\mathcal{M}_{\simeq}$.

Suppose $(\mathcal{E}, \mathcal{R})$ is a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S; hence $\mathcal{E} \not\Vdash \psi_S$. Therefore, if $\mathcal{R}' \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}_{\simeq}$ is a canonical simulation then $(\mathcal{E}, \mathcal{R}')$ is a counterexample for $\mathcal{M}_{\simeq}$ and ψ_S. The following proposition says that if $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ and $(\mathcal{E}, \mathcal{R})$ is invalid for $(\mathcal{M}, \equiv)$, then $(\mathcal{E}, \mathcal{R}')$ is invalid for $(\mathcal{M}, \simeq)$. Thus, a good refinement ensures that at least one of counterexamples "contained" within $(\mathcal{E}, \mathcal{R})$ is not a counterexample, and thereby eliminates at least one spurious counterexample that would not be eliminated by a bad refinement.

PROPOSITION 5.15. *Let $\mathcal{M}$ be a MDP with Q as the set of states, $\simeq$ and $\equiv$ be equivalence relations compatible with $\mathcal{M}$ such that $\simeq \subseteq \equiv$. Let $(\mathcal{E}, \mathcal{R})$ be a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S with $\mathsf{Q}_{\mathcal{E}}$ as the set of states. Let $\mathcal{R}' \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}_{\simeq}$ be a canonical simulation such that $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$. Then $(\mathcal{E}, \mathcal{R}')$ is a counterexample for $\mathcal{M}_{\simeq}$ and ψ_S. Further, if $(\mathcal{E}, \mathcal{R})$ is invalid for $(\mathcal{M}, \equiv)$ then $(\mathcal{E}, \mathcal{R}')$ is invalid for $(\mathcal{M}, \simeq)$.*

PROOF. That $(\mathcal{E}, \mathcal{R}')$ is a a counterexample for $\mathcal{M}_{\simeq}$ and ψ_S follows from the fact that $\mathcal{R}'$ is a simulation and that $\mathcal{E} \not\Vdash \psi_S$. Assume, by way of contradiction, that $(\mathcal{E}, \mathcal{R}')$ is valid and consistent with $(\mathcal{M}, \simeq)$. Then there exists a canonical simulation $\mathcal{R}_0 \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}$ such that $\mathsf{rel}^{\alpha}_{\simeq} \circ \mathcal{R}_0 \subseteq \mathcal{R}'$. This implies that $(\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathsf{rel}^{\alpha}_{\simeq}) \circ \mathcal{R}_0 \subseteq \mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}'$. But the left hand side is $\mathsf{rel}^{\alpha}_{\equiv} \circ \mathcal{R}_0$ while the right hand side is $\mathcal{R}$. This implies that $(\mathcal{E}, \mathcal{R})$ is a valid and consistent with $(\mathcal{M}, \equiv)$ (with $\mathcal{R}_0$ as the validating simulation). A contradiction! □

We conclude this section by showing that good refinements ensure progress in the CEGAR loop.

PROPOSITION 5.16. *Let $\mathcal{M}$ be a MDP, and $\simeq$ and $\equiv$ be equivalence relations compatible with $\mathcal{M}$ such that $\simeq \subseteq \equiv$. Let $(\mathcal{E}, \mathcal{R})$ be a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S. If $\simeq$ is a good $\equiv$-refinement for $(\mathcal{E}, \mathcal{R})$, then $\simeq \subsetneq \equiv$.*

PROOF. Fix a relation $\mathcal{R}' \subseteq \mathsf{Q}_{\simeq} \times \mathsf{Q}$ such that $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R}' = \mathcal{R}$ but $\mathcal{R}'$ is not a canonical simulation. We now proceed by contradiction. Assume that $\simeq = \equiv$. Then $\mathcal{M}_{\equiv}$ and $\mathcal{M}_{\simeq}$ are the same abstract MDP and $\mathsf{rel}^{\alpha}_{\simeq,\equiv}$ is the identity relation. Thus $\mathcal{R}'$ and $\mathcal{R}$ are the same relation. But $\mathcal{R}'$ is not a simulation which contradicts that fact that $(\mathcal{E}, \mathcal{R})$ is a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S. □

5.2.2 *Algorithm for Refinement.* In this section, we will show how an abstract model can be refined based on a spurious counterexample obtained as in Theorem 3.16. Before presenting our algorithm, we recall how the refinement step proceeds for non-probabilistic systems, through an example. This will help us highlight a couple of key points about the refinement step in the non-probabilistic case.

Example 5.17. Recall in the (non-probabilistic) Example 5.12, the counterexample checking step proceeded from the last state of the counterexample trace $\mathcal{K}_{\mathsf{cex}_3} = \{q_0, q_1, q_3\} \rightarrow \{q_5, q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$ and confirmed that the concrete state q_6 can simulate the path $\{q_5, q_6\} \rightarrow \{q_7, q_8\} \rightarrow \{q_{11}\}$. Since there is no "concrete" transition from $\{q_0, q_1, q_3\}$ to q_6, the counterexample checking step concludes that the counterexample is invalid. The state $\{q_0, q_1, q_3\}$ is the counterpart of "invalidating abstract state", the transition $t = \{q_0, q_1, q_3\} \rightarrow \{q_5, q_6\}$ is the counterpart of "invaliding transition", $\{q_6\}$ is counterpart of $R_{old}(\{q_5, q_6\})$ and $\emptyset$ is counterpart of $R(\{q_0, q_1, q_3\})$. The refinement step for non-probabilistic case is obtained by splitting the equivalence class $\mathsf{post}(t, \{q_0, q_1, q_3\}) = \{q_5, q_6\}$ into $\{q_6\} = R_{old}(\{q_5, q_6\})$ and $\{q_5\} = \{q_5, q_6\} \setminus R_{old}(\{q_5, q_6\})$.

On the other hand, suppose for the Kripke structure $\mathcal{K}_{\mathsf{ex}}$ and its abstraction $\mathcal{K}_{\mathsf{ab}}$, the counterexample $\mathcal{K}_{\mathsf{cex}_2} = \{q_0, q_1, q_3\} \rightarrow \{q_4\} \rightarrow \{q_{10}\} \rightarrow \{q_{11}\}$ is chosen instead of $\mathcal{K}_{\mathsf{cex}_3}$. In this case the counterexample generation algorithm declares the counterexample to be invalid when the initial state of $\mathcal{K}_{\mathsf{ex}}$, q_0, fails to be in $R(\{q_0, q_1, q_3\})$ during the counterexample checking algorithm. In this case, $\{q_0, q_1, q_3\}$ is the "invalidating" abstract state; $\{q_0, q_2, q_3\}$ is the counterpart of $R_{old}(\{q_0, q_1, q_3\})$ and $\{q_3\}$ is the counterpart of $R(\{q_0, q_1, q_3\})$. For this case, the invalidating abstract state $\{q_0, q_1, q_3\}$ is itself broken into $R_{old}(\{q_0, q_1, q_3\}) \setminus R(\{q_0, q_1, q_3\}) = \{q_0, q_1\}$ and $\{q_0, q_1, q_3\} \setminus (R_{old}(\{q_0, q_1, q_3\}) \setminus R(\{q_0, q_1, q_3\}) = \{q_3\})$. Thus, in this case the invalidating abstract state itself is broken into equivalence classes and not its successor!

Let us examine the refinement step outlined in Example 5.17 more carefully. There are two cases to consider: when the invalidating abstract state is not the initial state, where we only split abstract state that is the target of the invalidating transition; and when the invalidating abstract state is the initial state of the counterexample, where we also split the invalidating abstract state.

To generalize to probabilistic systems, we make the following observations. Suppose $\mathcal{M}_{\equiv}$ is the abstraction of $\mathcal{M}$ with respect to $\equiv$, and let $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ be a counterexample for $\mathcal{M}_{\equiv}$ and ψ_S obtained as in Theorem 3.16 and which is invalid for $(\mathcal{M}, \equiv)$. If $\bar{a}$ is the invalidating abstract state, and $\mu \in \delta_{\mathcal{E}}(\bar{a})$ is the invalidating transition, then $\mathsf{post}(\mu, \bar{a})$ may contain several states (including $\bar{a}$ itself). Hence, our refinement step will be forced to split several equivalence classes instead of one as in the case of non-probabilistic systems. Next, we observe that in the case when the counterexample is a "path" (or more generally a DAG), the algorithm to check validity only needs to "process" each state of the counterexample once. Hence, if (R_{old}, R) is the invalidating witness, then $R_{old}(\bar{a}) = \mathsf{rel}^{\gamma}_{\equiv}(a)$. Therefore, $R_{old}(\bar{a}) \setminus R(\bar{a})$ is always $\mathsf{rel}^{\gamma}_{\equiv}(a)$ except (possibly) when the invalidating state $\bar{a}$ is the initial state of the counterexample. This is the primary reason why for non-probabilistic systems the invalidating abstract state is never split, except when it is the initial state. However, when analyzing counterexamples that could be general MDPs, the counterexample checking algorithm will need to "process" each state multiple times, and then $R_{old}(\bar{a})$ need not be $\mathsf{rel}^{\gamma}_{\equiv}(a)$, at the time the counterexample is deemed to be invalid. Thus, in our refinement algorithm, we will be forced to also split the invalidating abstract state.

The above intuitions are formalized in the refinement step shown in Figure 14; recall that for each $\bar{d} \in \mathsf{Q}_{\mathcal{E}}$, we have $R(\bar{d}) \subseteq R_{old}(\bar{d}) \subseteq \mathsf{rel}_{\gamma}(d)$. We conclude by showing that the resulting refinement is a good refinement (and hence progress is ensured in the CEGAR loop).

THEOREM 5.18. *Let $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ be a counterexample, generated using Theorem 3.16, for $\mathcal{M}_{\equiv}$ and safety property ψ_S, where $\mathcal{M}_{\equiv}$ is the abstraction of $\mathcal{M}$ with respect to the compatible equivalence relation $\equiv$. If the counterexample checking algorithm in Figure 11 returns ("invalid", $\bar{a}$, μ, R_{old}, R), then the refinement $\simeq \subseteq \equiv$ obtained as in Figure 14 is a good $\equiv$-refinement for $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$.*

PROOF. Let $\mathcal{E} = (\mathsf{Q}_{\mathcal{E}}, q_{\mathcal{E}}, \delta_{\mathcal{E}}, \mathsf{L}_{\mathcal{E}})$. Recall that $\mathcal{M}_{\simeq} = (\mathsf{Q}_{\simeq}, q_{\simeq}, \delta_{\simeq}, \mathsf{L}_{\simeq})$ is the abstract MDP for $\mathcal{M}$ and $\simeq$, where the set of states of $\mathcal{M}_{\simeq}$ are equivalence classes under $\simeq$. Consider the *functional* relation $\mathcal{R} \subseteq \mathsf{Q}_{\mathcal{E}} \times \mathsf{Q}_{\simeq}$ defined as follows.

—$(\bar{a}, a') \in \mathcal{R}$ iff a' is the $\simeq$-equivalence class $R_{old}(\bar{a}) \setminus R(\bar{a})$.
—For each $\bar{b} \in \mathsf{post}(\mu, \bar{a}) \setminus \bar{a}$, $(\bar{b}, b') \in \mathcal{R}$ iff b' is the $\simeq$-equivalence class $R_{old}(\bar{b})$.
—For each $\bar{c} \in \mathsf{Q}_{\mathcal{E}} \setminus (\mathsf{post}(\mu, \bar{a}) \cup \bar{a})$, $(\bar{c}, c') \in \mathcal{R}$ iff $c' = c$.

Please note that it is easy to see that by construction $(q_{\mathcal{E}}, q_{\simeq}) \in \mathcal{R}$; if $\bar{a}$ is not the initial state then the observation follows immediately, and otherwise, observe that $q_{\mathcal{I}} \in R_{old}(\bar{a}) \setminus R(\bar{a})$. Further, we clearly have $\mathsf{rel}^{\alpha}_{\simeq,\equiv} \circ \mathcal{R} = \mathsf{rel}_{\mathsf{inj}}$. We will show that $\mathcal{R}$ is not a canonical simulation and hence we can conclude that $\simeq$ is a good refinement.

