8-2024

Certified policy verification and synthesis for MDPs under distributional reach-avoidance properties

S. AKSHAY

Krishnendu CHATTERJEE

Tobias MEGGENDORFER

Dorde ZIKELIC
Singapore Management University, dzikelic@smu.edu.sg

Citation

AKSHAY, S.; CHATTERJEE, Krishnendu; MEGGENDORFER, Tobias; and ZIKELIC, Dorde. Certified policy verification and synthesis for MDPs under distributional reach-avoidance properties. (2024). *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, Jeju, Korea, 2024 August 3-9*. 3-12.

Available at: https://ink.library.smu.edu.sg/sis_research/9340

Certified Policy Verification and Synthesis for MDPs under Distributional Reach-avoidance Properties

S. Akshay[1] , **Krishnendu Chatterjee**[2] , **Tobias Meggendorfer**[3] , **Đorđe Žikelić**[4]
[1]Indian Institute of Technology Bombay, India
[2]Institute of Science and Technology Austria (ISTA), Austria
[3]Lancaster University Leipzig, Germany
[4]Singapore Management University, Singapore
akshayss@cse.iitb.ac.in, krishnendu.chatterjee@ist.ac.at, tobias@meggendorfer.de, dzikelic@smu.edu.sg

Abstract

Markov Decision Processes (MDPs) are a classical model for decision making in the presence of uncertainty. Often they are viewed as state transformers with planning objectives defined with respect to paths over MDP states. An increasingly popular alternative is to view them as distribution transformers, giving rise to a sequence of probability distributions over MDP states. For instance, reachability and safety properties in modeling robot swarms or chemical reaction networks are naturally defined in terms of probability distributions over states. Verifying such distributional properties is known to be hard and often beyond the reach of classical state-based verification techniques.

In this work, we consider the problems of *certified* policy (i.e. controller) verification and synthesis in MDPs under distributional reach-avoidance specifications. By certified we mean that, along with a policy, we also aim to synthesize a (checkable) certificate ensuring that the MDP indeed satisfies the property. Thus, given the target set of distributions and an unsafe set of distributions over MDP states, our goal is to either synthesize a certificate for a given policy or synthesize a policy along with a certificate, proving that the target distribution can be reached while avoiding unsafe distributions. To solve this problem, we introduce the novel notion of distributional reach-avoid certificates and present automated procedures for (1) synthesizing a certificate for a given policy, and (2) synthesizing a policy together with the certificate, both providing *formal guarantees* on certificate correctness. Our experimental evaluation demonstrates the ability of our method to solve several non-trivial examples, including a multi-agent robot-swarm model, to synthesize certified policies and to certify existing policies.

1 Introduction

State transformer view of MDPs. Markov decision processes (MDPs) are a classical model for decision making in the presence of uncertainty. The prevalent view of MDPs defines them as *state transformers*. Under a policy that resolves non-determinism, an MDP defines a purely stochastic system that performs probabilistic moves from a state to another. This gives rise to a probability space over the set of all runs, i.e. infinite sequences of states, in the MDP [Baier and Katoen, 2008]. MDPs are a central object of study within the AI, planning, and formal methods communities. There is a rich body of work on scalable techniques for reasoning about various properties in MDPs such as discounted-sum and long-run average reward on one hand [Puterman, 1994] and the computation of precise probabilities with which a qualitative objective is satisfied on the other hand, including model checking over expressive logics such as PCTL* [Kwiatkowska *et al.*, 2007].

Distribution transformer view of MDPs. While there is a lot of literature on analyzing MDPs as state transformers, there are several application domains where these approaches fall short. For instance, consider a path planning problem for a robot swarm consisting of an arbitrary number of robots distributed over a set of states. The states are arranged in a topology that has obstacles that must be avoided and a target set of states that must be reached. We want that at least 90% of the robots must reach the target eventually, but at any intermediate step less than 10% must be stuck in an obstacle. In other words, we want to synthesize a policy to control the robot swarm such that a *distributional reach-avoid property* defined with respect to distributions of robots is satisfied. To do this under state-based view, we would need to take the product of state spaces for each agent and define policies over this product space, which would be highly inefficient for systems with many agents.

An alternative to the state-based view is to view MDPs as *distribution transformers*. In this view, starting from some initial probability distribution over MDP states and under a policy that resolves non-determinism, the MDP at each time step induces a new distribution over states, giving rise to a sequence of distributions. Reasoning about this sequence provides a much more natural framework for controlling multi-agent systems under specifications defined in terms of positions of each agent. This allows one to define *distributional policies* that prescribe actions to be performed by each agent based on the current distribution of positions of all other agents, thus providing an effective and compact way for agents to “communicate” their positions to each other. Hence, in contrast to the state-based view, distributional policies are more convenient for controlling multi-agent systems with large number of agents where it suffices to only reason about distributions of their positions and not about positions of each individual agent. In addition to robot-swarms [Baldoni *et al.*, 2008], the distribution transformer view of MDPs also naturally arises in other applications such as bio-chemical reaction networks [Korthikanti *et al.*, 2010; Henzinger *et al.*, 2009] where populations/concentration of cells are distributed across states.

Certification of policies. In this work, we consider the problem of *automated and formal certification of correctness* of policies with respect to distributional reach-avoidance properties in MDPs. This is important in safety-critical applications including robot-swarms or bio-chemical reaction networks, where it is imperative to provide guarantees on correctness prior to policy deployment. In order to ensure safe and correct behavior of such systems, we are interested in computing policies together with *certificates* that serve as formal proofs of correctness of policies and allow for safe and trustworthy policy deployment. We ask the following research questions:

1. *Certificates for policies.* What should be a certificate for formally reasoning about distributional reach-avoidance properties in MDPs? A good certificate should be an object that simultaneously allows *formal* and *automated* reasoning about its correctness.
2. *Formal policy verification with certificates.* Given an MDP and a policy, how do we *compute* such a certificate that formally proves correctness of the policy?
3. *Formal policy synthesis with certificates.* Given an MDP, how do we compute a policy together with a certificate that formally proves its correctness? Can we synthesize not only memoryless but also distributional policies?

Prior work and challenges. Recent years have seen increased interest in formal analysis of MDPs under the distributional view. It was shown [Akshay *et al.*, 2015] that the problem of deciding whether a policy is correct with respect to distributional reachability (and hence distributional reach-avoidance) properties is extremely hard; in fact as hard as the so-called Skolem problem, a long-standing number-theoretic problem whose decidability is unknown [Lipton *et al.*, 2022; Ouaknine and Worrell, 2012]. Moreover, it was shown in [Beauquier *et al.*, 2006] that distributional properties such as reachability and safety cannot be expressed in PCTL*, hence classical model checking methods are not applicable to them. Over the years, the verification community has often studied MDPs under the distributional view, however existing works are either theoretical in nature and focus on decidability of the problem or its variants for different subclasses of MDPs [Kwon and Agha, 2011; Agrawal *et al.*, 2015; Beauquier *et al.*, 2006; Doyen *et al.*, 2014; Akshay *et al.*, 2018] or study specialized logics for reasoning about distributional properties [Agrawal *et al.*, 2015; Beauquier *et al.*, 2006]. Existing automated methods are restricted to distributional safety [Akshay *et al.*, 2023].

