Fixed Point Certificates for Reachability and Expected Rewards in MDPs*

Krishnendu Chatterjee[1], Tim Quatmann[2], Maximilian Schäffeler[3], Maximilian Weininger[1(✉)], Tobias Winkler[2], and Daniel Zilken[1,2]

[1] Institute of Science and Technology Austria, Klosterneuburg, Austria
{Krishnendu.Chatterjee,Maximilian.Weininger}@ist.ac.at
[2] RWTH Aachen, Aachen, Germany
{tim.quatmann,tobias.winkler,daniel.zilken}@cs.rwth-aachen.de
[3] Technical University of Munich, Munich, Germany
maximilian.schaeffeler@tum.de

Abstract. The possibility of errors in human-engineered formal verification software, such as model checkers, poses a serious threat to the purpose of these tools. An established approach to mitigate this problem are *certificates*—lightweight, easy-to-check proofs of the verification results. In this paper, we develop novel certificates for model checking of *Markov decision processes* (MDPs) with quantitative reachability and expected reward properties. Our approach is conceptually simple and relies almost exclusively on elementary fixed point theory. Our certificates work for *arbitrary* finite MDPs and can be readily computed with little overhead using standard algorithms. We formalize the soundness of our certificates in Isabelle/HOL and provide a *formally verified certificate checker*. Moreover, we augment existing algorithms in the probabilistic model checker Storm with the ability to produce certificates and demonstrate practical applicability by conducting the first formal certification of the reference results in the Quantitative Verification Benchmark Set.

Keywords: Probabilistic model checking · Markov decision processes · Certificates · Reachability · Expected rewards · Proof assistant

1 Introduction

Markov decision processes (MDPs) [48,7,5] are *the* model for sequential decision making in probabilistic environments. Their many applications [53,32] frequently require computing *reachability probabilities* towards an (un-)desired system state, as well as the *expected rewards* (or costs) accumulated until doing so. *MDP model checking* amounts to computing (approximations of) these

* This project has received funding from the ERC CoG 863818 (ForM-SMArt), the Austrian Science Fund (FWF) 10.55776/COE12, a KI-Starter grant from the Ministerium für Kultur und Wissenschaft NRW, the DFG RTG 378803395 (ConVeY), the EU's Horizon 2020 research and innovation programmes under the Marie Sklodowska-Curie grant agreement Nos. 101034413 (IST-BRIDGE) and 101008233 (MISSION), and the DFG RTG 2236 (UnRAVeL). Experiments were performed with computing resources granted by RWTH Aachen University under project rwth1632.

A. Gurfinkel and M. Heule (Eds.): TACAS 2025, LNCS 15697, pp. 130–151, 2025.
https://doi.org/10.1007/978-3-031-90653-4_7

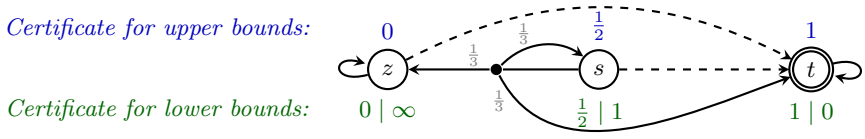

Fig. 1. An MDP with states $S = \{z, s, t\}$, two actions (distinguished by solid and dashed edges), uniform probabilities, and target set $T = \{t\}$. The annotations above and below each state are a certificate for upper and lower bounds on $\mathbb{P}^{\min}(\Diamond T)$, resp.

quantities in a *mathematically rigorous* way, with a formal guarantee of their correctness and precision. Various mature MDP model checking tools such as PRISM [42], mcsta [28], and Storm [33] exist. Figure 1 shows an example MDP.

Who checks the model checker? The possibility of errors in complex, human-engineered formal verification tools is a delicate issue: How *formal* is a verification result produced by an *informal*, i.e., unverified implementation? We highlight four sources of errors: (i) classic implementation bugs, (ii) unintentionally unsound algorithms [8,24], optimizations, and heuristics, (iii) numerical errors due to floating point arithmetic [27], and (iv) errors in third-party back end libraries or tools, e.g., commercial LP solvers [29].

Certifying algorithms [44] are a paradigm for establishing trust in implementations. A certifying algorithm produces a concise, easily verifiable proof—a *certificate*—of its result. The certificate can be checked independently, possibly even by an external, simpler program amenable to formal verification, or by a third party. Formally verified certificate checkers are already employed in tool competitions on software verification [12] or SAT-solving [9]. Existing proposals for certifying MDPs [34,22,35], however, have some drawbacks (detailed further below) hindering wider adoption in the community and its competitions [26,15,3].

The goal of this paper is to establish a new standard for certified MDP model checking, with a focus on applicability and extensibility.

Our contributions towards this goal are as follows:

- We present *fixed point certificates* for two-sided bounds on extremal reachability probabilities (Table 1) and expected rewards. Our certificates are sound and complete for *arbitrary* finite MDPs without structural restrictions.
- We formalize the theory in Isabelle/HOL [46], proving soundness of our certificates, and generate a *formally verified certificate checker implementation*.
- We implement a certifying variant of [29] *Interval Iteration* [8] with floating point arithmetic in Storm [33]. Using this, we give certified reference results for the Quantitative Verification Benchmark Set [32].

Extensibility towards further properties is enabled by our simple, clean theory summarized as four *guiding principles*: (GP1) We characterize the quantities of interest as a *fixed point* of basic, easy-to-evaluate *Bellman-type* operator [11]. The fundamental certification mechanism is to use *fixed point induction* for proving

Table 1. Our reachability certificates. Sound and complete for arbitrary finite MDPs.

Certificate	Condition(s)	*Explanation*
Upper bounds on **minimal** reachability probabilities: $\forall s \in S\colon \mathbb{P}_s^{\min}(\Diamond T) \leq x(s)$ [Proposition 3]		
$x \in [0,1]^S$	$\mathcal{B}^{\min}(x) \leq x$	***min**-Bellman operator **decreases** value of all states*
Upper bounds on **maximal** reachability probabilities: $\forall s \in S\colon \mathbb{P}_s^{\max}(\Diamond T) \leq x(s)$ [Proposition 3]		
$x \in [0,1]^S$	$\mathcal{B}^{\max}(x) \leq x$	***max**-Bellman operator **decreases** value of all states*
Lower bounds on **minimal** reachability probabilities: $\forall s \in S\colon \mathbb{P}_s^{\min}(\Diamond T) \geq x(s)$ [Proposition 4]		
$x \in [0,1]^S$	$\mathcal{B}^{\min}(x) \geq x$	***min**-Bellman operator **increases** value of all states*
$r \in \overline{\mathbb{N}}^S$	$\mathcal{D}^{\max}(r) \leq r$	*r upper bounds **maximal** distances to T*
	$x(s)>0 \implies r(s)<\infty$	*positive reachability necessitates finite distance*
Lower bounds on **maximal** reachability probabilities: $\forall s \in S\colon \mathbb{P}_s^{\max}(\Diamond T) \geq x(s)$ [Proposition 6]		
$x \in [0,1]^S$	$\mathcal{B}^{\max}(x) \geq x$	***max**-Bellman operator **increases** value of all states*
$r \in \overline{\mathbb{N}}^S$	$\mathcal{D}^{\min}_{x\uparrow}(r) \leq r$	*r upper bounds **min.** distances to T via x-**incr. actions***
	$x(s)>0 \implies r(s)<\infty$	*positive reachability necessitates finite distance*

bounds on the least or greatest fixed point. (GP2) We certify *qualitative* reachability properties using *ranking functions* which are amenable to fixed point induction, too. (GP3) As the basic Bellman operators frequently have undesired, spurious fixed points [13,24,39] (often related to *end components* [2]), we consider slight modifications requiring qualitative reachability information which we certify following GP2. (GP4) When GP3 is insufficient or not applicable, we implicitly include a *witness strategy* in our certificate.