Now, let $a_0 \in \mathsf{Q}_{\simeq}$ be the $\simeq$-equivalence class $\mathcal{R}_{old}(\bar{a}) \setminus R(\bar{a})$. Observe that $(\bar{a}, a_0) \in \mathcal{R}$. Next, recall that the violating transition $\mu \in \delta_{\mathcal{E}}(\bar{a})$. Hence the desired result

```
The refinement $\simeq$ is obtained from the equivalence $\equiv$ as follows.
If $\bar{a}$ is the invalidating abstract state, $\mu \in \delta_{\mathcal{E}}(\bar{a})$ the invalidating transition
and $(R_{old}, R)$ the invalidating witness then:
    The $\equiv$-equivalence class $\mathsf{rel}_{\gamma}(a)$ is broken into
        new $\simeq$-equivalence classes $R_{old}(\bar{a}) \setminus R(\bar{a})$ and $\mathsf{rel}_{\gamma}(a) \setminus (R_{old}(\bar{a}) \setminus R(\bar{a}))$
    For each $\bar{b} \in \mathsf{post}(\mu, \bar{a}) \setminus \bar{a}$, the $\equiv$-equivalence class $\mathsf{rel}_{\gamma}(b)$ is broken into
        new $\simeq$-equivalence classes $R_{old}(\bar{b})$ and $\mathsf{rel}_{\gamma}(b) \setminus R_{old}(\bar{b})$
    No other $\equiv$-equivalence class is refined.
```

Fig. 14. Refinement algorithm based on invalid counterexamples

will follow if we can show that for each $\mu_0 \in \delta_{\simeq}(a_0)$ we have that $\mu \not\preceq_{\mathcal{R}} \mu_0$.

We proceed by contradiction. Let μ' be such that $\mu' \in \delta_{\simeq}(a_0)$ and $\mu \preceq_{\mathcal{R}} \mu'$. By definition of abstractions, there is a $q \in \mathsf{rel}^{\gamma}_{\simeq}(a_0) = R_{old}(\bar{a}) \setminus R(\bar{a})$ and $\mu_1 \in \delta(q)$ such that $\mu' = [\mu_1]_{\simeq}$. From $\mu \preceq_{\mathcal{R}} \mu'$, we can conclude the following.

—$\mu(\bar{a}) \leq \mu_1(R_{old}(\bar{a}) \setminus R(\bar{a})) \leq \mu_1(R_{old}(a))$.

—For each $\bar{b} \in \mathsf{post}(\mu, \bar{a}) \setminus \bar{a}$, $\mu(\bar{b}) \leq \mu_1(R_{old}(\bar{b}))$.

—For $\bar{c} \in \mathsf{Q}_{\mathcal{E}} \setminus (\mathsf{post}(\mu, \bar{a}) \cup \bar{a})$, $\mu(\bar{c}) = 0 \leq \mu_1(R_{old}(\bar{c}))$.

For each $\bar{d} \in \mathsf{Q}_{\mathcal{E}}$, it follows from construction that $R_{old}(\bar{d}) \subseteq \mathsf{rel}^{\gamma}_{\equiv}(d)$. Therefore for all $\bar{d_0}, \bar{d_1} \in \mathsf{Q}_{\mathcal{E}}$ such that $\bar{d_0} \neq \bar{d_1}$, $R_{old}(\bar{d_0}) \cap R_{old}(\bar{d_1}) = \emptyset$. It now follows easily from the above observations that $\mu \preceq_{\mathcal{R}_{old}} \mu_1$ which contradicts Proposition 5.9. □

Example 5.19. Consider the DTMC $\mathcal{M}_{\mathsf{ex}}$ and its abstraction $\mathcal{M}_{\mathsf{ab}}$ from Example 4.7 and the property $\psi_S = \mathcal{P}_{\leq \frac{2}{3}}(\mathsf{X}(\neg \mathcal{P}_{< \frac{1}{2}}(\mathsf{X} P)))$ from Example 5.11. $\mathcal{M}_{\mathsf{ab}}$ violates ψ_S and the counter-example generation algorithm will output the counter-example $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ where $\mathcal{E}$ is shown in Figure 12. The counter-example is a valid counter-example, and the invalidating abstract state is the initial state $\overline{\{q_0\}}$ and the invalidating transition μ is the transition out of $\overline{\{q_0\}}$. Note that $\mathsf{post}(\overline{\{q_0\}}, \mu) = \{\overline{\{q_1, q_2\}}, \overline{\{q_3, q_4\}}\}$. Now $R_{old}(\overline{\{q_1, q_2\}}) = q_2$ and $R_{old}(\overline{\{q_3, q_4\}}) = q_3$. Thus the refinement procedure breaks two equivalence classes– the class $\{q_1, q_2\}$ is broken into $\{q_1\}$ and $\{q_2\}$ and the class $\{q_3, q_4\}$ is broken into $\{q_3\}$ and $\{q_4\}$ (strictly speaking the class $\{q_0\}$ is also broken $\{q_0\}$ and $\emptyset$). In contrast, the refinement procedure only breaks one class for non-probabilistic system.

We now exhibit an example in which all the steps of CEGAR are illustrated.

Example 5.20. Consider the DTMC $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ where–

—Q consists of $n+3$ states (here n is a some even number ≥ 2). We let $\mathsf{Q} = \{q_{\mathsf{f}}, q_{\mathsf{even}}, q_{\mathsf{odd}}\} \cup \{q_0, q_1 \ldots, q_{n-1}\}$.

—$q_{\mathcal{I}} = q_0$.

—δ is defined as follows. For each $1 \leq i < n$, $\delta(q_i) = \{\mu_i\}$ where μ_i is the unique probability distribution on Q_i that satisfies the following–

 —$\mu_i(q_{\mathsf{f}}) = \frac{1}{4}$,

 —$\mu_i(q_{\mathsf{odd}}) = \frac{1}{4}$ if i is odd,

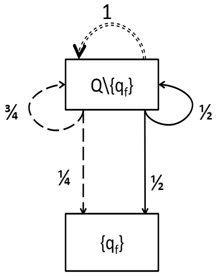

Fig. 15. MDP $\mathcal{M}_{\mathsf{ab}}$.

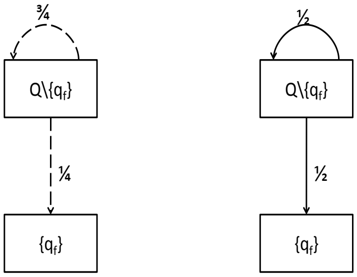

Fig. 16. $\mathcal{E}_{\mathsf{ex}_0}$ (left) and $\mathcal{E}_{\mathsf{ex}_1}$ (right).

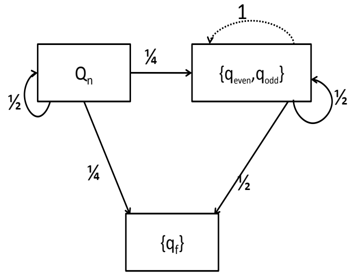

Fig. 17. $\mathcal{M}_{\mathsf{ab}_1}$.

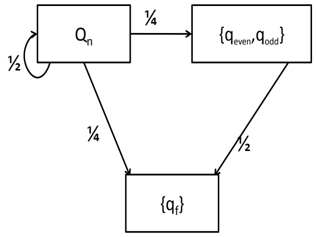

Fig. 18. $\mathcal{E}_{\mathsf{ex}_2}$.

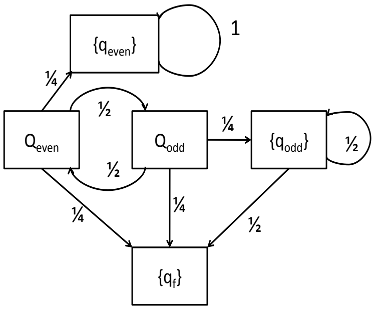

Fig. 19. $\mathcal{M}_{\mathsf{ab}_2}$.

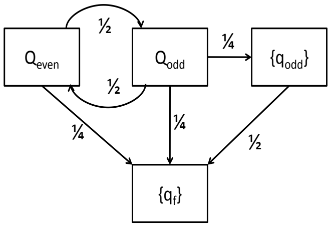

Fig. 20. $\mathcal{M}_{\mathsf{ex}_3}$.

  - $\mu_i(q_{\mathsf{even}}) = \frac{1}{4}$ if i is 0 or even,
  - $\mu_i(q_{i+1}) = \frac{1}{2}$ if $i < n-1$; and
  - $\mu_{n-1}(q_0) = \frac{1}{2}$.

  The probabilistic transition out of q_{f} assigns probability 0 to every state, the transition out of q_{odd} assigns probability $\frac{1}{2}$ to q_{f} and $\frac{1}{2}$ to q_{odd}; and the transition out of q_{even} assigns probability 1 to q_{even}.
- The labeling function L assigns proposition P to q_{f}.

Consider the safety formula $\psi = \mathcal{P}_{\leq \frac{1}{2}}(\Diamond P)$ and suppose we want to check whether

$\mathcal{M} \Vdash \psi$. We will start with a very coarse abstraction by dividing the set Q into two equivalence classes $\{q_{\mathsf{f}}\}$ and $\mathsf{Q} \setminus \{q_{\mathsf{f}}\}$. The resulting abstraction $\mathcal{M}_{\mathsf{ab}}$ is shown in Figure 15 (the initial state $Q \setminus \{q_{\mathsf{f}}\}$ has three non-deterministic choices shown by long-dashed lines, solid lines and double lines). Now $\mathcal{M}_{\mathsf{ab}}$ does not satisfy ψ and there are two possible minimal counterexamples which witness this fact. These counterexamples $\mathcal{E}_{\mathsf{ex}_0}$ and $\mathcal{E}_{\mathsf{ex}_1}$ are shown in Figure 16 (The initial state is $\mathsf{Q} \setminus \{q_{\mathsf{f}}\}$). For this example, we consider the case when the counterexample generated is $\mathcal{E}_{\mathsf{ex}_0}$.

Suppose that the counterexample generating algorithm generates the counterexample $\mathcal{E}_{\mathsf{ex}_0}$. The counterexample turns out to be invalid, and the refinement algorithm will split the equivalence class $\mathsf{Q} \setminus \{q_{\mathsf{f}}\}$ into two classes – $\{q_{\mathsf{even}}, q_{\mathsf{odd}}\}$ and $\mathsf{Q}_n = \{q_0, q_1 \ldots, q_{n-1}\}$. The splitting of $\mathsf{Q} \setminus \{q_{\mathsf{f}}\}$ results in the new abstract MDP $\mathcal{M}_{\mathsf{ab}_1}$ given in Figure 17 (the initial state is Q_n and the state $\{q_{\mathsf{even}}, q_{\mathsf{odd}}\}$ has two outgoing transitions).

Now, $\mathcal{M}_{\mathsf{ab}_1}$ also does not satisfy ψ and there is one minimal counterexample $\mathcal{E}_{\mathsf{ex}_2}$ witnessing this. This counterexample is shown in Figure 18. This counterexample turns out to be invalid and the refinement process now splits the two classes– $\{q_{\mathsf{even}}, q_{\mathsf{odd}}\}$ and $\mathsf{Q}_n = \{q_0, q_1 \ldots, q_{n-1}\}$. The class $\{q_{\mathsf{even}}, q_{\mathsf{odd}}\}$ is split into $\{q_{\mathsf{even}}\}$ and $\{q_{\mathsf{odd}}\}$. The class Q_n is split into $\mathsf{Q}_{\mathsf{even}} = \{q_0, q_2, \ldots, q_{n-2}\}$ and $\mathsf{Q}_{\mathsf{odd}} = \{q_1, q_3, \ldots, q_{n-1}\}$. The resulting abstract MDP $\mathcal{M}_{\mathsf{ab}_2}$ is shown in Figure 19 (the initial state is $\mathsf{Q}_{\mathsf{even}}$).

Now, $\mathcal{M}_{\mathsf{ab2}}$ also does not satisfy ψ, and there is exactly one minimal counterexample $\mathcal{E}_{\mathsf{ex}_3}$ which witnesses this fact. This counterexample is shown in Figure 20 (the initial state is $\mathsf{Q}_{\mathsf{even}}$). This turns out to be a valid counterexample and at this point our CEGAR loop terminates with the result— $\mathcal{M} \not\Vdash \psi$.

Remark 5.21. In the Electronic Appendix, we give another example demonstrating our CEGAR approach and show that model-checking for safety properties using the CEGAR loop can be faster than model-checking using direct methods.

5.3 Counter-example checking for weak safety

In this section we outline an algorithm that given a counterexample $\mathcal{E}$ for a MDP $\mathcal{M}$ and weak safety property ψ_{WS} either determines that $\mathcal{E}$ is not a valid counterexample, or finds a finite unrolling of $\mathcal{E}$ that is simulated by $\mathcal{M}$, and witnesses the fact that $\mathcal{M}$ does not satisfy ψ_{WS}. The algorithm unrolls $\mathcal{E}$ on the fly, and does not construct the unrolled MDP explicitly. The running time of the algorithm depends on the height of the unrolling, which if small, can result in faster checking than the algorithm shown in Figure 11. Before presenting the algorithm, we introduce some notation that we will find useful in describing the algorithm. Recall that given a MDP $\mathcal{M}$, a state q of $\mathcal{M}$ and $k \in \mathbb{N}$ the k-th unrolling of MDP rooted at q is denoted by $\mathcal{M}^q_k$.

Notation 5.22. Given $k \in \mathbb{N}$ and states $q, q' \in \mathcal{M}$, we say that $q \preceq_k q'$ if $\mathcal{M}^q_k \preceq \mathcal{M}^{q'}_k$. Given a PCTL formula ψ we say that $q \Vdash_k \psi$ if $\mathcal{M}^q_k \Vdash \psi$.