To the best of our knowledge, there exists no prior automated method for formal policy verification or synthesis in MDPs with respect to distributional reachability or reach-avoidance properties. Given the Skolem-hardness of the problem, a natural question to ask is how to address this problem in a way which provides formal correctness guarantees while at the same time being practically applicable. Motivated by the success of termination and safety analysis in program verification, we consider an over-approximative approach which may not terminate in all cases but which works in practice while preserving formal guarantees on the correctness of its outputs.

Contributions. Our contributions are as follows:

1. *Certificate for distributional reach-avoidance.* We introduce the novel notion of *distributional reach-avoid certificates*, and show that they provide a sound and complete proof rule for distributional reach-avoidance (Section 4).
2. *Algorithms for formal verification and synthesis.* We develop novel *template-based synthesis algorithms* for the formal synthesis and verification problems with respect to distributional reach-avoidance properties in MDPs.
   (a) First, we develop an algorithm for synthesizing *memoryless policies* along with *affine* distributional reach-avoid certificates. Memoryless policies can be efficiently deployed and executed and are thus preferred in practice. The algorithm is *sound and relatively complete* for deciding the existence of and for computing a memoryless policy and an affine distributional reach-avoid certificate, whenever they exist. While our notion of distributional reach-avoid certificates in Section 4 applies in the general case and provides a sound and complete proof rule for distributional reach-avoidance, our algorithm focuses on the family of *affine* distributional reach-avoid certificates for *practical reasons*, in order to allow for their fully automated and efficient computation. (Section 5).
   (b) While memoryless policies are preferred in practice, they are not always sufficient for solving distributional reach-avoid tasks and one may even require unbounded memory. To that end, we show that it suffices to restrict to the so-called *distributionally memoryless policies* (Section 4) and develop an algorithm for synthesizing them together with affine distributional reach-avoid certificates. The algorithm is sound but incomplete (Section 6).
   (c) Finally, in both cases, we also develop a certification algorithm that proves the correctness of a *given policy* by computing an affine distributional reach-avoidance certificate for it (Sections 5 and 6).
3. *Experimental evaluation.* We implement a prototype of our approach and show that it is able to solve several distributional reach-avoid tasks, including robot-swarms in gridworld environments. Our prototype tool achieves impressive results even when restricted to memoryless strategies, thus showing the effectiveness of our approach as well as the generality of the relative completeness guarantees provided by our first algorithm (Section 7).

Related work. Unlike in the distributional case discussed above, probabilistic reach-avoidance over MDP states is solvable in polynomial time [Baier and Katoen, 2008] and formal policy synthesis for state properties in finite MDPs has been extensively studied. In addition, recent years have seen increased interest in formal policy synthesis [Soudjani *et al.*, 2015; Lavaei *et al.*, 2020; Cauchi and Abate, 2019; Badings *et al.*, 2022; Badings *et al.*, 2023; Xue *et al.*, 2021; Zikelic *et al.*, 2023a; Zikelic *et al.*, 2023b; Grover *et al.*, 2022] and certification of policies [Alshiekh *et al.*, 2018; Jansen *et al.*, 2020; Lechner *et al.*, 2022] for continuous-state MDPs. However, none of these methods are applicable to the *distributional* reach-avoidance problem.

Of works considering distributional properties, the most closely related is the recent work of [Akshay *et al.*, 2023] which considers distributional safety and also proposes a template-based synthesis method. However, our work differs in three important ways. First, our method supports distributional reachability and reach-avoidance. Formal analysis of distributional reachability and reach-avoidance, and even the very definition of a certificate, is significantly more involved and as a result the proofs of our Theorems 1 and 2 are more challenging than the distributional safety setting. Second, we consider both *universal and existential* distributional problems (see Section 3 for definition), whereas they only considered the existential case, i.e. for a single initial distribution. Third, our automated method allows target and safe sets to be specified *both* in terms of strict and non-strict inequalities. This is one of the highlighted open problems in [Akshay *et al.*, 2023, Section 8].

Template-based synthesis was also extensively used for controller synthesis for state properties in deterministic [Jarvis-Wloszek *et al.*, 2003; Ahmadi and Majumdar, 2016] and stochastic systems [Prajna *et al.*, 2007], as well as in program analysis [Gulwani *et al.*, 2008]. In particular, our distributional reach-avoid certificate draws insights from ranking functions for termination [Colón and Sipma, 2001] and invariants for safety analysis in programs [Colón *et al.*, 2003].

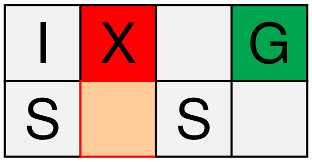
2 Preliminaries

A *probability distribution* on a (countable) set X is a mapping $\mu : X \to [0, 1]$, such that $\sum_{x \in X} \mu(x) = 1$. We write $\mathrm{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$ to denote its *support*, and $\Delta(X)$ to denote the set of all probability distributions on X.

Markov decision processes. A *Markov decision process (MDP)* is a tuple $\mathcal{M} = (S, Act, \delta)$, where S is a finite set of *states*, Act is a finite set of *actions*, overloaded to yield for each state s the set of *available actions* $Act(s) \subseteq Act$, and $\delta : S \times Act \to \Delta(S)$ is a *transition function* that for each state s and (available) action $a \in Act(s)$ yields a probability distribution over successor states. A *Markov chain* is an MDP where each state only has a single available action.

An infinite path in an MDP is a sequence $\rho = s_1 a_1 s_2 a_2 \cdots \in (S \times Act)^\omega$ such that $a_i \in Act(s_i)$ and $\delta(s_i, a_i, s_{i+1}) > 0$ for every $i \in \mathbb{N}$. A *finite path* is a finite prefix of an infinite path. We use ρ_i and ϱ_i to refer to the i-th state in the given (in)finite path, and IPaths_M and FPaths_M for the set of all (in)finite paths of M.

Dynamics of MDPs are defined in terms of policies. A *policy* in an MDP is a map $\pi : \mathsf{FPaths}_{\mathcal{M}} \to \Delta(Act)$, which given a finite path $\varrho = s_0 a_0 s_1 a_1 \ldots s_n$ yields a probability distribution $\pi(\varrho) \in \Delta(Act(s_n))$ on the actions to be taken next. A policy is *memoryless* if the probability distribution over actions only depends on the current state and not on the whole history, i.e. if $\pi(\varrho) = \pi(\varrho')$ whenever ϱ and ϱ' end in the same state. Fixing a policy π and initial distribution μ_0 induces a unique probability measure $\mathsf{Pr}_{\mathcal{M}^\pi, \mu_0}$ over infinite paths of $\mathcal{M}$ [Puterman, 1994].



I	🟥 X		🟩 G
S	🟧	S	

Figure 1: Gridworld example.

MDPs as distribution transformers. MDPs are traditionally viewed as *random generators* of paths, and one investigates the (expected) behaviour of a generated path, i.e. path properties. However, in this work we treat probabilistic systems as *(deterministic) transformers of distributions*.