Technical challenges still arise in concretely applying these guiding principles. For instance, a key novelty of our paper is a ranking-function type certificate for *not almost sure reachability* (Proposition 2), which is surprisingly involved.

Related work. Closest to our work are the previous proposals for certificates in MDP model checking: [34, Sec. 4] formally verifies a theory of certificates for reachability objectives, which is however limited to upper bounds on maximal and lower bounds on the minimal probabilities. [22] presents so-called "Farkas certificates" for reachability; however, it does not offer a formally verified certificate checker, is limited to MDPs without end components (ECs), and does not address certificate generation explicitly. With similar limitations, [6] provides Farkas certificates for *multiple* reachability or mean payoff objectives, which can be computed via linear programming. [35] suggests lifting the EC assumptions from [22] by certifying the full maximal EC decomposition. In contrast, our certificates are more concise as they handle ECs using at most one ranking function. Further, [35] proposes certificates for expected rewards, but they require the target to be reached almost surely, an assumption we do not have to make.

Witnessing subsystems [54,22,36,6] are an alternative certification paradigm. However, their verification requires more computational effort than the simple, state-wise operations needed for checking Farkas or fixed point certificates. Still, they utilize similar ideas: The backward reachable states in [54, Sec. 3.3.3] essentially use ranking functions, as do the constraints in [37, Sec. 5.2.2].

The term "certifying algorithms" was coined in [38]. Previous work on certificates for other verification problems includes [45,47,41,20,40]. Further, certificates were recently investigated in hardware verification [57,58,59,21] and

approximate model counting [52]. Finally, we mention that *Optimistic Value Iteration* (OVI) [31,4], the supermartingales in [43,1,51], the certificates for probabilistic pushdown systems from [55,56], and a recent strategy synthesis method for infinite MDPs [10] follow fixed point induction principles similar to GP1.

Paper outline. After the background on MDPs and fixed point theory (Section 2), we introduce ranking functions for qualitative reachability (Section 3). Building on this, we discuss quantitative reachability (Section 4, Table 1) and expected rewards (Section 5, [17, Tab. 2]). We explain how to compute certificates (Section 6) and report on experiments (Section 7). Omitted pen-and-paper proofs are in [17, App. C–E]. *All proofs regarding the soundness of the certificates, even standard results from the literature, are formalized in* Isabelle/HOL.

2 Preliminaries

A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (S, Act, P)$ where S is a finite set of *states*, Act is a finite set of *actions*, and $P\colon S \times Act \times S \to [0,1]$ is a *transition probability function* with the property that $\sum_{s' \in S} P(s,a,s') \in \{0,1\}$ for all $s \in S$ and $a \in Act$. For every $s \in S$, the set of *enabled* actions $a \in Act$ for which the above sum equals 1 is written $Act(s)$. It is required that $Act(s) \neq \emptyset$ for all $s \in S$. For $s \in S$ and $a \in Act(s)$, we define the *a-successors* of s as $Post(s,a) = \{s' \in S \mid P(s,a,s') > 0\}$. Notice that our MDPs do not have a distinguished initial state. See Figure 1 for an example MDP.

A (finite-state, discrete-time) *Markov chain* (DTMC) is the special case of an MDP with $|Act(s)| = 1$ for all $s \in S$. A (memoryless and deterministic) *strategy*[4] for an MDP $\mathcal{M} = (S, Act, P)$ is a function $\sigma\colon S \to Act$ such that for all $s \in S$ we have $\sigma(s) \in Act(s)$. We may apply σ to $\mathcal{M}$ to obtain the *induced DTMC* $\mathcal{M}^\sigma = (S, Act, P^\sigma)$ which, intuitively, only retains the actions chosen by σ. Formally, for all $s, s' \in S$ we define $P^\sigma(s, \sigma(s), s') = P(s, \sigma(s), s')$, and $P^\sigma(s,a,s') = 0$ for all $a \neq \sigma(s)$.

Reachability and Expected Rewards. Fix a DTMC (S, Act, P), a *target* set $T \subseteq S$, and a *reward function* $\mathrm{rew}\colon S \to \mathbb{R}_{\geq 0}$. We define two random variables $\Diamond T$ and $\mathrm{rew}^{\Diamond T}$ taking values in $\{0,1\}$ and $\overline{\mathbb{R}}_{\geq 0}$, respectively: For $s_0 s_1 \ldots \in S^\omega$ an infinite path, we set $\Diamond T(s_0 s_1 \ldots) = 1$ if and only if (iff) $\exists i \in \mathbb{N}\colon s_i \in T$. Moreover:

$$\mathrm{rew}^{\Diamond T}(s_0 s_1 \ldots) \;=\; \begin{cases} \sum_{k=0}^{\min\{i \mid s_i \in T\}} \mathrm{rew}(s_k) & \text{if } \exists i \in \mathbb{N}\colon s_i \in T \\ * & \text{else} \end{cases}$$

We consider both options $* = \infty$ and $* = \sum_{k=0}^{\infty} \mathrm{rew}(s_k)$ [19]. We focus on the former in the main body, as it is standard in the literature [7, Def. 10.71] and tool competitions [26,15]; we treat the latter in [17, App. F]. Intuitively, with $* = \infty$, $\mathrm{rew}^{\Diamond T}$ assigns ∞ to paths that never reach T. The other paths receive the sum of rewards collected until reaching T for the first time. Given a state $s \in S$,

[4] Aka. scheduler or policy. We do not define more general strategies as memoryless deterministic suffice for optimizing reachability probabilities and expected rewards.

we define the *reachability probability* $\mathbb{P}_s(\Diamond T)$ from s to T as the expected value (Lebesgue integral) of $\Diamond T$ w.r.t. the probability measure $\mathbb{P}_s$ on infinite paths of the DTMC with initial state fixed to s, see [7, Ch. 10] for the construction of $\mathbb{P}_s$. Similarly, the *expected reward* $\mathbb{E}_s(\mathrm{rew}^{\Diamond T})$ from s to T is the expected value of $\mathrm{rew}^{\Diamond T}$. When rew is clear from context, we write $\mathbb{E}_s(\Diamond T)$ instead of $\mathbb{E}_s(\mathrm{rew}^{\Diamond T})$.

Finally, given an MDP $\mathcal{M} = (S, Act, P)$, a state $s \in S$, a target set $T \subseteq S$, a reward function $\mathrm{rew}\colon S \to \mathbb{R}_{\geq 0}$, and $\mathrm{opt} \in \{\min, \max\}$ we define the *optimal reachability probability* $\mathbb{P}^{\mathrm{opt}}_s(\Diamond T) = \mathrm{opt}_\sigma \mathbb{P}^\sigma_s(\Diamond T)$ and the *optimal expected reward* $\mathbb{E}^{\mathrm{opt}}_s(\mathrm{rew}^{\Diamond T}) = \mathrm{opt}_\sigma \mathbb{E}^\sigma_s(\mathrm{rew}^{\Diamond T})$, where $\mathbb{P}^\sigma_s(\Diamond T)$ and $\mathbb{E}^\sigma_s(\mathrm{rew}^{\Diamond T})$ are the reachability probabilities and expected rewards in the induced DTMC $\mathcal{M}^\sigma$.