We observe the following two facts. If $q \preceq q'$ then $q \preceq_k q'$ for all k. The proof of Theorem 3.4 implies that for a weak safety formula ψ_{WS} if $q \not\Vdash \psi_{WS}$, then there is a k_0 s.t. $q \not\Vdash_{k_0} \psi_{WS}$. These two facts can be combined to obtain a counter-example checking algorithm for the weak-safety fragment of PCTL as we will describe shortly.

For the rest of the Section, we fix the following notation. $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ is the (original) MDP that we are checking against weak safety property ψ_{WS} and $\equiv$ is an equivalence relation that is compatible with $\mathcal{M}$. Assuming $\mathcal{M}_{\equiv}$ violates the safety property ψ_{WS}, we will denote the minimal counterexample obtained as in Theorem 3.16 by $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$. Let $\mathcal{E} = (\mathsf{Q}_{\mathcal{E}}, q_{\mathcal{E}}, \delta_{\mathcal{E}}, \mathsf{L}_{\mathcal{E}})$. For a state $a = [q]_{\equiv}$ in abstraction $\mathcal{M}_{\equiv}$, $\mathsf{rel}^{\gamma}_{\equiv}(a) = \{q' \mid q \equiv q'\}$ is the concretization map. The relation $\{(\bar{a}, q) \mid q \in \mathsf{rel}^{\gamma}_{\equiv}(a), \bar{a} \in \mathsf{Q}_{\mathcal{E}}\}$ will be denoted by $R_{\mathcal{I}}$. Finally, we will use ψ_{SL} to denote the strict liveness formula obtained by negating ψ_{WS}. $\mathsf{SLSubForm}(\psi_{SL})$ will denote the set of PCTL strict liveness subformulas of ψ_{SL} and $\mathsf{PathForm}(\psi_{SL})$ will denote the set of path subformulas of ψ_{SL}[4].

The proposed algorithm iteratively constructs the relations $R_k = R_{\mathcal{I}} \cap \preceq_k$ and $\mathsf{Sat}_k = \{(\bar{a}, \psi) \mid \bar{a} \Vdash_k \psi, \bar{a} \in Q_{\mathcal{E}}, \psi \in \mathsf{SLSubForm}(\psi_{SL})\}$. We make the following observations.

(1) If the set $R_k(\bar{a}) = \{q \mid (\bar{a}, q) \in R_k\}$ becomes $\emptyset$ for some k and $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$, then $(\mathcal{E}, \mathsf{rel}_{\mathsf{inj}})$ is invalid for $(\mathcal{M}, \equiv)$. We can also call the counterexample invalid if the initial concrete state $q_{\mathcal{I}}$ is not contained in $R_k(q_{\mathcal{E}})$.

(2) If $\mathsf{Sat}_k(q_{\mathcal{E}}, \psi_{SL})$ and $q_{\mathcal{I}} \in R_k(q_{\mathcal{E}})$ then $q_{\mathcal{I}} \Vdash_k \psi_{SL}$ also. Thus, the concrete MDP violates the given safety property and we can report this.

This iteration must end as a consequence of Theorem 3.4. The computation also needs to compute the function $\mathsf{MaxProb}_k$ defined as follows. Given $\bar{a} \in Q_{\mathcal{E}}$ and a path formula $\phi \in \mathsf{PathForm}(\psi_{SL})$, $\mathsf{MaxProb}_k$ gives the maximum probability (over all schedulers) of ϕ being true in $\mathcal{E}^{\bar{a}}_k$. The relations R_{k+1}, Sat_{k+1} and $\mathsf{MaxProb}_{k+1}$ can be computed using R_k, Sat_k, and $\mathsf{MaxProb}_k$ and do not need other previous values.

Figure 21 gives the details of this algorithm. At the beginning of $(k + 1)$-th unrolling of the while loop, the relations R_{curr} and $\mathsf{Sat}_{\mathsf{curr}}$ are the relations R_k and Sat_k respectively. The (doubly-indexed) array $\mathsf{MaxProb}_{\mathsf{curr}}[\bar{a}][\psi]$ is the function $\mathsf{MaxProb}_k[\bar{a}][\psi]$. Within the while loop, $R_{\mathsf{tmp},\bar{a}}$ is the set $R_{k+1}(\bar{a})$ while $\mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ is the set $\{\psi \mid \bar{a} \Vdash_{k+1} \psi \in \mathsf{SLSubForm}(\psi_{SL})\}$, and $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\psi]$ is $\mathsf{MaxProb}_{k+1}[\bar{a}][\psi]$. R_{curr}, $\mathsf{Sat}_{\mathsf{curr}}$, and $\mathsf{MaxProb}_{\mathsf{curr}}$ are updated *after* $R_{\mathsf{tmp},\bar{a}}$, $\mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ and $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}$ are computed for *all* states $\bar{a} \in \mathsf{Q}_{\mathcal{E}}$. The following proposition follows easily from the observations made in the Section.

PROPOSITION 5.23. *The algorithm in Figure 21 terminates. If the algorithm returns "Invalid Counter-example" then the counterexample obtained using Theorem 3.16 is not valid. If the algorithm returns "Safety Violated" then* $\mathcal{M} \nVdash \psi_{WS}$.

Finally, we observe that the algorithm in Figure 21 may be made more efficient in practice as follows. First, since we are dealing with strict liveness fragment, the sequence Sat_k is an increasing sequence and the function $\mathsf{MaxProb}_k[\bar{a}][\psi] \leq \mathsf{MaxProb}_{k+1}[\bar{a}][\psi]$. Hence, only needs to compute $\mathsf{Sat}_{k+1} \setminus \mathsf{Sat}_k$ and $\mathsf{MaxProb}_{k+1} - \mathsf{MaxProb}_k$. This optimization will be explored in future work.

[4] A path subformula of a PCTL-liveness formula ψ_L are formulas of the kind $\mathsf{X}\psi_L$ and $\psi_L \,\mathcal{U}\, \psi_L$

```
Initially
   $R_{\mathsf{curr}} = \{(\bar{a}, q) \mid q \in \mathsf{rel}_\gamma(\bar{a}), \bar{a} \in \mathsf{Q}_\mathcal{E}\}$
   $\mathsf{Sat}_{\mathsf{curr}} = \{(\bar{a}, \psi) \mid \bar{a} \Vdash_0 \psi, \psi \in \mathsf{SLSubForm}(\psi_{SL}), \bar{a} \in \mathsf{Q}_\mathcal{E}\}$
   $\mathsf{MaxProb}_{\mathsf{curr}}[\bar{a}][\phi] = \mathsf{MaxProb}_0(\bar{a})(\phi)$ for $\bar{a} \in \mathsf{Q}_\mathcal{E}, \phi \in \mathsf{PathForm}(\psi_{SL})$
While (true)
  do
        If $q_\mathcal{I} \notin R_{\mathsf{curr}}(q_\mathcal{E})$ return "Invalid Counter-example"
        If $\mathsf{Sat}(q_\mathcal{E}, \psi_{SL})$ return "Safety Violated"
        for each $\bar{a} \in \mathsf{Q}_\mathcal{E}$
          do
            $R_{\mathsf{tmp},\bar{a}} = \{q \mid (\bar{a}, q) \in R_{\mathsf{curr}}$ and $\forall \mu \in \delta_\mathcal{E}(\bar{a}) \exists \mu' \in \delta(q).\ \mu \preceq_{R_{\mathsf{curr}}} \mu'\}$
            If $R_{\mathsf{tmp},\bar{a}} = \emptyset$ return "Invalid Counter-example"
            COMPUTE$(\bar{a}, \mathsf{Sat}_{\mathsf{curr}}, \mathsf{MaxProb}_{\mathsf{curr}}, \mathsf{Sat}_{\mathsf{tmp},\bar{a}}, \mathsf{MaxProb}_{\mathsf{tmp},\bar{a}})$
          od
        $\mathsf{Sat}_{\mathsf{curr}} = \{(\bar{a}, \psi) \mid \psi \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}\}$
        $R_{\mathsf{curr}} = \{(\bar{a}, q) \mid q \in R_{\mathsf{tmp},\bar{a}}\}$
        $\mathsf{MaxProb}_{\mathsf{curr}}[\bar{a}][\phi] = \mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\phi]$ for each $a \in \mathsf{Q}_\mathcal{E}, \phi \in \mathsf{PathForm}(\psi)$
   od
```

The procedure COMPUTE returns $\mathsf{Sat}_{\mathsf{tmp},\bar{a}}$, the set of sub-formulas of ψ_{SL} satisfied by $\bar{a}$ in the "next" unrolling of the tree with root $\bar{a}$. It also returns $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}$, that given a path formula ϕ gives the maximum probability of ϕ being true in the next unrolling. COMPUTE is defined as follows.

```
COMPUTE$(\bar{a}, \mathsf{Sat}_{\mathsf{curr}}, \mathsf{MaxProb}_{\mathsf{curr}}, \mathsf{Sat}_{\mathsf{tmp},\bar{a}}, \mathsf{MaxProb}_{\mathsf{tmp},\bar{a}})$
  Fix an enumeration $\varphi_1, \ldots, \varphi_n$ of the set $\{\varphi \mid \varphi \in \mathsf{PathForm}(\psi_{SL}) \cup$
       $\mathsf{SLSubForm}(\psi_{SL})\}$ such that $size(\varphi_i) \leq size(\varphi_j)$ for $i \leq j$.
   Initially
     $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \emptyset$
     $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\phi] = 0$ for all $\phi \in \mathsf{PathForm}(\psi_{SL})$.
     For $i = 1$ to $n$
        If $\varphi_i$ is $p$ and $p \in \mathsf{L}_\mathcal{E}(\bar{a})$ then $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \mathsf{Sat}_{\mathsf{tmp},\bar{a}} \cup \{p\}$
        If $\varphi_i$ is $\neg p$ and $p \notin \mathsf{L}_\mathcal{E}(\bar{a})$ then $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \mathsf{Sat}_{\mathsf{tmp},\bar{a}} \cup \{\neg p\}$
        If $\varphi_i$ is $\psi_1 \vee \psi_2$ and ($\psi_1 \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ or $\psi_2 \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$) then
            $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \mathsf{Sat}_{\mathsf{tmp},\bar{a}} \cup \{\varphi_i\}$
        If $\varphi_i$ is $\psi_1 \wedge \psi_2$, $\psi_1 \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ and $\psi_2 \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ then
            $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \mathsf{Sat}_{\mathsf{tmp},\bar{a}} \cup \{\varphi_i\}$
        If $\varphi_i$ is $\neg\mathcal{P}_{\leq p}(\phi)$ and $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\phi] > p$
           then $\mathsf{Sat}_{\mathsf{tmp},\bar{a}} = \mathsf{Sat}_{\mathsf{tmp},\bar{a}} \cup \{\varphi_i\}$

        If $\varphi_i$ is $\mathsf{X}\psi$ then
            $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\varphi_i] = \max_{\mu \in \delta_\mathcal{E}(\bar{a})} \mu(\{\bar{b} \mid (\bar{b}, \psi) \in \mathsf{Sat}_{\mathsf{curr}}\})$
        If $\varphi_i$ is $\psi_1 \,\mathcal{U}\, \psi_2$ then
            If $\psi_2 \in \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ then $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\varphi_i] = 1$
            else
                If $\psi_1 \notin \mathsf{Sat}_{\mathsf{tmp},\bar{a}}$ then $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\varphi_i] = 0$
                else $\mathsf{MaxProb}_{\mathsf{tmp},\bar{a}}[\varphi_i] = \max_{\mu \in \delta_\mathcal{E}(\bar{a})} \sum_{\bar{b} \in \mathsf{Q}_\mathcal{E}} (\mu(\bar{b}))\ (\mathsf{MaxProb}_{\mathsf{curr}}[\psi_1 \,\mathcal{U}\, \psi_2][\bar{b}])$
```

Fig. 21. On the fly algorithm for checking Strict Liveness

6. RELATED WORK

Abstraction Schemes: Abstractions have been extensively studied in the context of probabilistic systems. General issues in defining good abstractions as well as specific proposals for families of abstract models are presented in [Jonsson and Larsen 1991; Huth 2004; Norman 2004; Huth 2005; D'Argenio et al. 2001; 2002; Fecher et al. 2006; Katoen et al. 2007; Monniaux 2005; Kwiatkowska et al. 2006; McIver and Morgan 2004]. MDPs as abstract models of probabilistic systems have been considered in a number of papers. They are natural abstract model because when one abstracts (whether in the non-probabilistic case or the probabilistic case) the main feature that abstract models have is nondeterminism, and in the context of probabilistic systems this means considering transition systems that have both non-determinism and probability, which is exactly the model of MDPs. Instead of explict nondeterministic choices, Markov Chains with interval bounds on transition probabilities have been proposed as abstract models in [Fecher et al. 2006]. Such "interval" Markov Chains are abstractions of the models we consider in this paper (and hence less precise), and are useful when the number of nondeterministic choices out of an abstract state are more than an exponential in the number of abstract states; in such cases model checking the interval Markov Chain can be faster, even if it yield less precise results. Another important scheme for abstract models proposed is where the system is abstracted as a game between two players. Like our construction of abstract MDPs, the states of the game correspond to equivalence classes of some relation on the concrete states. In the ensuing game, one player picks a concrete states in the equivalence class associated with the abstract states, and the second player picks one of the transitions out of the concrete state chosen by player 1. Such a game abstraction has an advantage over the MDP abstraction in that both upper and lower bounds on the probability of reaching states can be obtained by analyzing the game; the MDP abstraction will only allow one to obtain upper bounds. However, the game abstraction approach has so far only been applied to simple quantitative properties, and it is not clear how to extend it to a general logic fragment like PCTL-safety.