First, fix a Markov chain M. For a given initial distribution μ_0, we define the distribution at step i by $\mu_i(s) = \mathsf{Pr}_{\mu_0}[\{\rho \in \mathsf{IPaths}_{\mathsf{M}} \mid \rho_i = s\}]$, i.e. the probability to be in state s at step i. We write $\mu_i = \mathsf{M}(\mu_0, i)$ for the i-th distribution and $\mu_1 = \mathsf{M}(\mu_0)$ for the *one-step* application of this transformation. Likewise, we obtain the same notion for an MDP $\mathcal{M}$ combined with a policy π, and write $\mu_i = \mathcal{M}^\pi(\mu_0, i)$, $\mu_1 = \mathcal{M}^\pi(\mu_0)$. In summary, for a given initial distribution, a Markov chain induces a unique stream of distributions, and an MDP provides one for each policy. This naturally invites questions related to this induced stream of distributions. In their path interpretation, queries on MDPs such as *reachability* or *safety*, i.e. asking the probability of reaching or avoiding a set of states, allow for simple, polynomial time solutions [Puterman, 1994; Baier and Katoen, 2008]. However, the corresponding problems in the space of distributions are surprisingly difficult. Our goal is to enable efficient and automated reachability and safety analyses under the distribution transformer interpretation of Markov chains and MDPs. We start with an example.

Example 1. *Consider a* 2×4 *grid as depicted in Figure 1. Let us say that a robot swarm starts at the initial cell marked* I *in the top-left of the grid. From any cell any robot can move horizontally or vertically to an adjacent cell via non-deterministic moves, as long as the adjoining cell is not marked* X*. Cells marked* X *are obstacles that cannot be moved to. Cells marked* S *are stochastic where* 10% *of the robots remain in the cell while remaining* 90% *can move to adjoining cell. Further, orange cells are distributional obstacles, i.e. at any point only* 10% *of all robots in the swarm may be in the set of orange cells. One could visualize this as a narrow path that ensures that only few robots can go through it safely at any point. Finally, there is a goal cell* G*. The problem is then to go from* I *to* G *(at least* 90% *of the swarm must reach* G*), while the dynamics must follow the stochastic constraints in* S *cells, and at the same time avoid obstacles* X *and satisfy distributional constraints in orange cells.*

3 Distributional Reach-avoidance Problems

Consider a *target set* of distributions T and a *safe set* of distributions H, i.e. complement of the set of unsafe distributions. We consider a safe set rather than its complement for the simplicity of the presentation. Distributional reach-avoidance is concerned with verifying or computing an MDP policy under which the induced stream of distributions stays in H until it reaches T. There are several natural variants of the problem, depending on whether policy and initial distribution are given.

Consider an MDP $\mathcal{M} = (S, Act, \delta)$, a set of initial distributions $Init \subseteq \Delta(S)$,$T \subseteq \Delta(S)$, and $H \subseteq \Delta(S)$. Let π be a (memoryless/general) policy for $\mathcal{M}$. Then:

- For $\mu_0 \in Init$, we say that the MDP $\mathcal{M}$ satisfies (T, H)-*reach-avoidance from* μ_0 *under policy* π, if there exists $i \geq 0$ such that $\mathcal{M}^\pi(\mu_0, i) \in T$ and if for all $0 \leq j < i$ we have $\mathcal{M}^\pi(\mu_0, j) \in H$. When μ_0 is fixed, we also say that $\mathcal{M}$ satisfies *unit*-(T, H)*-reach-avoidance under* π.
- We say $\mathcal{M}$ satisfies *existential*-(T, H)*-reach-avoidance under* π if there exists a distribution $\mu_0 \in Init$ such that $\mathcal{M}$ satisfies (T, H)-reach-avoidance from μ_0 under π.
- We say MDP $\mathcal{M}$ satisfies *universal*-(T, H)*-reach - avoidance under* π if for all $\mu_0 \in Init$, $\mathcal{M}$ satisfies (T, U)-reach-avoidance from μ_0 under π.

Existential and universal reach-avoidance are relevant when the input distribution is not precise. While universal reach-avoidance is more restrictive than existential, it is motivated by robustness questions, where we do not precisely know the initial distribution so we ask that all distributions belonging to some uncertainty set behave in a certain way. We are now ready to formally define the problems that we consider:

Distributional Reach-avoidance

Given an MDP $\mathcal{M} = (S, Act, \delta)$, a set of initial distributions $Init \subseteq \Delta(S)$, $\mu_0 \in Init$, a target set $T \subseteq \Delta(S)$, and a safe set $H \subseteq \Delta(S)$:
(i) **The (unit/existential/universal) Policy Certification problem** asks if, given a policy π, there exists a certificate $\mathcal{C}$ which ensures that $\mathcal{M}$ satisfies (unit/existential/universal) (T, H)-reach-avoidance under π and allows automated reasoning about certificate correctness. We refer to the pair $(\pi, \mathcal{C})$ as a *certified policy*.
(ii) **The (unit/existential/universal) Policy Verification problem** asks to, given a policy π, compute a certificate $\mathcal{C}$ such that $(\pi, \mathcal{C})$ as a certified policy.
(iii) **The (unit/existential/universal) Policy Synthesis problem** asks to compute a policy π *together* with a certificate $\mathcal{C}$ such that $(\pi, \mathcal{C})$ as a certified policy.

We note that these problems are not yet fully well defined: Since $\Delta(S)$ is an uncountably infinite space, we need to describe how the sets *Init*, T, and H are represented. As common for continuous scenarios, we restrict to *affine* sets. Concretely, we assume that these sets are specified via a finite number of affine inequalities and are of the form

$$T = \Big\{\mu \in \Delta(S) \mid \bigwedge\nolimits_{j=1}^{N_T} (t_0^j + \sum\nolimits_{i=1}^{n} t_i^j \cdot \mu(s_i)) \bowtie 0\Big\},$$

$$H = \Big\{\mu \in \Delta(S) \mid \bigwedge\nolimits_{j=1}^{N_U} (h_0^j + \sum\nolimits_{i=1}^{n} h_i^j \cdot \mu(s_i)) \bowtie 0\Big\},$$

$$Init = \Big\{\mu \in \Delta(S) \mid \bigwedge\nolimits_{j=1}^{N_{Init}} (a_0^j + \sum\nolimits_{i=1}^{n} a_i^j \cdot \mu(s_i)) \bowtie 0\Big\},$$

where $S = \{s_1, \ldots, s_n\}$ are MDP states, t_j^i, h_i^j, and a_i^j are real-valued constants, $\bowtie \in \{\geq, >\}$, and N_T, N_H, N_{Init} are resp. numbers of affine inequalities defining T, H, and *Init*.

Hardness. Note that Policy Verification and Synthesis problems ask only to compute a certificate for a given policy (or policy and certificate), and not to decide their existence. The reason is that, as it turns out, both decision problems are computationally hard even in the setting of affine sets. This hardness emerges already in the case of (memoryless) policy verification for distributional reachability: Even if M is a Markov chain, $|Init| = 1$, $T = \{\mu \mid \mu(s_1) = \frac{1}{4}\}$, $H = \Delta(S)$, the problem is SKOLEM-hard[1] [Akshay *et al.*, 2015].