Fixed Point Theory. A *partial order* on a set X is a binary relation $\preceq$ that is reflexive, transitive, and antisymmetric; in this case, the tuple $(X, \preceq)$ is called a *poset*. Given a poset $(X, \preceq)$, we call $a \in X$ an *upper bound* on $Y \subseteq X$ if for all $b \in Y$ we have $b \preceq a$. If an upper bound a on Y is minimal among all upper bounds, it is the unique *supremum* (or least upper bound) and denoted $\sup Y$. *Lower bounds* and *infima* (or greatest lower bounds) are defined analogously.

The poset $(X, \preceq)$ is a *complete lattice* if $\sup Y$ and $\inf Y$ exist for every $Y \subseteq X$. Every complete lattice has a least and greatest element $\sup \emptyset$ and $\inf \emptyset$, respectively. The following complete lattices are of interest in this paper:

- $(\overline{\mathbb{N}}, \leq)$ where $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ are the *extended natural numbers* and $\leq$ is the usual order on $\mathbb{N}$ extended by $a \leq \infty$ for all $a \in \overline{\mathbb{N}}$. Notice that for every $Y \subseteq \mathbb{N}$, $\sup Y = \infty$ iff Y is infinite.
- Similarly, $(\overline{\mathbb{R}}_{\geq 0}, \leq)$, with $\overline{\mathbb{R}}_{\geq 0} = \mathbb{R}_{\geq 0} \cup \{\infty\}$ the *extended non-negative reals*, is a complete lattice. For every $Y \subseteq \mathbb{R}_{\geq 0}$, $\sup Y = \infty$ iff Y is unbounded.
- $([0,1], \leq)$, the totally ordered set of real probabilities.
- If $(X, \preceq)$ is an arbitrary complete lattice, then for all sets S, $(X^S, \ddot{\preceq})$ is a complete lattice, where $X^S = \{f \mid f\colon S \to X\}$ is the set of functions from S to X and the partial order $\ddot{\preceq}$ is defined as $f \ddot{\preceq} g \iff \forall s \in S\colon f(s) \preceq g(s)$. In the following, we overload notation and write $\preceq$ instead of $\ddot{\preceq}$. For example, if S is the set of states of an MDP, then we can think of $([0,1]^S, \leq)$ as the poset of "probability vectors" indexed by S, partially ordered entry-wise.

Let $(X, \preceq)$ be a poset. A function $\mathcal{F}\colon X \to X$ is called *monotone* if $\forall a, b \in X\colon a \preceq b \implies \mathcal{F}(a) \preceq \mathcal{F}(b)$. The following is *the* key tool of this paper:

Theorem 1 (Knaster-Tarski). *Let $(X, \preceq)$ be a complete lattice and $\mathcal{F}\colon X \to X$ be monotone. Then, the set of* fixed points *$(\{a \in X \mid \mathcal{F}(a) = a\}, \preceq)$ is also a complete lattice. In particular, $\mathcal{F}$ has a* least *and a* greatest *fixed point given by $\mathrm{lfp}\,\mathcal{F} = \inf\{a \in X \mid \mathcal{F}(a) \preceq a\}$ and, dually, $\mathrm{gfp}\,\mathcal{F} = \sup\{a \in X \mid a \preceq \mathcal{F}(a)\}$. As a consequence, the following* fixed point induction rules *are sound: $\forall a \in X$:*

- $\mathcal{F}(a) \preceq a \implies \mathrm{lfp}\,\mathcal{F} \preceq a$ (fixed point induction)
- $a \preceq \mathcal{F}(a) \implies a \preceq \mathrm{gfp}\,\mathcal{F}$ (fixed point co-induction)

Elements $a \in X$ with $\mathcal{F}(a) \preceq a$ (or $a \preceq \mathcal{F}(a)$) are called *(co-)inductive*.

Guiding Principle 1 (Fixed Point Induction) *We apply the theorem of Knaster-Tarski to monotone operators of the form* $\mathcal{F}\colon X^S \to X^S$*, where* X *is a complete lattice and* S *is finite, to certify upper bounds on* $\operatorname{lfp}\mathcal{F}$ *and lower bounds on* $\operatorname{gfp}\mathcal{F}$*. We call such* $\mathcal{F}$ Bellman-type operators.

Throughout the rest of the paper, we fix an MDP $\mathcal{M} = (S, Act, P)$, a set of target states $T \subseteq S$ and, for Section 5, a reward function $\mathrm{rew}\colon S \to \mathbb{R}_{\geq 0}$. Moreover, we let $\mathrm{opt} \in \{\min, \max\}$ and write $\overline{\min} = \max$ and $\overline{\max} = \min$.

3 Certifying Qualitative Reachability

Most of the certificates presented in the forthcoming Sections 4 and 5 enclose a certificate for a *qualitative* reachability property, e.g., $\mathbb{P}^{\mathrm{opt}}(\Diamond T) > 0$ or $\mathbb{P}^{\mathrm{opt}}(\Diamond T) < 1$. Our approach to this is summarized as follows:

Guiding Principle 2 (Ranking Functions) *To certify qualitative reachability properties, we rely on* ranking functions *formalized via appropriate operators capturing certain* distances *in the MDP when viewed as a graph.*

Definition 1 (Distance Operator). *Let* (S, Act, P)*,* T*, and* opt *be the fixed MDP, target set and optimization direction, respectively. (We omit these quantifications in the rest of the paper.) We define the following* distance operator*:*

$$\mathcal{D}^{\mathrm{opt}}\colon\ \overline{\mathbb{N}}^S \to \overline{\mathbb{N}}^S,\ \mathcal{D}^{\mathrm{opt}}(r)(s) = \begin{cases} 0 & \text{if } s \in T \\ 1 \ + \ \underset{a \in Act(s)}{\mathrm{opt}} \ \min_{s' \in Post(s,a)} r(s') & \text{if } s \in S \setminus T \end{cases}$$

$\mathcal{D}^{\mathrm{opt}}$ is a monotone Bellman-type operator on the complete lattice $(\overline{\mathbb{N}}^S, \leq)$ and thus has a least fixed point by Theorem 1. In fact, we even have the following:

Lemma 1 (Unique Fixed Point). $\mathcal{D}^{\mathrm{opt}}$ *has a unique fixed point.*

Intuitively, if $r = \operatorname{fp}\mathcal{D}^{\min}$, then $r(s)$ represents the length of a shortest path from every state $s \in S$ to T, or $r(s) = \infty$ if T is not reachable from s. For $r = \operatorname{fp}\mathcal{D}^{\max}$, $r(s)$ can be seen as the shortest path in the DTMC induced by a strategy that aims to avoid T or reach it as late as possible. We formalize this intuition in Lemma 2 (using the notation $\overline{\min} = \max$ and $\overline{\max} = \min$), and then in Proposition 1 apply Guiding Principle 1 to certify positive reachability.

Lemma 2. *Let* $r = \operatorname{fp}\mathcal{D}^{\mathrm{opt}}$*. Then for all* $s \in S$*,* $r(s) = \infty \iff \mathbb{P}_s^{\overline{\mathrm{opt}}}(\Diamond T) = 0$.

Proposition 1 (Certificates for $\mathbb{P}^{\mathrm{opt}}(\Diamond T) > 0$**).** *A function* $r \in \overline{\mathbb{N}}^S$ *is called a* valid certificate for positive opt-reachability *if* $\mathcal{D}^{\overline{\mathrm{opt}}}(r) \leq r$*. If* r *is valid, then* $\forall s \in S\colon r(s) < \infty \implies \mathbb{P}_s^{\mathrm{opt}}(\Diamond T) > 0$.