Recently, theorem-prover based algorithms for constructing abstractions of probabilistic systems based on predicates have been presented [Wachter et al. 2007]. Another notion that has been recently proposed is the notion of a "magnifying-lens abstraction" [de Alfaro and Roy 2007], which can be used to assist in the model checking process, by approximating the measure of the satisfaction of path formulas for sets of concrete states; the method is not an abstraction in the traditional sense in that neither is an abstract model explicitly constructed, nor is the model used for reasoning, one that simulates the concrete model.

Counterexamples: The notion of counterexamples is critical for the approach of counterexample guided abstraction refinement. Criteria for defining counterexamples are identified in [Clarke et al. 2002], along with a notion of counterexamples for branching-time properties and non-probabilistic systems. The problem of defining counterexamples for probabilistic systems has received considerable attention recently. Starting from the seminal papers [Aljazzar et al. 2005; Han and Katoen 2007a], the notion of sets of paths with high measure as counterexamples has been used for DTMCs, CTMCs, and MDPs [Han and Katoen 2007a; 2007b; Aljazzar and Leue 2007]. Another definition that has been proposed is that of DTMCs (or purely probabilistic models) in [Chatterjee et al. 2005; Hermanns et al. 2008]. Our notion of counterexample is different from these proposals and we demonstrate that these other proposals are not rich enough for the class of properties we consider.

Automatic Abstraction-Refinement: The foundations of automatic abstraction-refinement in the context of non-probabilistic systems were laid down in [Clarke et al. 2000; Clarke et al. 2002], and the technique has been applied to analyze software [Ball and Rajamani 2002; Majumdar et al. 2002; Kroening et al. 2004; Cook et al. 2005] and hardware designs [Clarke et al. 2002; Glussman et al. 2003; Bjesse and Kukula 2004] with success. CEGAR has also been extended to consider 3-valued abstractions [Gurfinkel and Chechik 2006; Shoham and Grumberg 2007]. A number of subtle heuristics have been identified that assist in the construction of abstractions, counterexample analysis, and refinement. Primarily these are identifying special equivalences on concrete states based on predicates [Das 2003; Lakhnech et al. 2001] and automata [Bouajjani et al. 2004] to construct abstractions efficiently; using SAT solvers to analyze counterexamples [Das 2003; Clarke et al. 2002; Glussman et al. 2003; Bjesse and Kukula 2004]; using SAT solvers, ILP, and learning [Clarke et al. 2002] to identify heuristics to split abstract states during refinement. In order for the CEGAR approach to scale to large systems, these heuristics, identified for non-probabilistic systems, will need to be borrowed and experimented with in the future. This paper focuses primarily on laying down the definitions and algorithms needed to serve as the foundations for the CEGAR approach for probabilistic systems.

In the context of probabilistic systems, automatic abstraction-refinement was first considered in [D'Argenio et al. 2001; 2002]. There are two main differences with our work. First, they consider only reachability properties. Second, the refinement process outlined in [D'Argenio et al. 2001; 2002] is not counterexample based, but rather based on partition refinement. What this means is that their refinement is biased towards separating states that are not bisimilar, rather than states that are "distinguished" by the property, and so it is likely that their method refines more than needed.

Counterexample guided refinement has been used as the basis of synthesizing winning strategies for 2-player stochastic games in [Chatterjee et al. 2005]. Though the problem 2-player games is more general than verification, the specific model considered by [Chatterjee et al. 2005] has some peculiarities and so does not subsume the problem or its solution presented here. First, in their model, states are partitioned into "non-deterministic" states that have purely nondeterministic transitions and "probabilistic" states that have purely probabilistic transitions. The abstraction does not abstract any of the "probabilistic states"; only the "non-deterministic" states are collapsed. This results in larger abstract models, and obviates certain issues in counterexample checking that we deal with. Second, they take counterexamples to be finite models without nondeterminism. They can do this because they consider a simpler class of properties than we do, and as we show in Section 3.3, DTMCs are not rich enough for all of safe-PCTL. Next, their counterexample checking algorithm is different than even the one used in the context of non-probabilistic systems. They consider a counterexample to be valid only if *all* the concrete states corresponding to the abstract states in the counterexample can simulate the behavior captured by the counterexample. Thus, they deem certain counterexamples to be spurious, even if they will be recognized as providing enough evidence for the violation of the property by other CEGAR schemes (including ours). Finally, they do not have a precise statement characterizing the qualities of their refinement algorithm.

In a recent paper, [Hermanns et al. 2008] consider CEGAR for probabilistic systems. They consider very special types of reachability properties namely, those that can be expressed by formulas of the form $\mathcal{P}_{\leq p}(\psi_1 \,\mathcal{U}\, \psi_2)$ where ψ_1 and ψ_2 are propositions (or boolean combinations of propositions). For this class of properties, they use DTMCs as the notion of counterexamples : the counterexample obtained is a pair $(\mathcal{S}, \mathcal{M}^{\mathcal{S}})$ where $\mathcal{S}$ is a memoryless scheduler. As we show, DTMCs cannot serve as counterexample for the richer class of properties considered here. For the counterexample checking algorithm, they generate a finite set, ap, of *abstract* paths of $\mathcal{M}^{\mathcal{S}}_{\equiv}$ in decreasing order of measures such that the total measure of these paths is $> p$. Then, they build (on-the-fly) a "concrete" scheduler which maximizes the measure of the paths in ap that are simulated by the original MDP. Let $\mathsf{p}_{\mathsf{total}}$ be the maximum probability (under all schedulers) of $\psi_1 \,\mathcal{U}\, \psi_2$ being satisfied by the abstract MDP, p_{ap} be the total measure of ap and $\mathsf{p}_{\mathsf{max}}$ be the maximum measure of "abstract" paths in ap simulated by the concrete MDP. If $\mathsf{p}_{\mathsf{max}} > p$ then the counterexample is declared to be valid and if $\mathsf{p}_{\mathsf{total}} - \mathsf{p}_{\mathsf{ap}} + \mathsf{p}_{\mathsf{max}} \leq p$ then the counterexample is declared to be invalid and the abstraction refined. If neither is the case, then [Hermanns et al. 2008] heuristically decide either to generate more abstract paths or to refine the abstraction. The refinement is based upon refining some "spurious" abstract path (namely, a path that is not simulated by the concrete system). There is, however, no formal statement characterizing progress based on the refinement algorithm outlined in [Hermanns et al. 2008].

Another automatic abstraction-refinement approach was recently proposed in [Kattenbelt et al. 2009], in which the abstract MDP is represented as a two-person stochastic game [Kwiatkowska et al. 2006]-one player picks the non-determinism of the concrete state and the other player picks the non-determinism introduced by the abstraction. They do not verify the game against a formula in a logic, instead they compute both upper and lower bounds on quantitative properties of the original MDP up-to a predefined precision. The quantitative properties considered are the maximum and minimum probabilities of reaching a state (they can also deal with expected rewards). Model-checking the two-person stochastic game not only yields upper and lower bounds on quantitative properties of the original MDP, but also the strategies for both the players that achieve the bounds. If the difference between the bounds is less than the pre-defined precision, the abstraction-refinement loop terminates otherwise they use the strategies to further refine the abstraction. The refinement is achieved by identifying an abstract state for which the strategy for achieving the upper and lower bounds is witnessed by distinct concrete states and refinement step separates these concrete states.

7. CONCLUSIONS AND FUTURE WORK

We presented a CEGAR framework for MDPs, where a MDP $\mathcal{M}$ is abstracted by another MDP $\mathcal{A}$ defined using an equivalence on the states of $\mathcal{M}$. Our main contributions when presenting this framework were a definition for the notion of a counterexample, along with algorithms to compute counterexamples, check their validity and perform automatic refinement based on an invalid counterexample. Our discussion in this paper considered only the analysis of finite state MDPs. A natural question to ask is how these ideas would translate when analyzing infinite state MDPs. As in the case of non-probabilistic systems, the notions of counterexamples, the validity of counterexamples, principles of refinement etc. will remain the same when moving to infinite state systems. However, just like for non-probabilistic systems, for even one iteration of the CEGAR loop to terminate for infinite state systems, the basic problems of constructing abstractions, and analyzing counterexamples must be decidable. These problems are in general undecidable for infinite state systems, and therefore CEGAR can only be applied to certain special tractable classes of infinite state MDPs.

There are a number of interesting questions left open for future investigation. First these ideas need to be implemented and experimented with. In order for this approach to be scalable, symbolic algorithms for the steps outlined here, will be required. In this paper, we considered abstractions defined using a general equivalence relations on concrete states. For non-probabilistic systems, it has been observed that the most useful type of equivalences are those defined using predicates [Das 2003] on concrete states. While the construction of abstractions for probabilistic systems using predicates has been explored [Hermanns et al. 2008], it will be important to see how counterexamples can be analyzed and predicates refined using SAT solvers and theorem provers. Further a number of key heuristics have been identified analyzing non-probabilistic systems using SAT solvers, and it will be interesting to see how those heuristics perform in the context of probabilistic systems.

Next, ideas and heuristics to improve the running time of our counterexample construction algorithm need to be explored. First, our algorithm makes multiple calls to a model checker which can be expensive, and it will be important to explore to leverage the results of previous model checking runs in subsequent model checking runs. Second, the order in which transitions are considered for elimination, crucially affects the final size of the counterexample. Good heuristics for ordering transitions to obtain small counterexamples, must be identified.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library by visiting the following URL: `http://www.acm.org/pubs/citations/journals/tocl/20YY-V-N/p1-Chadha`.

ACKNOWLEDGMENTS

The authors thank the following people: Chandra Chekuri for many fruitful discussions especially relating to the algorithmic aspects of finding counterexamples and checking validity of counterexamples; Radha Jagadeesan for discussions on simulations and the safety fragment of PCTL; anonymous referees for sending pointers to [Aljazzar and Leue 2007; Hermanns et al. 2008; Kattenbelt et al. 2009], for encouraging us to formally articulate the guarantees of our refinement algorithm, and improving the general presentation of the results.

REFERENCES

ALJAZZAR, H., HERMANNS, H., AND LEUE, S. 2005. Counterexamples for timed probabilistic reachability. In *Proceedings of International Conference on Formal Modelling and Analysis of Timed Systems*. 177–195.

ALJAZZAR, H. AND LEUE, S. 2007. Counterexamples for model checking of markov decision processes. Tech. Rep. soft-08-01, University of Konstanz.

BAIER, C., ENGELEN, B., AND MAJSTER-CEDERBAUM, M. E. 2000. Deciding bisimularity and simularity for probabilistic processes. *Journal of Computer and System Sciences 60,* 1, 187–231.

BAIER, C., KATOEN, J.-P., HERMANNS, H., AND WOLF, V. 2005. Comparative branching-time semantics for markov chains. *Information and Computation 200*, 149–214.

BALL, T. AND RAJAMANI, S. 2002. Generating abstract explanations of spurious counterexamples in C programs. Tech. Rep. 2002-09, Microsoft Research.

BIANCO, A. AND DE ALFARO, L. 1995. Model checking of probabalistic and nondeterministic systems. In *Proceedings of the International Conference on the Foundations of Software Technology and Theoretical Computer Science*. 499–513.

BJESSE, P. AND KUKULA, J. 2004. Using counter example guided abstraction refinement to find complex bugs. In *Proceedings of Design, Automation, and Test in Europe Conference and Exposition*. 156–161.

BOUAJJANI, A., HABERMEHL, P., AND VOJNAR, T. 2004. Abstract Regular Model Checking. In *Proceedings of the International Conference on Computer Aided Verification*. 372–286.

CHATTERJEE, K., HENZINGER, T., JHALA, R., AND MAJUMDAR, R. 2005. Counter-example Guided Planning. In *Proceedings of Uncertainity in Artificial Intelligence*. 104–111.

CLARKE, E., GRUMBERG, O., JHA, S., LU, Y., AND VEITH, H. 2000. Counterexample-guided abstraction refinement. In *Proceedings of the International Conference on Computer Aided Verification*. 154–169.