Proposition 1. *The decision problem variants of the unit/existential/universal Policy Verification and Synthesis problems for MDPs with respect to distributional reach-avoidance are as hard as the Skolem problem.*

As a result, we cannot expect to obtain an efficient, sound and complete decision procedure for our problem. To overcome this, we focus on asking for certificates and policies of a certain *special* form, and come up with a sound and *relatively* complete procedure to synthesize them, as explained below.

4 Proving Distributional Reach-avoidance

We now consider the Policy Certification problem discussed in Section 3. First, we show that in order to reason about distributional reach-avoidance, it suffices to restrict to the so-called *distributionally memoryless policies*. Second, we introduce our novel certificate for formally proving distributional reach-avoidance, which we call *distributional reach-avoid certificate*. We show that distributional reach-avoid certificates provide a sound and complete proof rule for proving distributional reach-avoidance under distributionally memoryless policies.

Distributionally memoryless policies. Let $\mathcal{M} = (S, Act, \delta)$ be an MDP, $\pi : \mathsf{FPaths}_{\mathcal{M}} \to \Delta(Act)$ a policy and $\mu_0 \in \Delta(S)$ an initial distribution. Let $\mu_0, \mu_1, \ldots$ be the stream of distributions induced by π from μ_0. We say that π is *distributionally memoryless*, if for any initial distribution $\mu_0 \in \Delta(S)$ and for any two finite paths $\varrho = s_0a_0s_1a_1 \ldots s_n$ and $\varrho' = s'_0a'_0s'_1a'_1 \ldots s'_m$ with $\mu_n = \mu_m$, we have $\pi(\varrho) = \pi(\varrho')$. Thus, probability distribution over actions only depends on the current distribution over states and not on the whole history. The following theorem shows that, in order to reason about distributional reach-avoidance, it suffices to restrict to distributionally memoryless policies. The proof of the following theorem can be found in the extended version [Akshay *et al.*, 2024].

Theorem 1. *Let* $T, H \subseteq \Delta(S)$ *be target and safe sets. MDP* $\mathcal{M}$ *satisfies unit/existential/universal-*(T, H)*-reach-avoidance*

[1] Intuitively, the SKOLEM problem asks for a given rational (or integer or real) matrix M, whether there exists $n \in \mathbb{N}$, such that $(M^n)_{1,1} = 0$ [Ouaknine and Worrell, 2012]. This problem (and its variants) has been the subject of intense research over the last 40 years, see e.g. [Lipton *et al.*, 2022]. Yet, quite surprisingly, it still remains open, even for matrices of size 5 and above.

under some policy if and only if there exists a distributionally memoryless policy π such that $\mathcal{M}$ satisfies satisfies unit/existential/universal-(T, H)-reach-avoidance under π.

Sound and complete certificate. Intuitively, given a target set of distributions T and a safe set of distributions H, a distributional (T, H)-reach-avoid certificate under policy π is a pair $\mathcal{C} = (R, I)$ of a *distributional ranking function* R and a *distributional invariant* I. The distributional invariant I is a set of distributions that is required to contain all distributions that are reachable under policy π and also to be contained in H, while the distributional ranking function R is a function that maps distributions over MDP states to reals, which is required to be nonnegative at all distributions contained in I and to decrease by at least 1 after every one-step evolution of the MDP until the target T is reached. We formalize this intuition.

Definition 1 (Distributional reach-avoid certificate)**.** *Let $\mathcal{M}$ be an MDP, $\mu_0 \in \mathit{Init}$ be an initial distribution and π be a policy. A* distributional (T, H)-reach-avoid certificate *for $\mathcal{M}$ from μ_0 under π is a pair (R, I), comprising of a function $R : \Delta(S) \to \mathbb{R}$ and a set of distributions $I \subseteq \Delta(S)$ such that the following conditions hold:*

1. Initial distribution in I. *We have $\mu_0 \in I$.*
2. Inductiveness of I until T. *The set I is* closed *under application of $\mathcal{M}^\pi$ to any non-target distribution contained in I, i.e. $\mathcal{M}^\pi(\mu) \in I$ holds for every $\mu \in I \backslash T$.*
3. Safety. *I is a subset of the safe set H, i.e. $I \subseteq H$.*
4. Nonnegativity of R. *For every $\mu \in I$, we have $R(\mu) \geq 0$.*
5. Strict decrease of R until T. *For every $\mu \in I \backslash T$, we have $R(\mu) \geq R(\mathcal{M}^\pi(\mu)) + 1$.*

Furthermore, a distributional (T, U)-reach-avoid certificate for $\mathcal{M}$ under π is said to be universal*, if it satisfies conditions 2-5 and in addition condition 1 is strengthened to $\mathit{Init} \subseteq I$, i.e. the set I must contain all distributions in Init.*

The following theorem establishes that distributional reach-avoid certificates provide a sound and complete proof rule for proving distributional reach-avoidance in MDPs under distributionally memoryless policies. The intuition behind the proof is as follows. Take the distribution transformer view of MDPs and consider the stream $\mu_0, \mu_1, \mu_2, \dots$ of distributions over MDP states induced by starting in μ_0 and repeatedly applying policy π. Then conditions 1-3 in Definition 1 together ensure that distributions in the stream stay in H at least until T is reached. On the other hand, conditions 4 and 5 in Definition 1 together ensure that a distribution in T must be eventually reached since R cannot be decreased by 1 indefinitely while remaining non-negative. Hence, T is eventually reached while H is not left in the process, and distributional reach-avoid certificates provide a sound proof rule. To prove completeness, we simply let $I = \{\mu_0, \mu_1, \mu_2, \dots\}$ be the stream of induced distributions, k be the smallest index such that $\mu_k \in T$ and define $R(\mu_i) = \max\{0, k - i\}$. One can then verify that (R, I) is indeed a correct distributional reach-avoid certificate. The formal proof can be found in [Akshay *et al.*, 2024].

Theorem 2 (Sound and complete certificates)**.** *Let $\mathcal{M}$ be an MDP, $\mu_0 \in \mathit{Init}$ and π be a distributionally memoryless policy. Then $\mathcal{M}$ satisfies*

1. *unit-(T, H)-reach-avoidance under π iff there exists a distributional (T, H)-reach-avoid certificate for $\mathcal{M}$ from μ_0 under π.*
2. *existential-(T, H)-reach-avoidance under π iff there exists a $\mu_0 \in \mathit{Init}$ and distributional (T, H)-reach-avoid certificate for $\mathcal{M}$ from μ_0 under π.*
3. *universal-(T, H)-reach-avoidance under π iff there exists a universal distributional (T, H)-reach-avoid certificate for $\mathcal{M}$ under π.*

From Proposition 1, it follows that giving a complete procedure for synthesizing (or indeed even checking existence of) distributional certificates is Skolem-hard. Hence in what follows, we provide an automated template-based overapproximation approach that exploits the advances in SMT-solvers to give as an implementable procedure. First, we restrict to memoryless policies in Section 5, then we address the general case of distributionally memoryless policies in Section 6.