Example 1. Consider the MDP in Figure 1 on page 2. The values on the bottom right of the states constitute a valid certificate r for positive min-reachability. To check that $\mathcal{D}^{\overline{\min}}(r) = \mathcal{D}^{\max}(r) \leq r$ is indeed true, we verify the following:

$$\mathcal{D}^{\max}(r)(s) \;=\; 1 + \max\Big\{ \underbrace{\min\{0,1,\infty\}}_{\text{solid action}}, \quad \underbrace{0}_{\text{dashed action}} \Big\} \;=\; 1+0 \overset{\checkmark}{\leq} 1 \;=\; r(s)\;,$$

and similar for z and t. As $r(s), r(t) < \infty$, we conclude $\mathbb{P}^{\min}_s(\Diamond T), \mathbb{P}^{\min}_t(\Diamond T) > 0$.

Remark 1 (Certificates for $\mathbb{P}^{\text{opt}}(\Diamond T) = 0$). While we do not need this in our paper, it is instructive to notice that with Lemmas 1 and 2 we can also certify *zero reachability probability*: By Knaster-Tarski, any r with $r \leq \mathcal{D}^{\overline{\text{opt}}}(r)$ witnesses $r \leq \text{fp}\, \mathcal{D}^{\overline{\text{opt}}}$, hence if $r(s) = \infty$ for a state s, then $\mathbb{P}^{\text{opt}}_s(\Diamond T) = 0$.

Certificates for non-almost-sure (a.s.) reachability, i.e., $\mathbb{P}^{\text{opt}}(\Diamond T) < 1$, are needed in Section 5. Ranking function-based certificates for this property are—perhaps surprisingly—more involved. In Definition 2 below we define a *complementary distance operator* that captures (approximately) the distance to the states Z from which T is avoided surely, i.e., $\mathbb{P}^{\text{opt}}_s(\Diamond T) = 0$ for all $s \in Z$. By Lemma 2, finite distance to Z witnesses positive $\overline{\text{opt}}$-reachability of Z and thus non-a.s. opt-reachability of T. A major complication is that Z is not given explicitly. We address this by (i) considering the *least* fp, and (ii) letting the operator only increment the distance if two successors do not have the same rank. For this, we use *Iverson bracket* notation: $[\varphi] = 1$ if φ is true; $[\varphi] = 0$, else. Together, (i) and (ii) ensure that $s \in S$ has rank 0 in the lfp if and only if $s \in Z$.

Definition 2 (Complementary Distance Operator). *We define the* complementary distance operator $\overline{\mathcal{D}}^{\text{opt}} \colon \overline{\mathbb{N}}^S \to \overline{\mathbb{N}}^S$, *with*

$$\overline{\mathcal{D}}^{\text{opt}}(r)(s) \;=\; \begin{cases} \infty & \text{if } s \in T \\ \underset{a \in Act(s)}{\text{opt}} \left(\min\limits_{s' \in Post(s,a)} r(s') \;+\; \big[\exists u, v \in Post(s,a) \colon r(u) \neq r(v)\big] \right) & \text{if } s \in S \setminus T \end{cases}$$

Note that unlike the distance operator $\mathcal{D}^{\text{opt}}$ from Definition 1, $\overline{\mathcal{D}}^{\text{opt}}$ *does not have a unique* fp: The constant $r = \vec{\infty}$ is always a trivial fixed point.

Lemma 3. *Let $r = \text{lfp}\, \overline{\mathcal{D}}^{\text{opt}}$. Then for all $s \in S$, $r(s) = \infty \iff \mathbb{P}^{\text{opt}}_s(\Diamond T) = 1$.*

Proposition 2 (Certificates for $\mathbb{P}^{\text{opt}}(\Diamond T) < 1$). *A function $r \in \overline{\mathbb{N}}^S$ is called a* valid certificate for non-a.s. opt-reachability *if $\overline{\mathcal{D}}^{\text{opt}}(r) \leq r$. If r is valid, then $\forall s \in S \colon r(s) < \infty \implies \mathbb{P}^{\text{opt}}_s(\Diamond T) < 1$.*

Remark 2 (Certificates for $\mathbb{P}^{\text{opt}}(\Diamond T) = 1$). Since $\overline{\mathcal{D}}^{\text{opt}}$ does *not* have a unique fp, we cannot use the trick from Remark 1 to certify $\mathbb{P}^{\text{opt}}(\Diamond T) = 1$ with ranking functions. Sections 4.2 and 4.3 present certificates for general lower bounds.

4 Certificates for Quantitative Reachability

This section presents our certificates for bounds on minimal and maximal reachability probabilities (Table 1). They are characterized via a *Bellman operator*:

Definition 3 (Bellman Operator for Reachability). *We define the Bellman operator for reachability* $\mathcal{B}^{\text{opt}}\colon [0,1]^S \to [0,1]^S$ *as usual:*

$$\mathcal{B}^{\text{opt}}(x)(s) \;=\; \begin{cases} 1 & \text{if } s \in T \\ \underset{a \in Act(s)}{\text{opt}} \sum_{s' \in Post(s,a)} P(s,a,s') \cdot x(s') & \text{if } s \in S \setminus T \end{cases}$$

Similar to $\mathcal{D}^{\text{opt}}$ from Section 3, $\mathcal{B}^{\text{opt}}$ is a monotone function on the complete lattice $([0,1]^S, \leq)$. Thus, $\mathcal{B}^{\text{opt}}$ has a least fixed point by Theorem 1.

Theorem 2 ([16, Sec. 3.5]). *For all* $s \in S$, $(\text{lfp}\, \mathcal{B}^{\text{opt}})(s) = \mathbb{P}^{\text{opt}}_s(\Diamond T)$.

We stress that $\mathcal{B}^{\text{opt}}$ has multiple fixed points in general. For instance, $x = \vec{1}$ is always a trivial fixed point. Theorem 2 states that the reachability probabilities are characterized as the *least* fixed point.

4.1 Upper Bounds on Optimal Reachability Probabilities

Following Guiding Principle 1, we obtain the following by Theorem 2:

Proposition 3 (Certificates for Upper Bounds on $\mathbb{P}^{\text{opt}}(\Diamond T)$). *A probability vector* $x \in [0,1]^S$ *satisfying* $\mathcal{B}^{\text{opt}}(x) \leq x$ *is a* valid certificate for upper bounds on opt-reachability. *If* x *is valid, then* $\forall s \in S\colon \mathbb{P}^{\text{opt}}_s(\Diamond T) \leq x(s)$.

Example 2. We verify that the numbers x above the states in Figure 1 on page 2 are a valid certificate for upper bounds on min-reachability: For s we check

$$\mathcal{B}^{\min}(x)(s) \;=\; \min\left\{\tfrac{1}{3}\cdot 0 + \tfrac{1}{3}\cdot\tfrac{1}{2} + \tfrac{1}{3}\cdot 1 \;,\; 1\cdot 1\right\} \;=\; \min\left\{\tfrac{1}{2}, 1\right\} \overset{\checkmark}{\leq} \tfrac{1}{2} \;=\; x(s)\;,$$

and similar for z and t. Thus Proposition 3 yields $\mathbb{P}^{\min}_s(\Diamond T) \leq \frac{1}{2}$. This particular certificate remains valid when changing $x(s)$ to any probability in $[\frac{1}{2}, 1]$. In general, however, increasing individual values may break inductivity.