CLARKE, E., GUPTA, A., KUKULA, J., AND STRICHMAN, O. 2002. SAT based abstraction-refinement using ILP and machine learning techniques. In *Proceedings of the International Conference on Computer Aided Verification*. 265–279.

CLARKE, E., JHA, S., LU, Y., AND VEITH, H. 2002. Tree-like counterexamples in model checking. In *Proceedings of the IEEE Symposium on Logic in Computer Science*. 19–29.

COOK, B., PODELSKI, A., AND RYBALCHENKO, A. 2005. Abstraction Refinement for Termination. In *Proceedings of the International Static Analysis Symposium*. 87–101.

D'ARGENIO, P. R., JEANNET, B., JENSEN, H. E., AND LARSEN, K. G. 2001. Reachability analysis of probabilistic systems by successive refinements. In *Proceedings of the International Workshop on Performance Modeling and Verification*. 39–56.

D'ARGENIO, P. R., JEANNET, B., JENSEN, H. E., AND LARSEN, K. G. 2002. Reduction and refinement strategies for probabilistic analysis. In *Proceedings of the International Workshop on Performance Modeling and Verification*. 57–76.

DAS, S. 2003. Predicate abstraction. Ph.D. thesis, Stanford University.

DE ALFARO, L. AND ROY, P. 2007. Magnifying-lens abstraction for Markov Decision Processes. In *Proceedings of the International Conference on Computer Aided Verification*. 325–338.

DESHARNAIS, J. 1999a. Labelled markov processes. Ph.D. thesis.

DESHARNAIS, J. 1999b. Logical characterisation of simulation for markov chains. Tech. Rep. CSR-99-8, University of Birmingham. Proceedings of the International Workshop on Performance Modeling and Verification.

Desharnais, J., Gupta, V., Jagadeesan, R., and Panangaden, P. 2000. Approximating labeled Markov processes. In *Proceedings of the IEEE Symposium on Logic in Computer Science*. 95–106.

Dodis, Y. and Khanna, S. 1999. Designing networks with bounded pairwise distance. In *Proceedings of the ACM Symposium on the Theory of Computing*. 750–759.

Fecher, H., Leucker, M., and Wolf, V. 2006. Don't know in probabilistic systems. In *Proceedings of the International SPIN Workshop on Model Checking Software*. 71–88.

Garey, M. and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman.

Glussman, M., Kamhi, G., Mador-Haim, S., Fraer, R., and Vardi, M. 2003. Multiple-Counterexample Guided Iterative Abstraction Refinement: An Industrial Evaluation. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 176–191.

Gurfinkel, A. and Chechik, M. 2006. Why waste a perfectly good abstraction? In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 212–226.

Han, T. and Katoen, J.-P. 2007a. Counterexamples in probabilistic model checking. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 72–86.

Han, T. and Katoen, J.-P. 2007b. Providing evidence of likely being on time — Counterexample generation for CTMC. In *Proceedings of the International Symposium on Automated Technology for Verification and Analysis*.

Hermanns, H., Wachter, B., and Zhang, L. 2008. Probabilistic CEGAR. In *Proceedings of the International Conference on Computer Aided Verification*. Also available as technical report number 33, AVACS, University of Saarland, 2007.

Huth, M. 2004. An abstraction framework for mixed non-deterministic and probabilistic systems. In *Validation of Stochastic Systems: A Guide to Current Research*. 419–444.

Huth, M. 2005. On finite-state approximants for probabilistic computation tree logic. *TCS 346*, 113–134.

Jonsson, B. and Larsen, K. G. 1991. Specification and refinement of probabilistic processes. In *Proceedings of the IEEE Symposium on Logic in Computer Science*. 266–277.

Katoen, J.-P., Klink, D., Leucker, M., and Wolf, V. 2007. Three-valued abstraction for continuous-time markov chains. In *Proceedings of the International Conference on Computer Aided Verification*. 311–324.

Kattenbelt, M., Kwiatkowska, M., Norman, G., and Parker, D. 2009. Abstraction refinement for probabilistic software. In *Proc. 10th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI'09)*, N. Jones and M. Muller-Olm, Eds. LNCS, vol. 5403. Springer, 182–197.

Kemeny, J. and Snell, J. 1976. *Denumerable Markov Chains*. Springer-Verlag.

Kroening, D., Groce, A., and Clarke, E. 2004. Counterexample Guided Abstraction Refinement via Program Execution. In *Proceedings of the International Conference on Formal Engineering Methods*. 224–238.

Kwiatkowska, M., Norman, G., and Parker, D. 2006. Game-based abstraction for markov decision processes. In *Proceedings of the International Conference on Quantitative Evaluation of Systems*. 157–166.

Lakhnech, Y., Bensalem, S., Berezin, S., and Owre, S. 2001. Incremental verification by abstraction. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 98–112.

Majumdar, R., Henzinger, T., Jhala, R., and Sutre, G. 2002. Lazy abstraction. In *Proceedings of the ACM Symposium on Principles of Programming Languages*. 58–70.

McIver, A. and Morgan, C. 2004. *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer.

Monniaux, D. 2005. Abstract interpretation of programs as markov decision processes. *Science of Computer Programming 58*, 179–205.

NORMAN, G. 2004. Analyzing randomized distributed algorithms. In *Validation of Stochastic Systems: A Guide to Current Research*. 384–418.

PAPOULIS, A. AND PILLAI, S. U. 2002. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, New York.

RUTTEN, J. M., KWIATKOWSKA, M., NORMAN, G., AND PARKER, D. 2004. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*. AMS.

SEGALA, R. 2006. Probability and nondeterminism in operational models of concurrency. In *Proceedings of the International Conference on Concurrency Theory*. 64–78.

SEGALA, R. AND LYNCH, N. A. 1994. Probabilistic simulations for probabilistic processes. In *Proceedings of the International Conference on Concurrency Theory*. 481–496.

SEGALA, R. AND LYNCH, N. A. 1995. Probabilistic simulations for probabilistic processes. *Nord. J. Comput. 2,* 2, 250–273.

SHOHAM, S. AND GRUMBERG, O. 2007. A Game-Based framework for CTL Counterexamples and 3-Valued Abstraction-Refinement. *ACM Transactions on Computational Logic 9,* 1.

WACHTER, B., ZHANG, L., AND HERMANNS, H. 2007. Probabilistic model checking modulo theories. In *Proceedings of the International Conference on Quantitative Evaluation of Systems*.

ZHANG, L., HERMANNS, H., EISENBRAND, F., AND JANSEN, D. N. 2007. Flow faster: Efficient decision algorithms for probabilistic simulation. In *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 155–170.

This document is the online-only appendix to:

A Counterexample Guided Abstraction-Refinement Framework for Markov Decision Processes

ROHIT CHADHA and MAHESH VISWANATHAN
Dept. of Computer Science, University of Illinois at Urbana-Champaign

ACM Transactions on Computational Logic, Vol. V, No. N, Month 20YY, Pages 1–0??.

A. SEMANTICS OF PCTL

We first give the semantics of PCTL for MDPs (since every DTMC can be thought of as a MDP, this suffices). Assume that we are given a MDP $\mathcal{M} = (\mathsf{Q}, q_0, \delta, \mathsf{L})$. As described in Section 2, given a scheduler $\mathcal{S} : \mathsf{Q}^+ \to \mathsf{Q}$, let $((\mathsf{Q} \cup \{\bot\})^\omega, E, \mu_{\mathcal{S}})$ be the σ-algebra generated by $\mathcal{M}$ and $\mathcal{S}$. Given $q \in \mathsf{Q}$ and a PCTL formula ψ, the relation $q \Vdash_{\mathcal{M}} \psi$ is defined by induction on ψ in Table I.

$q \Vdash_{\mathcal{M}} \mathsf{tt}$	always
$q \Vdash_{\mathcal{M}} \mathsf{ff}$	never
$q \Vdash_{\mathcal{M}} \neg\psi$	if $q \not\Vdash \psi$
$q \Vdash_{\mathcal{M}} \psi_1 \vee \psi_2$	if $q \Vdash_{\mathcal{M}} \psi_1$ or $q \Vdash_{\mathcal{M}} \psi_2$
$q \Vdash_{\mathcal{M}} \psi_1 \wedge \psi_2$	if $q \Vdash_{\mathcal{M}} \psi_1$ and $q \Vdash_{\mathcal{M}} \psi_2$
$q \Vdash_{\mathcal{M}} \mathcal{P}_{\triangleleft}(\mathsf{X}\psi)$	if for each scheduler $\mathcal{S}$, $\mu_{\mathcal{S}}(\{\pi \in (\mathsf{Q} \cup \{\bot\})^\omega \mid q \Vdash_{\pi,\mathcal{M}} \mathsf{X}\psi\}) \triangleleft p$
$q \Vdash_{\mathcal{M}} \mathcal{P}_{\triangleleft}(\psi_1 \,\mathcal{U}\, \psi_2)$	if for each scheduler $\mathcal{S}$, $\mu_{\mathcal{S}}(\{\pi \in (\mathsf{Q} \cup \{\bot\})^\omega \mid q \Vdash_{\pi,\mathcal{M}} \psi_1 \,\mathcal{U}\, \psi_2\}) \triangleleft p$

Where the relations $q \Vdash_{\pi,\mathcal{M}} (X\psi)$ and $q \Vdash_{\pi,\mathcal{M}} (\psi_1 \,\mathcal{U}\, \psi_2)$ for $\pi = q_0 q_1, \ldots$ are defined as follows–

$q \Vdash_{\pi,\mathcal{M}} (X\psi)$	if $q_0 = q$, $q_1 \neq \bot$ and $q_1 \Vdash_{\mathcal{M}} \psi$
$q \Vdash_{\pi,\mathcal{M}} (\psi_1 \,\mathcal{U}\, \psi_2)$	if $q_0 = q$ and there is a $j \geq 0$ such that $q_i \neq \bot$ for all $i \leq j$, $q_j \Vdash \psi_2$ and $q_r \Vdash \psi_1$ for all $r < j$.

Table I. Semantics for PCTL for DTMCs

B. EXISTENCE OF CANONICAL SIMULATIONS

PROPOSITION 2.8. *Given disjoint MDP's $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ and $\mathcal{M}' = (\mathsf{Q}', q'_{\mathcal{I}}, \delta', \mathsf{L}')$, let $\mathcal{R} \subseteq (\mathsf{Q}+\mathsf{Q}') \times (\mathsf{Q}+\mathsf{Q}')$ be a simulation relation on the direct sum of Q and Q'. Let $\mathcal{R}_1 = \mathcal{R} \cap (\mathsf{Q} \times \mathsf{Q}')$. Then the relation $\mathcal{R}_0 = \mathsf{rel}_{id_{\mathsf{Q}}} \cup \mathcal{R}_1 \cup \mathsf{rel}_{id_{\mathsf{Q}'}}$ is a simulation relation.*

PROOF. Clearly $\mathcal{R}_0$ is reflexive and transitive. Fix $q \in \mathsf{Q}$ and $q' \in \mathsf{Q}'$ such that $q \, \mathcal{R}_0 \, q'$. Please note that by definition $q \, \mathcal{R} \, q'$. Hence, $\mathsf{L}(q) = \mathsf{L}(q')$. We need to show that for any $\mu \in \delta(q)$ there is a $\mu_1 \in \delta'(q')$ such that $\mu \preceq_{\mathcal{R}_0} \mu_1$. Since $\mathcal{R}$ is a simulation relation there is a a $\mu' \in \delta'(q')$ such that $\mu \preceq_{\mathcal{R}} \mu'$. Fix one such μ'. We claim that $\mu \preceq_{\mathcal{R}_0} \mu'$ also.

We need to show that for any $\mathcal{R}_0$-closed set $\mathsf{Q}_0 \subseteq \mathsf{Q} \cup \mathsf{Q}'$, we have that $\mu(\mathsf{Q}_0) \leq \mu'(\mathsf{Q}_0)$. Now let $\mathsf{Q}_1 = \mathsf{Q}_0 \cap \mathsf{Q}$ and $\mathsf{Q}_2 = \mathsf{Q}_0 \cap \mathsf{Q}'$. We have that $\mu(\mathsf{Q}_0) = \mu(\mathsf{Q}_1)$ and $\mu'(\mathsf{Q}_0) = \mu'(\mathsf{Q}_2)$. Thus, we need to show that $\mu(\mathsf{Q}_1) \leq \mu'(\mathsf{Q}_2)$.