5 Algorithm for Memoryless Policies

We now consider the (unit/existential/universal) Policy Synthesis and Verification problems defined in Section 3 under memoryless policies and present our algorithms for solving these problems. Due to space restrictions, in what follows we directly present our algorithm for solving the existential Policy Synthesis problem under memoryless policies. We then explain how this algorithm can be straightforwardly extended to solve the other problems under the memoryless restriction.

Our algorithm simultaneously synthesizes an initial distribution (for the existential problem), a memoryless policy and an *affine* distributional reach-avoid certificate. Restricting to affine distributional reach-avoid certificates (formalized below) ensures efficient and automated computation. While we cannot provide completeness guarantees due to this restriction, we show that the algorithm is sound and *relatively complete*, i.e. it is guaranteed to compute a memoryless policy and an affine distributional reach-avoid certificate when they exist.

A distributional reach-avoid certificate $\mathcal{C} = (R, I)$ is *affine* if it can be specified via affine expressions and inequalities over the distribution space $\Delta(S)$. That is, the distributional ranking function R is of the form $R = r_0 + \sum_{i=1}^n r_i \cdot \mu(s_i)$ and the distributional invariant I is of the form

$$I = \Big\{\mu \in \Delta(S) \mid \bigwedge_{j=1}^{N_I} (b_0^j + \sum_{i=1}^n b_i^j \cdot \mu(s_i)) \geq 0\Big\}$$

where $S = \{s_1, \dots, s_n\}$ are MDP states, b_j^i are real-valued constants and N_I is the number affine inequalities that define I. While the values of variables r_i and b_i^j will be computed by our algorithm, N_I is an algorithm parameter which we refer to as the *template size*. Note that we require all affine inequalities that specify I to be non-strict. This is the technical requirement for our method to provide relative completeness.

Input. The algorithm takes as input an MDP $\mathcal{M} = (S, \mathit{Act}, \delta)$ together with affine sets of initial distributions Init, target distributions T and safe distributions H. It also takes as input the template size parameter N_I.

Algorithm outline. The algorithm employs a template-based synthesis approach and proceeds in three steps. First, it fixes symbolic templates for an initial distribution $\mu_0 \in \mathit{Init}$, a memoryless policy π, and an affine distributional reach-avoid certificate $\mathcal{C} = (R, I)$. Symbolic variables that define the templates are at this stage of *unknown* value. Second, the algorithm collects a system of constraints over the symbolic template variables that encode that $\mu_0 \in \mathit{Init}$, π is a memoryless policy, and $\mathcal{C}$ is a correct distributional reach-avoid certificate. Third, it solves the resulting system of constraints, to get concrete instances of μ_0, π, and $\mathcal{C}$. We detail these steps.

Step 1 – Fixing templates. The algorithm fixes templates for μ_0, π and $\mathcal{C} = (R, I)$:

- *Template for* μ_0. For each MDP state s_i, $1 \leq i \leq n$, the algorithm introduces a symbolic template variable m_i to encode the probability of initially being in s_i.
- *Template for* π. Since the algorithm searches for a memoryless policy, for each state action pair $s_i \in S$ and $a_j \in \mathit{Act}$ we fix a symbolic template variable p_{s_i,a_j} to encode the probability of taking action a_j in state s_i. If $a_j \notin \mathit{Act}(s_i)$, we set $p_{s_i,a_j} = 0$.
- *Template for* R. The template for R is defined by introducing $n + 1$ real-valued symbolic template variables $r_0, \dots, r_n$ and letting $R = r_0 + \sum_{i=1}^{n} r_i \cdot \mu(s_i)$.
- *Template for* I. The template for R is defined by introducing real-valued symbolic template variables b_i^j for each $1 \leq j \leq N_I$ and $0 \leq i \leq n$, with

$$I = \Big\{ \mu \in \Delta(S) \mid \bigwedge\nolimits_{j=1}^{N_I} (b_0^j + \sum\nolimits_{i=1}^{n} b_i^j \cdot \mu(s_i)) \geq 0 \Big\}.$$

Step 2 – Constraint collection. The algorithm now collects constraints over the template variables that together encode that $\mu_0 \in \mathit{Init}$, π is a memoryless policy and $\mathcal{C}$ is a distributional reach avoid certificate. The constraint Φ_{init} encodes that $\mu_0 \in \mathit{Init}$, Φ_π encodes that π is a memoryless policy, whereas $\Phi_1, \dots, \Phi_5$ encode the 5 defining conditions in Definition 1:

- $\Phi_{\text{init}} \equiv (\mu_0 \in \mathit{Init}) \equiv \bigwedge_{j=1}^{N_{\mathit{Init}}} (a_0^j + \sum_{i=1}^{n} a_i^j \cdot m_i) \bowtie 0$, where recall a_i^j's are real constants defining Init.
- $\Phi_\pi \equiv \bigwedge_{i=1}^{n} \Big(\sum_{j=1}^{|\mathit{Act}|} p_{s_i,a_j} = 1 \wedge \bigwedge_{j=1}^{|\mathit{Act}|} (p_{s_i,a_j} \geq 0) \Big)$
- $\Phi_1 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, \mathit{Init}(\mathbf{x}) \Rightarrow I(\mathbf{x})$.
- $\Phi_2 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, I(\mathbf{x}) \Rightarrow I(\text{step}(\mathbf{x}))$, where $\text{step}(\mathbf{x})(x_i) = \sum_{s_k \in S, a_j \in \mathit{Act}(s_k)} p_{s_k,a_j} \cdot \delta(s_k, a_j, s_i) \cdot x_j$ yields the distribution after applying one step of policy π.
- $\Phi_3 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, I(\mathbf{x}) \Rightarrow H(\mathbf{x})$.
- $\Phi_4 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, I(\mathbf{x}) \Rightarrow R(\mathbf{x}) \geq 0$.
- $\Phi_5 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, I(\mathbf{x}) \backslash T(\mathbf{x}) \Rightarrow R(\mathbf{x}) \geq R(\text{step}(\mathbf{x})) - 1$, where $\text{step}(\mathbf{x})$ is defined as above.

Step 3 – Constraint solving. The initial constraint Φ_{init} and the policy constraint Φ_π are purely existentially quantified over template variables. However, $\Phi_1, \dots, \Phi_5$ all contain alternation of an existential quantifier over the symbolic template variables followed by a universal quantifier over the distribution $\mathbf{x} \in \mathbb{R}^n$ over the MDP state probabilities. Quantifier alternation over real-valued arithmetic is in general hard to handle directly and can lead to inefficiency in solvers. To that end, our algorithm first translates these constraints into equisatisfiable *purely existentially quantified* constraints, before feeding the resulting constraints to an off-the-shelf solver.