4.2 Lower Bounds on Minimal Reachability Probabilities

With Theorem 1, we can only certify lower bounds on *greatest* fixed points. Lower bounds on reachability probabilities—which constitute the *least* fixed point of $\mathcal{B}^{\text{opt}}$—are thus more involved. We propose to tackle this situation as follows:

Guiding Principle 3 (Modified Bellman Operators) *We often* modify *a basic Bellman-type operator to restrict its set of fixed points and enforce a certain extremal (i.e., least or greatest) fixed point of interest.*

We now focus on min-reachability first and modify $\mathcal{B}^{\min}$ as follows:

$$\tilde{\mathcal{B}}^{\min}\colon [0,1]^S \to [0,1]^S,\ \tilde{\mathcal{B}}^{\min}(x)(s) = \begin{cases} \mathcal{B}^{\min}(x)(s) & \text{if } \mathbb{P}^{\min}_s(\Diamond T) > 0 \\ 0 & \text{if } \mathbb{P}^{\min}_s(\Diamond T) = 0 \end{cases}$$

Lemma 4 (Unique Fixed Point [7, Thm. 10.109]). *$\tilde{\mathcal{B}}^{\min}$ has a unique fixed point $\operatorname{fp} \tilde{\mathcal{B}}^{\min} = \operatorname{lfp} \mathcal{B}^{\min}$.*

By Lemma 4 and Theorem 2, any probability vector $x \le \tilde{\mathcal{B}}^{\min}(x)$ witnesses that $x(s) \le \mathbb{P}^{\min}_s(\Diamond T)$ for all $s \in S$. However, evaluating $\tilde{\mathcal{B}}^{\min}(x)$ is not straightforward as it requires determining, for each $s \in S$ whether $\mathbb{P}^{\min}_s(\Diamond T) > 0$. Hence, we include an additional certificate for positive reachability from Section 3:

Proposition 4 (Certificates for Lower Bounds on $\mathbb{P}^{\min}(\Diamond T)$).
A tuple of probability vector and ranking function $(x, r) \in [0,1]^S \times \overline{\mathbb{N}}^S$ is a valid certificate for lower bounds on min-reachability *if*

1) $\mathcal{D}^{\max}(r) \le r$, 2) $x \le \mathcal{B}^{\min}(x)$, 3) $\forall s \in S \setminus T\colon x(s) > 0 \implies r(s) < \infty$.

If (x, r) is valid, then $\forall s \in S\colon \mathbb{P}^{\min}_s(\Diamond T) \ge x(s)$.

Example 3. We apply Proposition 4 to the MDP in Figure 1. The pairs $x(v) \mid r(v)$ below each state v constitute a valid certificate (x, r) for lower bounds on min-reachability. Indeed, we have shown in Example 1 that it satisfies Condition 1) $\mathcal{D}^{\max}(r) \le r$. Condition 2) $x \le \mathcal{B}^{\min}(x)$ holds as well; in fact, we even have $x = \mathcal{B}^{\min}(x)$, see Example 2. For the additional Condition 3), notice that s is the only state in $S \setminus T$ with $x(s) > 0$, and that $r(s) < \infty$ holds as required. We conclude that $\mathbb{P}^{\min}_s(\Diamond T) \ge \frac{1}{2}$.

4.3 Lower Bounds on Maximal Reachability Probabilities

Our approach for lower bounds on $\mathbb{P}^{\min}(\Diamond T)$ from Section 4.2 does not immediately extend to max-reachability because $\tilde{\mathcal{B}}^{\max}$ (a modification of $\mathcal{B}^{\max}$ analogous to $\tilde{\mathcal{B}}^{\min}$) does *not* have a unique fixed point in general, see [17, App. D.3] for a concrete counterexample. This problem is caused by end components [2, Def. 3.13]. Towards a solution, we observe that, essentially by definition,

$$\forall s \in S\colon \quad \mathbb{P}^{\max}_s(\Diamond T) \ge x(s) \iff \exists \text{ Strategy } \sigma\colon \mathbb{P}^{\sigma}_s(\Diamond T) \ge x(s)\ .$$

In words, a lower bound on a max-reachability probability is always witnessed by some strategy.[5] Hence we adopt the following:

Guiding Principle 4 (Witness Strategies) *In some cases, especially when progress towards a target is required, it is helpful to certify a witness strategy.*

[5] Dually, an upper bound on a min-reachability probability is also witnessed by a strategy, but our corresponding certificates from Proposition 3 do not rely on this.

Certificates with an Explicit Witness Strategy. Recall from Section 2 that given a strategy $\sigma\colon S \to Act$ for MDP $\mathcal{M}$, we can consider the induced DTMC $\mathcal{M}^\sigma$. We write $\mathcal{B}^\sigma$ for the Bellman operator associated with $\mathcal{M}^\sigma$ (notice that a DTMC is just a special case of an MDP). Further, we let $\tilde{\mathcal{B}}^\sigma$ be the corresponding modified Bellman operator. By Theorem 2 and Lemma 4:

Lemma 5. *$\tilde{\mathcal{B}}^\sigma$ has a unique fixed point $(\mathrm{fp}\,\tilde{\mathcal{B}}^\sigma)(s) = \mathbb{P}^\sigma_s(\Diamond T)$ for all $s \in S$.*

Thus, we can certify lower bounds similar to Proposition 4 (we write $\mathcal{D}^\sigma$ for the distance operator $\mathcal{D}^{\mathrm{opt}}$ in the DTMC induced by σ):

Proposition 5 (Certificates for Lower Bounds on $\mathbb{P}^{\max}(\Diamond T)$ + Strategy). *A triple $(x, r, \sigma) \in [0,1]^S \times \overline{\mathbb{N}}^S \times Act^S$ is a* valid certificate for lower bounds on max-reachability with witness strategy *if*

$$1)\ \mathcal{D}^\sigma(r) \le r, \quad 2)\ x \le \mathcal{B}^\sigma(x), \quad 3)\ \forall s \in S \setminus T\colon x(s) > 0 \implies r(s) < \infty.$$

If (x, r, σ) is valid, then $\forall s \in S\colon \mathbb{P}^{\max}_s(\Diamond T) \ge \mathbb{P}^\sigma_s(\Diamond T) \ge x(s)$.

Certificates without a Witness Strategy. Increasing the size of the certificate by including the strategy can be avoided, as it can be "read off" from the certifying probability vector $x \in [0,1]^S$. To this end, we define the *x-increasing* actions of state $s \in S$: $Act^{\uparrow}_x(s) = \{a \in Act(s) \mid x(s) \le \sum_{s' \in Post(s,a)} P(s,a,s') \cdot x(s')\}$. If $x \le \mathcal{B}^{\max}(x)$, then $Act^{\uparrow}_x(s)$ contains at least one action. Next, we define a variant of the distance operator which only considers x-increasing actions:

$$\mathcal{D}^{\min}_{x\uparrow}\colon \overline{\mathbb{N}}^S \to \overline{\mathbb{N}}^S,\ \mathcal{D}^{\min}_{x\uparrow}(r)(s) = \begin{cases} 0 & \text{if } s \in T \\ 1 + \min\limits_{a \in Act^{\uparrow}_x(s)} \min\limits_{s' \in Post(s,a)} r(s') & \text{if } s \in S \setminus T \end{cases}$$

Proposition 6 (Certificates for Lower Bounds on $\mathbb{P}^{\max}(\Diamond T)$). *A tuple $(x, r) \in [0,1]^S \times \overline{\mathbb{N}}^S$ is a* valid certificate for lower bounds on max-reachability *if*

$$1)\ \mathcal{D}^{\min}_{x\uparrow}(r) \le r, \quad 2)\ x \le \mathcal{B}^{\max}(x), \quad 3)\ \forall s \in S \setminus T\colon x(s) > 0 \implies r(s) < \infty.$$

If (x, r) is valid, then $\forall s \in S\colon \mathbb{P}^{\max}_s(\Diamond T) \ge x(s)$.