Now, consider the set $\mathcal{R}(\mathsf{Q}_1) = \{q_b \in \mathsf{Q} \cup \mathsf{Q}' \mid \exists q_a \in \mathsf{Q}_1 \text{ s.t. } q_a \, \mathcal{R} \, q_b\}$. Now since $\mathcal{R}$ is a preorder, $\mathcal{R}(\mathsf{Q}_1)$ is $\mathcal{R}$-closed and $\mathsf{Q}_1 \subseteq \mathcal{R}(\mathsf{Q}_1)$. From $\mathsf{Q}_1 \subseteq \mathcal{R}(\mathsf{Q}_1)$, we can conclude that $\mu(\mathsf{Q}_1) \leq \mu(\mathcal{R}(\mathsf{Q}_1))$. Also, since $\mathcal{R}(\mathsf{Q}_1)$ is $\mathcal{R}$-closed and $\mu \preceq_{\mathcal{R}} \mu'$ we have that $\mu(\mathcal{R}(\mathsf{Q}_1)) \leq \mu'(\mathcal{R}(\mathsf{Q}_1))$. Hence, we get that $\mu(\mathsf{Q}_1) \leq \mu'(\mathcal{R}(\mathsf{Q}_1))$. Now, please note that $\mu'(\mathcal{R}(\mathsf{Q}_1)) = \mu'(\mathcal{R}(\mathsf{Q}_1) \cap \mathsf{Q}')$. Hence, the result will follow if we can show that $\mathcal{R}(\mathsf{Q}_1) \cap \mathsf{Q}' \subseteq \mathsf{Q}_2$.

Pick $q_b \in \mathcal{R}(\mathsf{Q}_1) \cap \mathsf{Q}'$. We have by definition that $q_b \in \mathsf{Q}'$ and there exists $q_a \in \mathsf{Q}_1$ such that $q_a \, \mathcal{R} \, q_b$. Now, please note that as $\mathsf{Q}_1 \subseteq \mathsf{Q}$, we get $q_a \, \mathcal{R}_0 \, q_b$ (by definition of $\mathcal{R}_0$). Also as $\mathsf{Q}_1 \subseteq \mathsf{Q}_0$, we get that $q_a \in \mathsf{Q}_0$. Since Q_0 is a $\mathcal{R}_0$-closed set, $q_b \in \mathsf{Q}_0$. As $q_b \in \mathsf{Q}'$, we get $q_b \in \mathsf{Q}_2$ also. Since q_b was an arbitrary element of $\mathcal{R}(\mathsf{Q}_1) \cap \mathsf{Q}'$, we can conclude that $\mathcal{R}(\mathsf{Q}_1) \cap \mathsf{Q}' \subseteq \mathsf{Q}_2$. □

C. SIMULATIONS AND PCTL

LEMMA 2.11. *Let $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ be a MDP. For any states $q, q' \in \mathsf{Q}$, $q \preceq q'$ implies that for every liveness formula ψ_L, if $q \Vdash_{\mathcal{M}} \psi_L$ then $q' \Vdash_{\mathcal{M}} \psi_L$ and that for every safety formula ψ_S, if $q' \Vdash_{\mathcal{M}} \psi_S$ then $q \Vdash_{\mathcal{M}} \psi_S$.*

PROOF. The proof is by induction on the length of the safety and liveness formulas. We discuss the case when ψ_S is of the form $\mathrm{Pr}_{\lhd p}(\psi_{L_1} \, \mathcal{U} \, \psi_{L_2})$. Assume that $q' \Vdash_{\mathcal{M}} \psi_S$. We need to show that $q \Vdash_{\mathcal{M}} \psi_S$.

There are two cases to consider.

— The first case is when $q \nVdash_{\mathcal{M}} \psi_{L_1}$. There are two further possibilities.
(1) $q \Vdash_{\mathcal{M}} \psi_{L_2}$. Then (by induction hypothesis), $q' \Vdash_{\mathcal{M}} \psi_{L_2}$ also. Since $q' \Vdash \mathrm{Pr}_{\lhd p}(\psi_{L_1} \mathcal{U} \psi_{L_2})$, we must have that p is 1 and $\lhd$ is $\leq$. Clearly $q \Vdash_{\mathcal{M}} \mathrm{Pr}_{\leq 1}(\psi_{L_1} \mathcal{U} \psi_{L_2})$ also.
(2) $q \nVdash_{\mathcal{M}} \psi_{L_2}$. Now, if $p > 0$, then $q \Vdash_{\mathcal{M}} \mathrm{Pr}_{\lhd p}(\psi_{L_1} \, \mathcal{U} \, \psi_{L_2})$ trivially. If p is 0 then since $q' \Vdash_{\mathcal{M}} \psi_S$, it follows that $\lhd$ must be $\leq$. Now, $q \Vdash_{\mathcal{M}} \mathrm{Pr}_{\leq 0}(\psi_{L_1} \, \mathcal{U} \, \psi_{L_2})$ and the result follows in this case.

— The second case is when $q \Vdash_{\mathcal{M}} \psi_{L_1}$. By induction hypothesis, $q' \Vdash_{\mathcal{M}} \psi_{L_1}$ also. Let $\mathcal{R} \subseteq \mathsf{Q} \times \mathsf{Q}$ be a simulation relation such that $q \, \mathcal{R} \, q'$. Now, let $\mathsf{Q}_0 \subset \mathsf{Q}$ be the set $\{q_0 \in \mathsf{Q} \mid q_0 \Vdash_{\mathcal{M}} \psi_{L_1} \vee \psi_{L_2}\}$. Clearly $q, q' \in \mathsf{Q}_0$. Let δ_0 be the restriction

of δ on Q_0. That is $\delta_0(q) = \{\mu_{|\mathsf{Q}_0} \mid \mu \in \delta(q)\}$. Pick a new label $P_{\psi_{L_2}}$ and for each $q_0 \in Q_0$ let $\mathsf{L}_0(q_0) = \{P_{\psi_{L_2}}\}$ if $q_0 \Vdash_{\mathcal{M}} \psi_{L_2}$ and $\emptyset$ otherwise. Consider the MDP $\mathcal{M}_0 = (\mathsf{Q}_0, q, \delta_0, \mathsf{L}_0)$. It is easy to see that for any $q_0 \in \mathsf{Q}_0$, $q_0 \Vdash_{\mathcal{M}} \psi_S$ iff $q_0 \Vdash_{\mathcal{M}_0} \mathrm{Pr}_{\lhd p}(\Diamond P_{\psi_{L_2}})$.
Let $\mathcal{R}_0$ be the restriction of $\mathcal{R}$ to Q_0, *i.e.*, $\mathcal{R}_0 = \mathcal{R} \cap (Q_0 \times Q_0)$. We make the following observations on $\mathcal{R}_0$.
(1) $q\mathcal{R}_0q'$.
(2) $\mathcal{R}_0$ is reflexive and transitive (which follows from reflexivity and transitivity of $\mathcal{R}$).
(3) We claim that if $\mu \preceq_{\mathcal{R}} \mu'$ then $\mu_{|\mathsf{Q}_0} \preceq_{\mathcal{R}_0} \mu'_{|\mathcal{R}_0}$ also. For this claim it suffices to show that if $A \subseteq \mathsf{Q}_0$ is $\mathcal{R}_0$-closed then A must be $\mathcal{R}$-closed also. In order to demonstrate this, fix a $\mathcal{R}_0$-closed set A_0. Now, let q_1 be any element of Q such that $q_1 \in \mathcal{R}(A_0)$. Then there must be a $q_0 \in \mathsf{Q}_0$ such that $q_0\mathcal{R}q_1$. Now, since $q_0 \in \mathsf{Q}_0$, it must be the case that $q_0 \Vdash \psi_{L_1}$ or $q_0 \Vdash \psi_{L_2}$. In either case, by induction hypothesis, we get that $q_1 \Vdash \psi_{L_1}$ or $q_1 \Vdash \psi_{L_2}$ (since $\mathcal{R}$ is a simulation relation). Therefore $q_1 \in \mathsf{Q}_0$, and hence by definition $q_0\mathcal{R}_0q_1$. Since A_0 is $\mathcal{R}_0$- closed set, we get that $q_1 \in A_0$. Thus, A_0 is $\mathcal{R}$-closed set.
(4) Now, let $q_0 \, \mathcal{R}_0 \, q_0'$. From the above claim and the fact that $\mathcal{R}$ is a simulation, it can be easily shown that for each $\mu_0 \in \delta_0(q_0)$ there is a $\mu_0' \in \delta_0(q_0')$ such that $\mu_0 \preceq_{\mathcal{R}_0} \mu'_{|\mathsf{Q}_0}$.
(5) Similarly we can show that if $q_0 \, \mathcal{R}_0 \, q_0'$ then if $P_{\psi_{L_2}} \in \mathsf{L}_0(q_0)$ then $P_{\psi_{L_2}} \in \mathsf{L}_0(q_0')$ also. Thus the set $Q_1 = \{q_0 \in \mathsf{Q}_0 \mid P_{\psi_{L_2}} \in \mathsf{L}_0(q_0)\}$ is $\mathcal{R}_0$-closed set.
(6) Since $\mathcal{R}_0$ enjoys properties 2, 4 and 5, results of [D'Argenio et al. 2001] imply that for any $q_0\mathcal{R}_0q_0'$, if $q_0' \Vdash_{\mathcal{M}_0} \mathrm{Pr}_{\lhd p}(\Diamond P_{\psi_{L_2}})$ then $q_0 \Vdash_{\mathcal{M}_0} \mathrm{Pr}_{\lhd p}(\Diamond P_{\psi_{L_2}})$ also. The result now follows by observing that $q \, \mathcal{R}_0 \, q'$. □

D. HARDNESS OF FINDING MINIMAL COUNTEREXAMPLES

THEOREM 3.10. *Given a MDP $\mathcal{M}$, a safety formula ψ_S such that $\mathcal{M} \nVdash \psi_S$, and a number $k \leq 2|\mathcal{M}|$, deciding whether there is a counterexample $(\mathcal{E}, \mathcal{R})$ of size $\leq k$ is* NP*-complete.*

PROOF. The problem is in NP because one can guess a counterexample $(\mathcal{E}, \mathcal{R})$ of size k and check if $\mathcal{E}$ violates ψ_S. The hardness result is achieved by a reduction from the exact 3-cover problem [Garey and Johnson 1979] which is formally defined as follows.

> Given a set X such that $|\mathsf{X}| = 3q$ and a collection $\mathcal{C}$ of subsets of X such that for each $C \in \mathcal{C}$, $|C| = 3$, is there an *exact* 3*-cover* for X. In other words, is there a collection of pairwise disjoint sets $\mathcal{B} \subseteq \mathcal{C}$ such that $\mathsf{X} = \cup_{B \in \mathcal{B}} B$.

Before, outlining the proof, it is useful to recall what a 3*-cover* (not necessarily exact) for X is: The collection $\mathcal{B}$ is said to be an 3*-cover*, if $\mathcal{B} \subseteq \mathcal{C}$ is a collection (not necessarily disjoint) such that $\mathsf{X} = \cup_{B \in \mathcal{B}} B$.

Note that without loss of generality we can assume that for each $x \in \mathsf{X}$ there is a $C \in \mathcal{C}$ such that $x \in C$ (if this is not the case, we can simply answer no in polynomial time). Note that $|\mathcal{B}| = q$ for an exact cover. Also note that X has

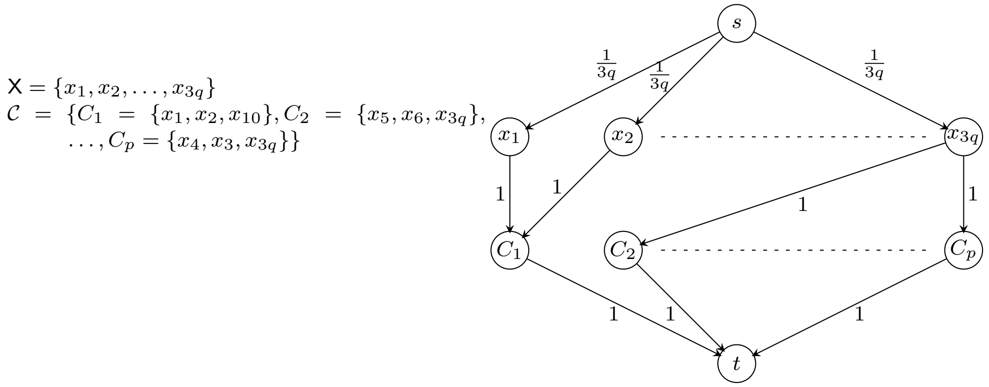
$\mathsf{X} = \{x_1, x_2, \ldots, x_{3q}\}$
$\mathcal{C} = \{C_1 = \{x_1, x_2, x_{10}\}, C_2 = \{x_5, x_6, x_{3q}\}, \ldots, C_p = \{x_4, x_3, x_{3q}\}\}$
s
$\frac{1}{3q}$
$\frac{1}{3q}$
$\frac{1}{3q}$
x_1
x_2
x_{3q}
1
1
1
1
C_1
C_2
C_p
1
1
1
t

Fig. 22. A problem instance of exact 3-cover and the constructed MDP

an exact 3-cover $\mathcal{B} \subseteq \mathcal{C}$ iff there is cover $\mathcal{B}' \subseteq \mathcal{C}$ such that $|\mathcal{B}'| \leq q$. (Actually no collection $\mathcal{B}'$ such that $|\mathcal{B}'| < q$ can cover X, so $\leq$ is mainly a matter of convenience.)