We begin by noting that $\Phi_1, \Phi_2, \Phi_4, \Phi_5$ can all be expressed as conjunctions of finitely many constraints of the form

$$\begin{aligned} \forall \mathbf{x} \in \mathbb{R}^n .\, (\text{affexp}_1(\mathbf{x}) \bowtie 0) \wedge \cdots \wedge (\text{affexp}_N(\mathbf{x}) \bowtie 0) \\ \Longrightarrow (\text{affexp}(\mathbf{x}) \geq 0), \end{aligned} \tag{1}$$

with each $\text{affexp}_i(\mathbf{x})$ and $\text{affexp}(\mathbf{x})$ being an affine expression over $\mathbf{x}$ whose affine coefficients are either concrete real values or symbolic template variables, and each $\bowtie \in \{\geq, >\}$. The inequalities on the left-hand-side of the implication may be both strict and non-strict, however the inequalities on the right-hand-side of each of $\Phi_1, \Phi_2, \Phi_4, \Phi_5$ are always non-strict, since we assumed that the template for I is specified via non-strict affine inequalities. Now, to remove quantifier alternation, we apply the translation of [Asadi *et al.*, 2021, Corollary 1] which is an extension of Farkas' lemma [Farkas, 1902] that allows strict inequalities on the left-hand-side of the implication (we provide this translation in the extended version of the paper [Akshay *et al.*, 2024]). This allows us to translate each constraint of the form as in Eq. (1) into an equisatisfiable purely existentially quantified system of quadratic constraints with real-valued variables, where variables are either symbolic template variables or fresh symbolic variables introduced in translation.

On the other hand, $\Phi_3 \equiv \forall \mathbf{x} \in \mathbb{R}^n .\, I(\mathbf{x}) \Rightarrow H(\mathbf{x})$ is a conjunction of constraints of the form

$$\begin{aligned} \forall \mathbf{x} \in \mathbb{R}^n .\, (\text{affexp}_1(\mathbf{x}) \geq 0) \wedge \cdots \wedge (\text{affexp}_N(\mathbf{x}) \geq 0) \\ \Longrightarrow (\text{affexp}(\mathbf{x}) \bowtie 0), \end{aligned} \tag{2}$$

since I is specified in terms of non-strict inequalities but H can be specified in terms of both strict and non-strict inequalities. However, since the set of distributions contained in I is topologically closed and bounded as $I \subseteq \Delta(S)$, Eq. (2) is equivalent to the constraint obtained by replacing $\text{affexp}(\mathbf{x}) \bowtie 0$ above by $\text{affexp}(\mathbf{x}) \geq \epsilon$, with $\epsilon > 0$ being a newly introduced symbolic variable. The latter constraint is of the form as in Eq. (1), hence we may again apply the above translation.

Finally, the algorithm feeds the resulting system to an off-the-shelf SMT solver, and any solution gives a concrete instance of initial distribution $\mu_0 \in \mathit{Init}$, memoryless policy π and affine distributional reach-avoid certificate $\mathcal{C}$.

The following theorem (proved in [Akshay *et al.*, 2024]) establishes soundness, relative completeness and an upper bound on the computational complexity of our algorithm. Soundness and relative completeness follow from the fact that the quantifier removal procedure yields an equisatisfiable system of constraints. The PSPACE upper bound follows since the quantifier removal procedure reduces the problem to solving a sentence in the existential first-order theory of the reals.

Theorem 3. Soundness*: If the algorithm returns initial distribution $\mu_0 \in \mathit{Init}$, memoryless policy π, and affine inductive distributional reach-avoid certificate $\mathcal{C}$, then the MDP $\mathcal{M}$ satisfies existential (T, H)-reach-avoidance under π.*

Relative completeness*: If there exists $\mu_0 \in \mathit{Init}$, a memoryless policy π, and an affine distributional (T, H)-reach-avoid certificate $\mathcal{C}$ from μ_0 under π, then there exists a template size $N_I \in \mathbb{N}$ such that μ_0, π, and $\mathcal{C}$ are computed by the algorithm.*

Complexity*: The runtime of the algorithm is in PSPACE in the size of the encoding of the MDP, Init, T, H, and the template size parameter $N_I \in \mathbb{N}$.*

Extension to unit/universal policy synthesis. We now show how the above algorithm can be extended to solve unit and universal Policy Synthesis problems as well. For unit Policy Synthesis, the initial distribution μ_0 is given, hence in Step 1 of the algorithm we do not need to fix a template for it and in Step 2 we remove the constraint Φ_{init}. For universal Policy Synthesis, we now need distributional reach-avoidance to hold from all initial distributions in $Init$. Hence, in Step 1 we again do not need to fix a template for μ_0 whereas in Step 2 we set $\Phi_1 \equiv \forall \mathbf{x} \in \mathbb{R}^n.\ (Init(\mathbf{x}) \Rightarrow I(\mathbf{x}))$. In both cases, the rest of the algorithm proceeds analogously as above. Moreover, as in Theorem 3, both algorithms are sound, relatively complete and of PSPACE runtime in the size of the problem.

Extension to policy verification. The above algorithm is also easily extended to solve the unit/existential/universal Policy Verification problems. In particular, in these problems the memoryless policy π is given and need not be computed. Thus, in Step 1 above we need not fix the templates p_{s_i,a_j} for the policy as these are given concrete values, and in Step 2 we remove the constraint Φ_π. The rest of the algorithms proceed analogously as in the Policy Synthesis case, and as a corollary of Theorem 3 we can show that they are sound, relatively complete and of PSPACE runtime in the size of the problem.

6 Extension to General Policies

We also extend the algorithms for memoryless policies in Section 5 to sound but incomplete procedures for solving unit/existential/universal Policy Synthesis and Verification problems under distributionally memoryless policies. While memoryless policies are preferred in practice since they can be efficiently deployed and executed, it was shown in [Akshay *et al.*, 2023] that there exist MDPs for which memoryless or even bounded memory policies are insufficient for ensuring distributional safety, and the same example shows that bounded memory strategies are insufficient for distributional reach-avoidance. However, by Theorem 1, it always suffices to restrict to distributionally memoryless policies.

The key challenge in extending our algorithms in Section 5 is the design of an appropriate policy template, since now it is insufficient to introduce one template variable p_{s_i,a_j} for each state-action pair as these probabilities may depend on history. However, it turns out that distributionally memoryless policies do admit a sufficiently simple template specified in terms of quotients of symbolic affine expressions over $\mathbf{x}$. In [Akshay *et al.*, 2024], we show how to extend our algorithm for existential Policy Synthesis to compute distributionally memoryless policies. We also provide extensions to unit/universal Policy Synthesis and to Policy Verification. Finally, we prove that the algorithm is sound and runs in PSPACE. However, the algorithm does not provide relative completeness guarantees, since our policy templates are not general and only allow affine expressions over distribution probabilities.

7 Experimental Evaluation

We implemented a prototype of our method in Python 3, using `SymPy` [Meurer *et al.*, 2017] for symbolic expressions and `PySMT` [Gario and Micheli, 2015] to manage SMT solvers. We employ `Yices` 2.6 [Dutertre, 2014] as solving backend. We also evaluated `z3` [de Moura and Bjørner, 2008] and `mathsat` [Cimatti *et al.*, 2013], but `Yices` seemed to consistently perform best. Our experiments were executed on consumer hardware (AMD Ryzen 3600 CPU with 16 GB RAM). Our implementation is publicly available at https://zenodo.org/records/11082466.