5 Certificates for Expected Rewards

We present certificates for bounds on expected rewards in the "$* = \infty$" semantics that assigns infinite reward to paths not reaching T, with the other case in [17, App. F]. We employ the reward variant of the Bellman operator:

Definition 4 (Bellman Operator for Expected Rewards). *We define the Bellman operator for expected rewards $\mathcal{E}^{\mathrm{opt}}\colon \overline{\mathbb{R}}^S_{\ge 0} \to \overline{\mathbb{R}}^S_{\ge 0}$ as follows:*

$$\mathcal{E}^{\mathrm{opt}}(x)(s) \;=\; \begin{cases} 0 & \text{if } s \in T \\ \mathrm{rew}(s) + \mathrm{opt}\limits_{a \in Act(s)} \sum\limits_{s' \in Post(s,a)} P(s,a,s') \cdot x(s') & \text{if } s \in S \setminus T \end{cases}$$

The above definition assumes that multiplication by ∞ *absorbs* positive numbers, i.e., $p \cdot \infty = \infty$ for all $p > 0$, and $a + \infty = \infty + a = \infty$ for all $a \in \overline{\mathbb{R}}_{\geq 0}$.

Again, $\mathcal{E}^{\text{opt}}$ is a monotone function on the complete lattice $(\overline{\mathbb{R}}^S_{\geq 0}, \leq)$ and thus has a least and a greatest fixed point by Theorem 1. Unfortunately, as it turns out, the sought-after expected rewards $\mathbb{E}^{\text{opt}}_s(\Diamond T)$, $s \in S$, are *neither of these two fixed points*. Indeed, $\operatorname{lfp} \mathcal{E}^{\text{opt}}$ corresponds to the expected rewards in the semantics considered in [17, App. F], and $\operatorname{gfp} \mathcal{E}^{\text{opt}}$ is a trivial upper bound assigning ∞ to all states, see the example in Section 5.1.

Remark 3 (Asymmetry and Duality). In Section 4, an asymmetry between upper and lower bounds arose as the reachability probabilities are a *least* fixed point. Further, for the case of maximizing reachability, spurious fixed points occurred and we required a witness strategy to "make progress" towards the targets (the fact that this case requires special treatment of end components is well established in literature, e.g., [31]). For *safety* objectives, where the goal is to avoid a set of bad states, the situation is dual: The safety probabilities are a *greatest* fixed point, so the lower bound case is simple, and when minimizing the upper bound, we require a witness strategy. The $* = \infty$ semantics for expected rewards share some similarities with a safety objective, since the value is maximized (i.e., is infinite) when the target set is avoided. This section thus differs from Section 4 in two ways: (i) Everything is dual, as $* = \infty$ is "safety-like", and (ii) additional complications arise from the trivial greatest fixed point $\operatorname{gfp} \mathcal{E}^{\text{opt}} = \vec{\infty}$, see below.

5.1 Lower Bounds on Optimal Expected Rewards

Due to the absorptive property of multiplication by ∞, $\operatorname{gfp} \mathcal{E}^{\text{opt}}$ may assign ∞ to states that actually have finite value: For instance, in the DTMC in Figure 2, the gfp assigns ∞ to s because $\infty = \text{rew}(s) + \frac{1}{2} \cdot \infty + \frac{1}{2} \cdot 0$, while in fact $\mathbb{E}_s(\Diamond T) = 2 \cdot \text{rew}(s) < \infty$. To address this, we force the values of states that a.s. reach the target to be finite as follows:

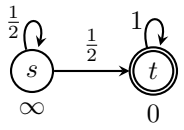

Fig. 2. A DTMC.

Lemma 6. *Let* $x \in \overline{\mathbb{R}}^S_{\geq 0}$ *be such that 1)* $x \leq \mathcal{E}^{\text{opt}}(x)$ *and 2) for all* $s \in S$*:* $\mathbb{P}^{\overline{\text{opt}}}_s(\Diamond T) = 1 \implies x(s) < \infty$ *. Then it holds for all* $s \in S$ *that* $x(s) \leq \mathbb{E}^{\text{opt}}_s(\Diamond T)$.

Intuitively, Lemma 6 requires that a lower bound on $\mathbb{E}^{\min}_s(\Diamond T)$ can only be infinite if T cannot be reached a.s., i.e. $\mathbb{P}^{\max}_s(\Diamond T) < 1$ (dually for $\mathbb{E}^{\max}$). Combining Lemma 6 and a certificate for *non-a.s. reachability* (Section 3) yields:

Proposition 7 (Certificates for Lower Bounds on $\mathbb{E}^{\text{opt}}(\Diamond T)$**).** *A tuple* $(x, r) \in \overline{\mathbb{R}}^S_{\geq 0} \times \overline{\mathbb{N}}^S$ *is a* valid certificate for lower bounds on opt-exp. rewards *if*

$$1)\ \overline{\mathcal{D}^{\text{opt}}}(r) \leq r, \quad 2)\ x \leq \mathcal{E}^{\text{opt}}(x), \quad 3)\ \forall s \in S\colon x(s) = \infty \implies r(s) < \infty.$$

If (x, r) *is valid, then* $\forall s \in S\colon \mathbb{E}^{\text{opt}}_s(\Diamond T) \geq x(s)$.

5.2 Upper Bounds on Maximal Expected Rewards

Next we focus on upper bounds on maximal expected rewards. Using Guiding Principle 3 as for *lower* bounds on *minimal* reachability probabilities (Section 4.2), we obtain such certificates via a modified Bellman operator:

$$\tilde{\mathcal{E}}^{\max}\colon \overline{\mathbb{R}}^S_{\geq 0} \to \overline{\mathbb{R}}^S_{\geq 0}, \quad \tilde{\mathcal{E}}^{\max}(x)(s) \;=\; \begin{cases} \mathcal{E}^{\max}(x)(s) & \text{if } \mathbb{P}^{\min}_s(\Diamond T) > 0 \\ \infty & \text{if } \mathbb{P}^{\min}_s(\Diamond T) = 0 \end{cases}$$

Lemma 7. *For all* $s \in S$, $(\operatorname{lfp} \tilde{\mathcal{E}}^{\max})(s) = \mathbb{E}^{\max}_s(\Diamond T)$.

We stress that unlike $\tilde{\mathcal{B}}^{\min}$ from Section 4.2, $\tilde{\mathcal{E}}^{\max}$ does not have a *unique* fixed point, see Figure 2. Nonetheless, with Lemma 7, Guiding Principle 1, and the certificates for positive reachability from Proposition 1, we obtain:

Proposition 8 (Certificates for Upper Bounds on $\mathbb{E}^{\max}(\Diamond T)$). *A tuple* $(x, r) \in \overline{\mathbb{R}}^S_{\geq 0} \times \overline{\mathbb{N}}^S$ *is a* valid certificate for upper bounds on max-exp. rewards *if*

1) $\mathcal{D}^{\max}(r) \leq r$, 2) $\mathcal{E}^{\max}(x) \leq x$, 3) $\forall s \in S\colon x(s) < \infty \implies r(s) < \infty$.

If (x, r) *is valid, then* $\forall s \in S\colon \mathbb{E}^{\max}_s(\Diamond T) \leq x(s)$.