The reduction as follows. We first construct a MDP $\mathcal{M} = (\mathsf{Q}, q_{\mathcal{I}}, \delta, \mathsf{L})$ as follows. For the set of states, we take $\mathsf{Q} = \mathsf{X} \cup \mathcal{C} \cup \{s, t\}$ where s and t are two distinct elements not in $\mathsf{X} \cup \mathcal{C}$. The initial state $q_{\mathcal{I}}$ is taken to be s. There is one probabilistic transition out of s, μ_s, such that $\mu_s(x) = \frac{1}{3q}$ for each $x \in \mathsf{X}$ and $\mu_s(q) = 0$ for all $q \in \mathsf{Q} \setminus \mathsf{X}$. From each $x \in \mathsf{X}$, $\delta(x) = \{\mu_{x,C} \mid x \in C, C \in \mathcal{C}\}$ where $\mu_{x,C}$ assigns probability 1 to C and 0 otherwise. For each C, there is one probabilistic transition out of C, μ_C, which assigns probability 1 to t and is 0 otherwise. There is no transition out of t. Finally, the set of propositions, we will pick a proposition P_q for each $q \in \mathsf{Q}$ and P_q will be true only in the state q. For the safety formula, we take $\psi_S = \mathcal{P}_{<1}(\mathsf{tt}\, \mathcal{U}\, P_t)$. Clearly $\mathcal{M}$ violates ψ_S. The reduction is shown in Figure 22. The result now follows from the following claim.

Claim: X has an exact 3-cover $\mathcal{B} \subset \mathcal{C}$ iff there is a counterexample $(\mathcal{E}, \mathcal{R})$ for $\mathcal{M}$ and ψ_S of size $\leq 2(2 + 4q) + 7q + 3q(1 + \lceil \log 3q \rceil) + 4q$.

Proof of the claim:
($\Rightarrow$) Assume that $\mathcal{B} \subseteq \mathcal{C}$ is an exact 3-cover of X. We have $|\mathcal{B}| = q$. Consider a MDP $\mathcal{M}'$ which is the same as $\mathcal{M}$ except that its states are $\{\bar{q} \mid q \in \mathsf{Q}\}$ instead of Q. Now delete all states $\bar{C}$ of $\mathcal{M}'$ such that $C \notin \mathcal{B}$. Let the resulting MDP be called $\mathcal{E}$ and the set of its states be denoted by $\mathsf{Q}_{\mathcal{E}}$. Note that the $G(\mathcal{E})$ has $2 + 4q$ nodes and $7q$ edges. Furthermore, from the initial state there is a probabilistic transition which assigns probability $\frac{1}{3q}$ to each $\{\bar{x} \mid x \in \mathsf{X}\}$. It takes $1 + \lceil \log 3q \rceil$ bits to represent $\frac{1}{3q}$ (1 for the numerator and $\lceil \log 3q \rceil$ for the denominator). For each $\bar{x}$ such that $x \in \mathsf{X}$ there is a probabilistic transition which assigns probability 1 to $\bar{B}$ where $B \in \mathcal{B}$ is such that $x \in B$. Finally, from each $B \in \mathcal{B}$ there is a probabilistic transition that assigns probability 1 to $\bar{t}$. The size of the MDP $\mathcal{E}$ is seen to be $2 + 4q + 7q + 3q(1 + \lceil \log 3q \rceil) + 4q$. Now, let $\mathcal{R}$ be the relation $\{(\bar{q}, q) \mid q \in \mathsf{Q}_{\mathcal{E}}\}$. Clearly $(\mathcal{E}, \mathcal{R})$ is a counterexample and one can easily check that $|\mathcal{E}, \mathcal{R}| = 2(2 + 4q) + 7q + 3q(1 + \lceil \log 3q \rceil) + 4q$.

($\Leftarrow$) Assume that there is a counterexample $(\mathcal{E}, \mathcal{R})$ of size $\leq 2(2 + 4q) + 7q +$

$3q(1 + \lceil \log 3q \rceil) + 4q$. Thus we have that $\mathcal{E} \preceq \mathcal{M}$, $\mathcal{R}$ is a canonical simulation and $\mathcal{E}$ violates ψ_S. Now note that since every node of $\mathcal{M}$ is labeled by a unique proposition, $\mathcal{R}$ is functional. In other words each state q_1 of $\mathcal{E}$ is related to at most one state of Q. Observe that since the safety formula ψ_S is $\mathcal{P}_{<1}(\mathsf{tt}\, \mathcal{U}\, P_t)$, there is a memoryless scheduler $\mathcal{S}$ such that $\mathcal{E}^{\mathcal{S}}$ violates ψ_S. Let $\mathcal{E}^{\mathcal{S}} = (\mathsf{Q}_{\mathcal{E}}, q_{\mathcal{E}}, \delta_{\mathcal{E}}, \mathsf{L}_{\mathcal{E}})$. For each $q_1 \in \mathsf{Q}_{\mathcal{E}}$, let μ_{q_1} denote the unique probabilistic transition out of $\mathcal{E}^{\mathcal{S}}$.

Note that we have $q_{\mathcal{E}} \mathcal{R} s$. Consider the set $\mathsf{Q}_{\mathsf{X}} = \mathsf{post}(q_{\mathcal{E}}, \mu_{q_{\mathcal{E}}})$. Since $\mathcal{E}$ is simulated by $\mathcal{M}$; it follows that each element of Q_{X} must be labeled by some proposition P_x for some $x \in \mathsf{X}$. Given $x \in \mathsf{X}$, if $\mathsf{Q}_x \subseteq \mathsf{Q}_{\mathsf{X}}$ is the set of states labeled by P_x then we must have $q \mathcal{R} x$ for each $q \in \mathsf{Q}_x$. We also have that $\mu_{q_{\mathcal{E}}}(\mathsf{Q}_x) \leq \frac{1}{3q}$ and $\mathsf{Q}_{x_1} \cap \mathsf{Q}_{x_2} = \emptyset$ for $x_1 \neq x_2$. Now note that the probability of reaching P_t from $q_{\mathcal{E}}$ is 1 in $\mathcal{E}^{\mathcal{S}}$. Hence it must be the case that $\mu_{q_{\mathcal{E}}}(\mathsf{Q}_{\mathsf{X}}) = 1$ and thus $\mathsf{Q}_x \neq \emptyset$ for any $x \in \mathsf{X}$. Therefore $|\mathsf{Q}_{\mathsf{X}}| \geq 3q$ and the total size of the numbers $\{\mu_{q_{\mathcal{E}}}(q) \mid \mu_{q_{\mathcal{E}}}(q) > 0, q \in \mathsf{Q}_{\mathsf{X}}\}$ is at least $|\mathsf{Q}_{\mathsf{X}}|(1 + \lceil \log 3q \rceil)$.

Now given $q \in \mathsf{Q}_x$ consider $\mathsf{post}(q, \mu_q)$. Again as the total probability of reaching P_t is 1 in $\mathcal{E}^{\mathcal{S}}$, $\mathsf{post}(q, \mu_q)$ cannot be empty. Furthermore, as each $q \in \mathsf{Q}_x$ is simulated by x, it follows that each element $q' \in \mathsf{post}(q, \mu_q)$ must be labeled by a single $P_B \in \mathcal{C}$ such that $x \in B$. Let $\mathsf{Q}_{\mathcal{C}} = \cup_{q \in \mathsf{Q}_{\mathsf{X}}} \mathsf{post}(q, \mu_q)$. By the above observations it follows that if we consider the set $\mathcal{B} = \{B \in \mathcal{C} \mid P_B \text{ labels a node in } \mathsf{Q}_{\mathcal{C}}\}$ then $\mathcal{B}$ covers X. Also since every state of $\mathsf{Q}_{\mathcal{C}}$ is labeled by a single proposition, we get $|\mathsf{Q}_{\mathcal{C}}| \geq |\mathcal{B}|$.

We can show by similar arguments that for each $q \in \mathsf{Q}_{\mathcal{C}}$ the set $\mathsf{post}(q, \mu_q)$ is nonempty and each node of $\mathsf{post}(q, \mu_q)$ must be labeled by P_t. Let $\mathsf{Q}_t = \cup_{q \in \mathsf{Q}_{\mathcal{C}}} \mathsf{post}(q, \mu_q)$. We have that $|\mathsf{Q}_t| \geq 1$.

Note that the sets Q_{X}, Q_t and $\mathsf{Q}_{\mathcal{C}}$ are pairwise disjoint and do not contain $q_{\mathcal{E}}$. Hence, the labeled underlying graph of $\mathcal{E}$, $G_\ell(\mathcal{E})$, has at least $1 + |\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}| + |\mathsf{Q}_t|$ vertices. As $\mathcal{R}$ is functional, $\mathcal{R}$ also contains at least $1 + |\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}| + |\mathsf{Q}_t|$ elements. Furthermore, it is easy to see that the underlying graph has at least $2|\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}|$ edges; and the total size of numbers used as probabilities in $\mathcal{E}$ is at least $|\mathsf{Q}_{\mathsf{X}}|(1 + \lceil \log(3t) \rceil) + |\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}|$. Hence the total size of $(\mathcal{E}, R)$ is at least $2(1 + |\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}| + |\mathsf{Q}_t|) + 2|\mathsf{Q}_{\mathsf{X}}| + |\mathsf{Q}_{\mathcal{C}}| + |\mathsf{Q}_{\mathsf{X}}|(1 + \lceil \log(3t) \rceil) + |\mathsf{Q}_{\mathcal{C}}|$. Since $|Q_t| \geq 1$ and $|\mathsf{Q}_{\mathsf{X}}| \geq 3q$; the total size is at least $2(2 + 3q + |\mathsf{Q}_{\mathcal{C}}|) + 6q + |\mathsf{Q}_{\mathcal{C}}| + 3q(1 + \lceil \log(3t) \rceil) + 3q + |\mathsf{Q}_{\mathcal{C}}|$. By hypothesis, the total size is $\leq 2(2 + 4q) + 7q + 3q(1 + \lceil \log 3q \rceil) + 4q$ and we get that $|\mathsf{Q}_{\mathcal{C}}| \leq q$. But $|\mathsf{Q}_{\mathcal{C}}| \geq |\mathcal{B}|$ and hence $|\mathcal{B}| \leq q$. Since $\mathcal{B}$ is a cover; it follows that X must have an exact 3-cover. □

THEOREM 3.11. *Given a MDP $\mathcal{M}$, a safety formula ψ_S and $n = |\mathcal{M}| + |\psi_S|$ such that $\mathcal{M} \not\Vdash \psi_S$. The smallest counterexample for $\mathcal{M}$ and ψ_S cannot be approximated in polynomial time to within $O(2^{\log^{1-\epsilon} n})$ unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{poly \log(n)})$.*

PROOF. The in-approximability follows from a reduction of the Directed Network Steiner Problem [Dodis and Khanna 1999]. Directed Network Steiner Problem is formally defined as follows.

> Given a directed graph G, m pairs $\{s_i, t_i\}_{i=1}^m$ a sub-graph $G' = (V', E')$ of G satisfies the Steiner condition if s_i has path in G' to t_i for all i. The Directed Steiner network problem asks for a sub-graph G' such that G' satisfies the Steiner condition and has the smallest size amongst all

subgraphs of G which satisfy the Steiner condition.

It is shown in [Dodis and Khanna 1999] that the smallest sub-graph cannot be approximated to within $O(2^{\log^{1-\epsilon}(n_g)})$ where n_g is the sum $m+$ size (vertices+edges) of G unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n_g^{poly\log(n_g)})$. Also note that since ϵ is arbitrary the smallest sub-graph cannot be found to within $O(2^{\log^{1-\epsilon}(n_g\log(n_g))})$. (Changing n_g to $n_g\log(n_g)$ does not make a difference $\mathsf{DTIME}(n_g^{poly\log(n_g)})$).

We now give the reduction. Given a graph $G = (V, E)$, let $|V| = n_v$, $|E| = n_e$, $n_g = n_e + n_v$. Recall, that the network steiner problem has m pairs (s_i, t_i). Let n_s be the number of distinct s_i's in (s_i, t_i). In other words n_s is the cardinality of the set $\{s_i \mid 1 \le i \le m\}$. Clearly $n_s \le n_g$. Please note that for the directed network Steiner problem we can assume that n_g is $O(n_e)$.