As mentioned in the Introduction, we are not aware of any existing automated methods for solving this task in the distributional reach-avoidance setting, thus we do not have a reasonable baseline to compare against. The evaluation of our prototype is aimed at showing that distributional certificates can be found on reasonably sized systems, without heuristics and optimization.

Benchmarks. We evaluate our method on several distributional reach-avoid tasks. Most are modelling a robot swarm in different gridworld environments. Each model requires that at some point in time at least 90% of the robots are in the target set of states, while no more than 10% of robots may be in the unsafe set of states at any intermediate step. Here, we present the following five models: **Running**, i.e. the environment in Example 1, **Double**, a 3x5 grid where the robots start in two different locations and need to reach two goal states, and three grids of size 5x4, 8x8, and 20x10, comprising various transition dynamics as well as limited and forbidden regions. For all grid world models, we consider both (unit) Policy Verification task in which a policy is fixed and needs to be verified, and (unit) Policy Synthesis task where a policy together with a certificate needs to be computed. For all examples, we find the template size $N_I = 1$ to be sufficient. We also present results on **Insulin**, a pharmacokinetics system [Agrawal *et al.*, 2015, Ex. 2], based on [Chadha *et al.*, 2011], and **PageRank** [Agrawal *et al.*, 2015, Fig. 3]. As these two are Markov chains, the verification and synthesis tasks coincide. A more detailed description of models can be found in [Akshay *et al.*, 2024].

Results. Our results are shown in Table 1. Our prototype is able to solve 7/7 Policy Verification tasks with ease, and 3/5 Policy Synthesis tasks. A notable feature of this performance is that it is applicable to robot swarms with arbitrarily many agents, where the model size in the state-based view of MDPs would be intractable for classical model checking tools. This shows that our method is capable of solving highly non-trivial distributional reach-avoidance tasks. Furthermore, our results show that memoryless policies are sufficient in many scenarios.

We believe that the reason behind better scalability of our tool on Policy Verification compared to Policy Synthesis is that the final SMT query in Policy Verification tasks is structurally simpler. In particular, observe that the query for verification of **Grid** 20x10 is larger than the synthesis query for **Grid** 8x8, yet it is solved much faster. To provide further insight, we provide an example SMT query generated for the **Grid** 5x4 example in [Akshay *et al.*, 2024]. Thus, improvements in SMT solvers will improve the scalability of our approach.

Practical observations. Somewhat surprisingly, we often observe that the computationally more expensive part of our implementation is construction of constraints (Steps 1 and 2 in Section 5), especially for the larger certification examples.

Model	$\|S\|$	Act.	Inv.	SMT	Var.	Con.	Ops
Running	7	19	1s	<1s	64	81	556
			2s	<1s	82	105	776
Double	11	30	3s	<1s	88	113	795
			7s	<1s	115	148	1114
Grid 5x4	15	29	2s	<1s	112	145	1018
			6s	1s	133	173	1395
Grid 8x8	32	99	8s	1s	216	284	2086
			19s	T/O	312	410	3144
Grid 20x10	88	280	68s	2s	642	910	6556
			238s	T/O	921	1276	9507
Insulin	5	-	2s	3s	74	88	790
PageRank	5	-	2s	<1s	52	65	571

Table 1: Summary of our experiments. For each model we list, from left to right, the number of (reachable) states, the number of actions, the time used for invariant generation, the time spent by the SMT solver, and the total number of variables, constraints, and operations in the query sent to the solver. The first line for each model is the policy verification query, the second line is the policy synthesis query, where applicable. T/O denotes a timeout after 10 minutes.

We believe that this is due to our naive usage of `SymPy` to extract constraints, since in theory this procedure should run in polynomial time without any complicated data structures. In our prototype implementation, we did not aim for efficient extraction and manipulation of affine expressions. Improvements on this end, e.g. by manually implementing a tailored polynomial representation, would further decrease the runtime of our tool.

We also observe that performance of SMT solvers is highly volatile, with their runtimes sometimes increasing 10- or 100-fold on the same instance, presumably due to running into a bad randomized initialization. Implementing a tighter integration with such solvers and, in particular, providing them with heuristical guidance could further improve performance.

8 Conclusion

We considered the distributional reach-avoidance problem in MDPs, for which we introduced distributional reach-avoid certificates and proposed fully automated template-based synthesis algorithms for solving policy verification and synthesis problems under distributional reach-avoidance. Our work opens several avenues for future work. It would be interesting to consider practical heuristics for template-based synthesis. One could also consider more general distributional properties, ultimately paving the way towards distributional LTL. Finally, our template-based synthesis assumes that the structure of the template, i.e. the number of conjunctive clauses in invariants, is provided a priori. This is a known limitation of many template-based synthesis methods, and exploring effective heuristics for template search is an interesting direction.

Acknowledgements

This work was supported in part by the ERC-2020-CoG 863818 (FoRM-SMArt), the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant, Google Research Award 2023 and the SBI Foundation Hub for Data and Analytics.

References

[Agrawal *et al.*, 2015] Manindra Agrawal, S. Akshay, Blaise Genest, and P. S. Thiagarajan. Approximate verification of the symbolic dynamics of markov chains. *J. ACM*, 62(1):2:1–2:34, 2015.

[Ahmadi and Majumdar, 2016] Amir Ali Ahmadi and Anirudha Majumdar. Some applications of polynomial optimization in operations research and real-time decision making. *Optim. Lett.*, 10(4):709–729, 2016.

[Akshay *et al.*, 2015] S. Akshay, Timos Antonopoulos, Joël Ouaknine, and James Worrell. Reachability problems for markov chains. *Inf. Process. Lett.*, 115(2):155–158, 2015.

[Akshay *et al.*, 2018] S. Akshay, Blaise Genest, and Nikhil Vyas. Distribution-based objectives for markov decision processes. In *LICS*, pages 36–45. ACM, 2018.

[Akshay *et al.*, 2023] S. Akshay, Krishnendu Chatterjee, Tobias Meggendorfer, and Dorde Zikelic. Mdps as distribution transformers: Affine invariant synthesis for safety objectives. In *CAV (3)*, volume 13966 of *Lecture Notes in Computer Science*, pages 86–112. Springer, 2023.

[Akshay *et al.*, 2024] S. Akshay, Krishnendu Chatterjee, Tobias Meggendorfer, and Đorđe Žikelić. Certified policy verification and synthesis for mdps under distributional reach-avoidance properties, 2024.

[Alshiekh *et al.*, 2018] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, pages 2669–2678. AAAI Press, 2018.

[Asadi *et al.*, 2021] Ali Asadi, Krishnendu Chatterjee, Hongfei Fu, Amir Kafshdar Goharshady, and Mohammad Mahdavi. Polynomial reachability witnesses via stellensätze. In *PLDI*, pages 772–787. ACM, 2021.