5.3 Upper Bounds on Minimal Expected Rewards

Our approach for this case parallels the one for *lower* bounds on *maximal* reachability probabilities from Section 4.3. The modified Bellman operator $\tilde{\mathcal{E}}^{\min}$ (defined analogous to $\tilde{\mathcal{E}}^{\max}$ from above) does *not* characterize the minimal expected rewards as its least fixed point. The problem are, again, end components, see [17, App. E.5] for a counter-example. Following Guiding Principle 4 and Section 4.3, we can, however, certify upper bounds on $\mathbb{E}^{\min}(\Diamond T)$ by including a witness strategy (see [17, App. E.6]).

As with lower bounds on max-reachability, it is also possible to avoid this explicit witness strategy: We define the *x-decreasing actions* of s as $Act^{\downarrow}_x(s) = \{a \in Act(s) \mid x(s) \geq \operatorname{rew}(s) + \sum_{s' \in Post(s,a)} P(s,a,s') \cdot x(s')\}$. If $\mathcal{E}^{\min}(x) \leq x$, then $Act^{\downarrow}_x(s) \neq \emptyset$. We define a distance operator with $\mathcal{D}^{\min}_{x\downarrow}$ that only considers x-decreasing actions completely analogous to $\mathcal{D}^{\min}_{x\uparrow}$ from Section 4.3.

Proposition 9 (Certificates for Upper Bounds on $\mathbb{E}^{\min}(\Diamond T)$). *A tuple* $(x, r) \in \overline{\mathbb{R}}^S_{\geq 0} \times \overline{\mathbb{N}}^S$ *is a* valid certificate for upper bounds on min-exp. rewards *if*

1) $\mathcal{D}^{\min}_{x\downarrow}(r) \leq r$, 2) $\mathcal{E}^{\min}(x) \leq x$, 3) $\forall s \in S\colon x(s) < \infty \implies r(s) < \infty$.

If (x, r) *is valid, then* $\forall s \in S\colon \mathbb{E}^{\min}_s(\Diamond T) \leq x(s)$.

6 Computing Certificates

In Sections 3 to 5 we described *what* certificates are and discussed their verification conditions. We now elaborate on *how to compute* certificates. To this

end, we first discuss computation of (co-)inductive value vectors x and then focus on the ranking functions r required by some certificates (see Table 1). We stress that a sound certificate checker detects any wrong results produced by buggy implementations of the methods discussed in this section. Indeed, during implementation of the certificate computation algorithms in Storm, checking the certificates helped finding and resolving implementation bugs.

As we enter the realm of numeric computation, some remarks are in order. For computational purposes we assume that the transitions probabilities are *rational numbers*, i.e., fractions of integers. Moreover:

Our goal is to compute a certificate with a rational value vector x and to check it with exact, arbitrary precision rational number arithmetic.

Certificates via Exact Algorithms. The conceptually easiest certifying MDP model checking algorithm is to compute the rational reachability probabilities or expected rewards *exactly*. The resulting value vector is both inductive and coinductive. Thus, exact algorithms yield a certificate essentially as a by-product. We refer to [29,30] for an in-depth comparison of exact algorithms based on *Policy Iteration* (PI), *Rational Search* (RS), and *Linear Programming* (LP). The practically most efficient algorithm is PI with exact LU decomposition as linear equation solver; see [30, Secs. 2.2 and 4.2] for a description of the algorithm.

Certificates via Approximate Algorithms. In practice, most probabilistic model checkers use algorithms that are not exact but approximate: They employ *approximate, fixed-precision floating point arithmetic* and use a variant of VI that only returns an approximate result, namely for each state an interval $[\ell, u]$ containing the exact value, such that $|\ell - u| \leq \varepsilon$ for a given ε (typically 10^{-6}). They do this because (i) when using exact arithmetic, fractions can grow very large, hindering scalability, (ii) VI-based algorithms often outperform PI, albeit not as dramatically as folklore claimed [29,30], and (iii) approximate results usually suffice. We now exemplify with the VI-variant *Interval Iteration* (II) [8,24] how to make an approximate, floating point-based algorithm certifying, leaving other variants such as *optimistic VI* [31] and *Sound VI* [49] for future work.

II for reachability[6] works by first *collapsing end components* [13,24] of the MDP to ensure that $\mathcal{B}^{\text{opt}}$ has a *unique* fixed point. II then runs two instances of VI in parallel, starting from $x^{(0)} = \mathbf{0}$ and $y^{(0)} = \mathbf{1}$:

$$\mathbf{0} = x^{(0)} \leq \mathcal{B}^{\text{opt}}(x^{(0)}) = x^{(1)} \leq \ \ldots \ \text{fp}\,\mathcal{B}^{\text{opt}} \ \ldots \ \leq y^{(1)} = \mathcal{B}^{\text{opt}}(y^{(0)}) \leq y^{(0)} = \mathbf{1}$$

Both sequences contain (co-)inductive vectors only and converge to the fixed point. The iteration can be stopped when the difference is as small as desired.

However, as we demonstrate experimentally (Section 7), *inexact floating point arithmetic usually breaks (co-)inductivity of the elements in the II sequences*, as was already reported in [55] in a similar setting. More precisely, let $\mathcal{B}^{\text{opt}}_{\mathbb{F}}$ be a "floating point variant" of $\mathcal{B}^{\text{opt}}$, i.e., the (exact) result of each operation is rounded *to a nearest float*. This the default rounding mode in IEEE 754. Let

[6] For expected rewards, II additionally requires computing an upper bound, see [8].

$x_{\mathbb{F}}^{(i)}$ be the i-th element, $i > 0$, in the lower VI sequence of $\mathcal{B}_{\mathbb{F}}^{\text{opt}}$ starting from $\vec{0}$. Then, due to rounding errors, $x_{\mathbb{F}}^{(i)} \leq \mathcal{B}^{\text{opt}}(x_{\mathbb{F}}^{(i)})$ does *not* hold in general, i.e., $x_{\mathbb{F}}^{(i)}$ might not be co-inductive. We propose two ways to mitigate this problem: *Safe rounding* [27] and *Smooth II*.

First, safe rounding amounts to configuring the IEEE754 rounding mode so that results of floating point computations are always rounded towards 0 when iterating from below, and towards ∞ when iterating from above. While safe rounding provably yields sound bounds [27], it may slow down or even prevent convergence of II. Nonetheless, in practice, II with safe rounding finds significantly more certificates than II with default rounding (Section 7).

Second, for Smooth II we define the γ-*smooth Bellman operator* ($\gamma \in [0, 1)$)

$$\mathcal{B}_{\gamma}^{\text{opt}}(x) \;=\; \gamma \cdot x + (1 - \gamma) \cdot \mathcal{B}^{\text{opt}}(x) \;,$$

where scalar multiplication and addition are component-wise. $\mathcal{B}_{\gamma}^{\text{opt}}$ and $\mathcal{B}^{\text{opt}}$ have the same fixed points, and every (co-)inductive value vector w.r.t. $\mathcal{B}_{\gamma}^{\text{opt}}$ is also (co-)inductive w.r.t. $\mathcal{B}^{\text{opt}}$ [17, App. G.1]. The key property of $\mathcal{B}_{\gamma}^{\text{opt}}$ compared to $\mathcal{B}^{\text{opt}}$ is that the former enforces *ultimately strictly monotonic VI sequences*. This mitigates the floating point rounding issues. Notice, however, that smoothing slows down convergence. Smoothing and safe rounding may be combined.

Computing Ranking Functions. We briefly outline how to obtain the unique and least fixed points of $\mathcal{D}^{\text{opt}}$ and $\overline{\mathcal{D}}^{\text{opt}}$, respectively (see Definitions 1 and 2).