We construct a DTMC $\mathcal{M}$ with states $V \cup \{s\}$, where s is a new vertex. s is the initial state of $\mathcal{M}$ and has a probabilistic transition μ_s such that $\mu_s(s_i) = \frac{1}{n_g}$ for each $1 \le i \le m$ and $\mu_s(v) = 0$ if $v \notin \{s_i | 1 \le i \le m\}$. Every other state v has a transition μ_v such that $\mu_v(v') = \frac{1}{n_g}$ if $(v, v') \in E$; otherwise $\mu_v(v') = 0$. Finally, we will have as propositions $\{P_v \mid v \in V \cup \{s\}\}$, where the proposition P_v holds at exactly the state v. Since it takes $O(\log(n_g))$ bits to represent $\frac{1}{n_g}$, the size of DTMC $\mathcal{M}$ is easily seen to be $n_v + 1 + n_e + n_s + (n_e + n_s)(O(\log(n_g)))$.

Consider the safety formula $\psi_S = \psi_{S1} \vee \psi_{S2}$ where $\psi_{S_1} = \bigvee_{i=1}^{m} \mathcal{P}_{\le 0}(\Diamond(s_i \wedge (\neg\mathcal{P}_{\le 0}(\Diamond t_i))))$ and $\psi_{S_2} = \bigvee_{i=1}^{n_s} \mathcal{P}_{\le 0}(\mathsf{X}\, s_i)$.

The sum $n_m = |\mathcal{M}| + |\psi_S|$ is easily seen to be $O(n_g log(n_g))$.

Claim:

(1) If G has a sub-graph $G' = (V', E')$ with $|V_1'| = n_1$ and $|E'| = n_2$ such that G' satisfies Steiner condition then $\mathcal{M}$ has a counterexample of size $= 2n_1 + 2 + n_2 + n_s + (n_2 + n_s)(\log(n_g))$.

(2) If the DTMC $\mathcal{M}$ has a counterexample $(\mathcal{E}, \mathcal{R})$ such that $G_\ell(\mathcal{E})$, the underlying labeled graph of $\mathcal{E}$, has n_1+1 vertices and n_2+n_s edges then the graph G has a sub-graph $G' = (V', E')$ with $|V_1'| \le n_1$ and $|E'| \le n_2$ such that G' satisfies the Steiner condition. Furthermore, $|(\mathcal{E}, \mathcal{R})| \ge 2n_1+2+n_2+n_s+(n_2+n_s)(\log(n_g))$.

Proof of the claim:

(1) First assume that G has a sub-graph $G' = (V', E')$ with less than n_1 vertices and less than n_2 edges such that G' satisfies the Steiner condition. Consider the DTMC $\mathcal{M}'$ obtained from $\mathcal{M}$ by restricting the set of states to $V' \cup \{s\}$. Now take an isomorphic copy of $\mathcal{M}'$ with $\{\bar{v} | v \in V'\} \cup \bar{s}$ as the set of states and call it $\mathcal{E}$. Clearly, $\mathcal{E}$ violates ψ_S and the relation $\{(\bar{u}, u) \mid u \in V' \cup s\}$ is a canonical simulation of $\mathcal{E}$ by $\mathcal{M}$. Hence $(\mathcal{E}, \mathcal{R})$ is a counterexample and it is easy to see that $|(\mathcal{E}, \mathcal{R})| \le 2n_1 + 2 + n_2 + n_s + (n_2 + n_s)(\log(n_g))$.

(2) Let $\mathcal{E} = (\mathsf{Q}_\mathcal{E}, q_\mathcal{E}, \delta_\mathcal{E}, \mathsf{L}_\mathcal{E})$ and $G_\ell(\mathcal{E}) = (V', \{E_j'\}_{j=1}^{k})$. Note that since each vertex of $\mathcal{M}$ is labeled by a unique proposition and $\mathcal{R}$ is a canonical simulation, $\mathcal{R}$ must be total and functional (totality is a consequence of the fact that we can remove any nodes of $\mathcal{E}$ that are not reachable from $q_\mathcal{E}$). In other words there is a function $g : \mathsf{Q} \to V \cup \{s\}$ such that $\mathcal{R} = \mathsf{rel}_g$. Again the definition of simulation and the construction of DTMC $\mathcal{M}$ gives us that if $(q_1, q_2) \in \cup_{j=1}^{k} E_j'$,

we must have $(g(q_1), g(q_2)) \in E \cup \{(s, s_i) \mid 1 \leq i \leq m\}$ and $\mu'_{q_1}(q_2) \leq \frac{1}{n_g}$ for any probabilistic transition $\mu'_{q_1} \in \delta_{\mathcal{E}}(q_1)$. From the latter observation, we get that $|(\mathcal{E}, \mathcal{R})| \geq 2n_1 + 2 + n_2 + n_s + (n_2 + n_s)(\log(n_g))$.

Consider the equivalence relation $q_1 \equiv q_2$ defined on Q as $q_1 \equiv q_2$ iff $g(q_1) = g(q_2)$. Let $[q]$ denote the equivalence class of q under $\equiv$. Let $G_2 = \{V'', E''\}$ be the graph such that V'' is the set of equivalence classes under the relation $\equiv$ and $([q_1], [q_2]) \in E''$ if $(q_1, q_2) \in \cup_{j=1}^{k} E'$. Please observe first that G_2 is isomorphic to a subgraph of $(V \cup \{s\}, E \cup \{(s, s_i) \mid 1 \leq i \leq n_s\})$ with the function $h([q]) = g(q)$ witnessing this graph isomorphism. Please note that by the fact that $\mathcal{M}_1$ violates ψ_1, it can be easily shown that there is a path from s_i to t_i in G_2. Also since $\mathcal{M}_1$ violates ψ_2, G_2 contains edges (s, s_i) for each $1 \leq i \leq n_s$. We get by the above observations G must contain a subgraph $G' = (V', E')$ with paths from s_i to t_i for all i such that $|V'_1| \leq n_1$ and $|E'| \leq n_2$. (**End proof the claim.**)

From the above two observations it easily follows that if G has a Steiner subgraph of minimum size $n_{\min_1}$ and $\mathcal{M}$ has a counterexample of minimum size $n_{\min_2}$ then $\frac{n_{\min_2}}{n_{\min_1}} = O(log(n_g))$. Now assume that there is a polynomial time algorithm to compute the minimal counterexample within a factor of $O(2^{\log^{1-\epsilon}(n_m)})$ then this algorithm produces a counterexample of size $k \leq O(2^{\log^{1-\epsilon}(n_m)})n_{\min_2}$. Thus the counterexample size is $\leq O(2^{\log^{1-\epsilon}(n_m)})O(\log(n_g))n_{\min_1}$. From the proof of the part 2 of the above claim it follows that we can extract from the counterexample in polynomial time a Steiner sub-graph of G of size $\leq O(2^{\log^{1-\epsilon}(n_m)})n_{\min_1}$. Now n_m is $O(n_g \log(n_g))$ and thus we have achieved an approximation within $O(2^{\log^{1-\epsilon}(n_g \log(n_g))})$. The result now follows. □

E. CEGAR LOOP CAN BE FASTER THAN DIRECT MODEL-CHECKING

Example E.1. Consider the MDP $\mathcal{M} = (\mathsf{Q}, \delta, q_{\mathcal{I}}, \mathsf{L})$ where

—The set of states Q has $n + 2$ states. We let $\mathsf{Q} = \{q_i \mid 1 \leq i \leq n\} \cup \{q_{\mathsf{fail}}, q_{\mathsf{delay}}\}$.

—$q_{\mathcal{I}} = q_1$.

—δ is defined as follows. For each $1 \leq i \leq n$, $\delta(q_i) = \{\mu_i^1, \mu_i^2\}$ where

(1) $\mu_i^1(q_i) = 0$, $\mu_i^1(q_j) = \frac{1}{4(n-1)}$ if $j \neq i$, $\mu_i^1(q_{\mathsf{delay}}) = \frac{1}{4}$ and $\mu_i^1(q_{\mathsf{fail}}) = \frac{1}{4}$.

(2) $\mu_i^2(q_j) = \frac{1}{3n}$ for each j, $\mu_i^2(q_{\mathsf{delay}}) = 0$ and $\mu_i^2(q_{\mathsf{fail}}) = \frac{1}{3}$.

From state q_{delay}, there is only probabilistic transition and this transition assigns probability 1 to q_{fail}. There is no transition out of state q_{fail}.

—The state q_{fail} is labeled by proposition P.

The property we want to check is $\phi = \mathcal{P}_{\leq \frac{2}{3}}(\Diamond P)$.

Our CEGAR algorithm proceeds as follows.

The initial equivalence classes are $\{q_{\mathsf{fail}}\}$ and $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$. The initial abstraction $\mathcal{M}_1$ has two states– $\{q_{\mathsf{fail}}\}$ and $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$. There are three transitions out of the initial state $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$. The first one assigns probability 1 to $\{q_{\mathsf{fail}}\}$, the second one assigns probability $\frac{1}{2}$ to $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ and $\frac{1}{4}$ to $\{q_{\mathsf{fail}}\}$ and the third one assigns probability $\frac{1}{3}$ to $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ and $\frac{1}{3}$ to $\{q_{\mathsf{fail}}\}$. There is no transition out of $\{q_{\mathsf{fail}}\}$. While constructing this abstraction, we have to do $O(n^2)$ additions (for each state

q_i and each transition out of q_i, we have to add n numbers). Therefore the cost of constructing this abstraction is $O(n^2)$.

Now, we check $\mathcal{M}_1$ against the property ϕ, which turns out to be false and the counterexample generation algorithm gives exactly one counterexample $\mathcal{E}$. The counterxample $\mathcal{E}$ is a DTMC with two states — one corresponding to $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ and the other corresponding to $\{q_{\mathsf{fail}}\}$. The transition out of $\mathsf{Q}\setminus\{q_{\mathsf{fail}}\}$ assigns probability 1 to $\{q_{\mathsf{fail}}\}$ (there is no transition out of $\{q_{\mathsf{fail}}\}$). Please note that the constructed $\mathcal{M}_1$ and the counterexample $\mathcal{E}$ are independent of n. So the time-complexity of model-checking $\mathcal{M}_1$ and generating $\mathcal{E}$ is $O(1)$.

Now, we check whether $\mathcal{E}$ is a valid counter-example. In this case, it turns out that counter-example validity checking is also $O(n^2)$. This is because we have to check if the concrete state q_i simulates the abstract $\mathsf{Q}\setminus\{q_{\mathsf{fail}}\}$ and for this we have to check if any of the two transitions out of q_i in the concrete MDP $\mathcal{M}$ simulates the transition out of $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ in the counter-example, and for checking each transition we have to add n numbers. The check immediately gives that each q_i is indeed not simulated by $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ and thus the counterexample is not valid. The refinement step now splits $\mathsf{Q} \setminus \{q_{\mathsf{fail}}\}$ into two equivalence classes– $\mathsf{Q}_n = \{q_i \mid 1 \leq i \leq n\}$ and $\{q_{\mathsf{delay}}\}$. So in total, the time-complexity of these two steps is $O(n^2)$.

The new abstraction $\mathcal{M}_2$ which occurs as a result of this splitting consists of 3 states— Q_n, $\{q_{\mathsf{delay}}\}$ and $\{q_{\mathsf{fail}}\}$. There are two transitions out of Q_n. The first one assigns probability $\frac{1}{4}$ each to Q_n, $\{q_{\mathsf{delay}}\}$ and $\{q_{\mathsf{fail}}\}$. The second one assigns probability $\frac{1}{3}$ each to Q_n and $\{q_{\mathsf{fail}}\}$ and assigns probability 0 to $\{q_{\mathsf{delay}}\}$. There is only one transition out of $\{q_{\mathsf{delay}}\}$ which assigns probability 1 to $\{q_{\mathsf{fail}}\}$. There is no transition out of q_{fail}. The construction of this new abstraction also takes $O(n^2)$ time.

Now $\mathcal{M}_2$ satisfies ϕ and the CEGAR algorithm ends by saying that the property ϕ is satisfied. Please note again that model-checking $\mathcal{M}_2$ is independent of n.

Therefore, the whole CEGAR algorithm takes $O(n^2)$ time.

On, the other hand were we to do the model-checking by value iteration [Rutten et al. 2004], each step of value iteration involves $O(n^2)$ multiplications and $O(n^2)$ additions. Furthermore, strictly speaking, the value iteration will never terminate (since the convergence to $\frac{2}{3}$ is in the limit). In practice, we stop the value iteration assuming some pre-defined machine error. In that case, the number of value iteration steps depend on this pre-defined machine error and indeeed can be very large.

Remark E.2. Please note that we have used an MDP in the above example. One can also give examples in which model-checking DTMCs for safety properties is also going to be faster using our approach. For example, consider the DTMC $\mathcal{M}'$ which is the same as $\mathcal{M}$ in Example E.1 except that $\delta(q_i)$ consists of only one transtion— μ_i^1. Now, an analysis similar to the one given in Example E.1 will demonstrate that model-checking $\mathcal{M}'$ against the property $\phi = \mathcal{P}_{\leq \frac{2}{3}}(\diamond P)$ is going to be faster than model-checking by value-iteration.