[Badings *et al.*, 2022] Thom S. Badings, Alessandro Abate, Nils Jansen, David Parker, Hasan A. Poonawala, and Mariëlle Stoelinga. Sampling-based robust control of autonomous systems with non-gaussian noise. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9669–9678. AAAI Press, 2022.

[Badings *et al.*, 2023] Thom S. Badings, Licio Romao, Alessandro Abate, and Nils Jansen. Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In *AAAI*, pages 14701–14710. AAAI Press, 2023.

[Baier and Katoen, 2008] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

[Baldoni *et al.*, 2008] Roberto Baldoni, François Bonnet, Alessia Milani, and Michel Raynal. On the solvability of anonymous partial grids exploration by mobile robots. In *OPODIS*, volume 5401 of *Lecture Notes in Computer Science*, pages 428–445. Springer, 2008.

[Beauquier *et al.*, 2006] Danièle Beauquier, Alexander Moshe Rabinovich, and Anatol Slissenko. A logic of probability with decidable model checking. *J. Log. Comput.*, 16(4):461–487, 2006.

[Cauchi and Abate, 2019] Nathalie Cauchi and Alessandro Abate. Stochy-automated verification and synthesis of stochastic processes. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 258–259, 2019.

[Chadha *et al.*, 2011] Rohit Chadha, Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. Model checking mdps with a unique compact invariant set of distributions. In *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*, pages 121–130. IEEE Computer Society, 2011.

[Cimatti *et al.*, 2013] Alessandro Cimatti, Alberto Griggio, Bastiaan Schaafsma, and Roberto Sebastiani. The MathSAT5 SMT Solver. In Nir Piterman and Scott Smolka, editors, *Proceedings of TACAS*, volume 7795 of *LNCS*. Springer, 2013.

[Colón and Sipma, 2001] Michael Colón and Henny Sipma. Synthesis of linear ranking functions. In *TACAS*, volume 2031 of *Lecture Notes in Computer Science*, pages 67–81. Springer, 2001.

[Colón *et al.*, 2003] Michael Colón, Sriram Sankaranarayanan, and Henny Sipma. Linear invariant generation using non-linear constraint solving. In *CAV*, volume 2725 of *Lecture Notes in Computer Science*, pages 420–432. Springer, 2003.

[de Moura and Bjørner, 2008] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[Doyen *et al.*, 2014] Laurent Doyen, Thierry Massart, and Mahsa Shirmohammadi. Limit synchronization in markov decision processes. In *FoSSaCS*, volume 8412 of *Lecture Notes in Computer Science*, pages 58–72. Springer, 2014.

[Dutertre, 2014] Bruno Dutertre. Yices 2.2. In *CAV*, volume 8559 of *Lecture Notes in Computer Science*, pages 737–744. Springer, 2014.

[Farkas, 1902] Julius Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1902(124):1–27, 1902.

[Gario and Micheli, 2015] Marco Gario and Andrea Micheli. Pysmt: a solver-agnostic library for fast prototyping of smt-based algorithms. In *SMT workshop*, volume 2015, 2015.

[Grover *et al.*, 2022] Kush Grover, Jan Kretínský, Tobias Meggendorfer, and Maximilian Weininger. Anytime guarantees for reachability in uncountable markov decision processes. In *CONCUR*, volume 243 of *LIPIcs*, pages 11:1–11:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[Gulwani *et al.*, 2008] Sumit Gulwani, Saurabh Srivastava, and Ramarathnam Venkatesan. Program analysis as constraint solving. In *PLDI*, pages 281–292. ACM, 2008.

[Henzinger *et al.*, 2009] Thomas A. Henzinger, Maria Mateescu, and Verena Wolf. Sliding window abstraction for infinite markov chains. In *CAV*, volume 5643 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2009.

[Jansen *et al.*, 2020] Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields (invited paper). In *CONCUR*, volume 171 of *LIPIcs*, pages 3:1–3:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Jarvis-Wloszek *et al.*, 2003] Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Some controls applications of sum of squares programming. In *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, volume 5, pages 4676–4681. IEEE, 2003.

[Korthikanti *et al.*, 2010] Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. Reasoning about mdps as transformers of probability distributions. In *QEST*, pages 199–208. IEEE Computer Society, 2010.

[Kwiatkowska *et al.*, 2007] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. Stochastic model checking. In *SFM*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270. Springer, 2007.

[Kwon and Agha, 2011] YoungMin Kwon and Gul A. Agha. Verifying the evolution of probability distributions governed by a DTMC. *IEEE Trans. Software Eng.*, 37(1):126–141, 2011.

[Lavaei *et al.*, 2020] Abolfazl Lavaei, Mahmoud Khaled, Sadegh Soudjani, and Majid Zamani. AMYTISS: parallelized automated controller synthesis for large-scale stochastic systems. In *CAV (2)*, volume 12225 of *Lecture Notes in Computer Science*, pages 461–474. Springer, 2020.

[Lechner *et al.*, 2022] Mathias Lechner, Dorde Zikelic, Krishnendu Chatterjee, and Thomas A. Henzinger. Stability verification in stochastic control systems via neural network supermartingales. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7326–7336. AAAI Press, 2022.

[Lipton *et al.*, 2022] Richard Lipton, Florian Luca, Joris Nieuwveld, Joël Ouaknine, David Purser, and James Worrell. On the skolem problem and the skolem conjecture. In *LICS*, pages 5:1–5:9. ACM, 2022.

[Meurer *et al.*, 2017] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondrej Certík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason Keith Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Stepán Roucka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony M. Scopatz. Sympy: symbolic computing in python. *PeerJ Comput. Sci.*, 3:e103, 2017.

[Ouaknine and Worrell, 2012] Joël Ouaknine and James Worrell. Decision problems for linear recurrence sequences. In *RP*, volume 7550 of *Lecture Notes in Computer Science*, pages 21–28. Springer, 2012.

[Prajna *et al.*, 2007] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans. Autom. Control.*, 52(8):1415–1428, 2007.

[Puterman, 1994] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

[Soudjani *et al.*, 2015] Sadegh Esmaeil Zadeh Soudjani, Caspar Gevaerts, and Alessandro Abate. FAUST ^{2} : Formal abstractions of uncountable-state stochastic processes. In *TACAS*, volume 9035 of *Lecture Notes in Computer Science*, pages 272–286. Springer, 2015.

[Xue *et al.*, 2021] Bai Xue, Renjue Li, Naijun Zhan, and Martin Fränzle. Reach-avoid analysis for stochastic discrete-time systems. In *ACC*, pages 4879–4885. IEEE, 2021.

[Zikelic *et al.*, 2023a] Dorde Zikelic, Mathias Lechner, Thomas A. Henzinger, and Krishnendu Chatterjee. Learning control policies for stochastic systems with reach-avoid guarantees. In *AAAI*, pages 11926–11935. AAAI Press, 2023.

[Zikelic *et al.*, 2023b] Dorde Zikelic, Mathias Lechner, Abhinav Verma, Krishnendu Chatterjee, and Thomas A. Henzinger. Compositional policy learning in stochastic control systems with formal guarantees. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.