First, $\text{fp}\,\mathcal{D}^{\text{opt}}$ can be computed via VI from $r^{(0)} = \vec{\infty}$. This iteration converges in finitely many steps. Second, to compute $\text{lfp}\,\overline{\mathcal{D}}^{\text{opt}}$ we propose to perform VI from $r^{(0)}$ with $r^{(0)}(s) = [\overline{\mathbb{P}_s^{\text{opt}}}(\Diamond T) = 1] \cdot \infty$ for all $s \in S$. The condition in the Iverson bracket can be evaluated using standard graph analysis [7, Section 10.6.1]. This iteration converges in finitely many steps as well, see [17, Apps. G.2 and G.3] for details and a practically more efficient algorithm.

7 Experimental Evaluation

Implementation. We implemented certificate computation as discussed in Section 6 in Storm [33]. Given a higher-level model description (PRISM [42] or Jani [14]), and a reachability probability or expected reward query, our implementation proceeds in three steps: First, Storm builds an explicit MDP from the description. Second, it computes a certificate for both lower *and* upper bounds, such that the relative difference between the two values is at most $\varepsilon = 10^{-6}$ for each MDP state. Finally, Storm checks the validity of the certificate.

Following the discussion in Section 6, we consider the following algorithms: Regarding exact computation, we use PI with *exact* LU decomposition, called PI^X. For approximate computation with floating point arithmetic, we employ II. Further, to investigate the impact of the rounding error mitigation techniques from Section 6, we complement II with either safe rounding (denoted II_{rnd}), smoothing with parameter γ (denoted $\text{II}^{\circlearrowleft\gamma}$; we consider $\gamma \in \{0.05, 0.8, 0.9, 0.95\}$), or a combination of both (denoted $\text{II}_{\text{rnd}}^{\circlearrowleft\gamma}$). Overall, we compare PI^X and seven

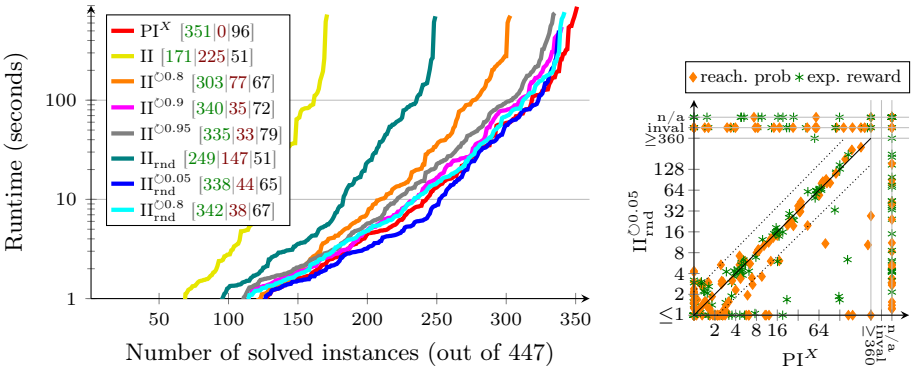

Fig. 3. RQ1: Runtime for computing certificates of PI^X and several combinations of mitigation techniques with II (left); and detailed comparison of PI^X and $\mathrm{II}_{\mathrm{rnd}}^{\circlearrowleft 0.05}$ (right).

variants of II. We employ additional standard modifications of the algorithms, namely: We eliminate end components whenever possible, apply topological optimizations for PI^X and II, and apply Gauß-Seidl Bellman updates for II [8,29].

In all three steps of the implementation, we represent numbers as arbitrary precision rationals implemented in GMP [23]—except when running II in the second step (in which case we convert rationals to their nearest floats, potentially yielding invalid certificates). We thus certify reachability probabilities and expected rewards with respect to the *exact* MDP without rounding errors.

The MDP and the certificate computed with Storm can be exported and checked by an independent *formally verified certificate checker*. To construct the latter, we verified the correctness of the certificate checking algorithms in the interactive proof assistant Isabelle/HOL [46], extending previous work on MDPs [34,50] by total rewards and qualitative reachability properties. Based on this library, we proved correct the soundness of the certificates described in Sections 3 to 5. We used Isabelle/HOL's code export mechanism [25] to obtain a verified, executable Standard ML implementation that employs exact rational arithmetic. The construction of the MDP from a PRISM or Jani model as well as export and parsing of MDPs and certificates are currently not verified.

Benchmarks and Setup. We use all 366 benchmark instances from the quantitative verification benchmark set (QVBS) [32] that (i) consider an MDP with a reachability or reward objective and (ii) for which Storm can build an explicit representation within 5 minutes. Additionally, since the QVBS contains no models exhibiting non-trivial ECs, we include 71 structurally diverse models from various sources detailed in [30, Sec. 5.3]. Overall, we consider the complete *alljani* set from [30]. We invoke Storm for each combination of benchmark instance and certificate algorithm and report the overall runtime (walltime). All experiments ran on Intel Xeon 8468 Sapphire 2.1 Ghz systems. We used Slurm to limit the individual executions to 4 CPU cores and 16 GB of RAM, with a time limit of 900 s. Next, we discuss our findings by answering three research questions (RQs).

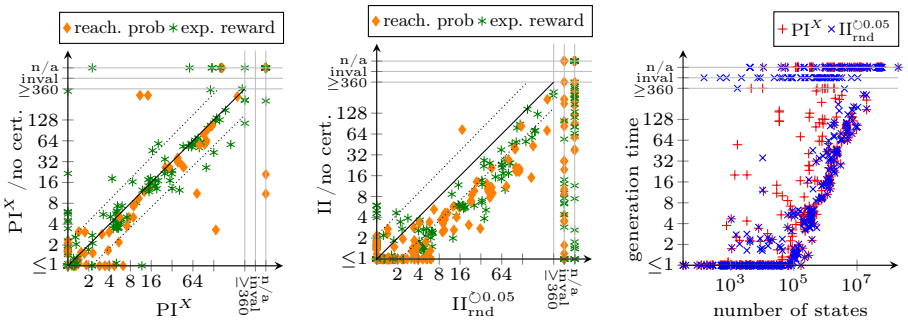

Fig. 4. RQ2: Runtime overhead of certified MDP model checking for PI^X (left) and II (middle), and scalability of both with respect to the number of states (right).

RQ1: Best algorithm for certificate generation? Figure 3 (left) compares the runtimes of PI^X and our seven II variants. A point (x, y) for algorithm A indicates that there are x instances for which A computes a *valid* certificate within y seconds (including time for model construction but excluding time for exporting the certificate files). The triples $[v|w|u]$ in the legend indicate that the algorithm produced a total of v valid and w invalid certificates (with invalidity likely due to floating point issues), while for the remaining u instances no result was found within the resource limits. As expected, all certificates produced by the exact PI^X are valid, while standard II produces many invalid certificates. Safe rounding and smoothing improve the number of valid certificates. Notably, $\mathrm{II}^{\circlearrowleft\gamma}$ (only smoothing) performs best for γ values close to 1, while the performance of $\mathrm{II}^{\circlearrowleft\gamma}_{\mathrm{rnd}}$ (smoothing *and* safe rounding) is less sensitive towards γ; see [17, App. H] for more details. Among all II variants, $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$ shows the best overall performance.

The scatter plot in Figure 3 (right) further compares PI^X and $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$. A data point (x, y) corresponds to one benchmark instance, where x and y are runtimes of PI^X and $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$. A point at ≥ 300 indicates a runtime between 300 and 900 seconds, *inval* means an invalid certificate, and *n/a* denotes an aborted computation due to time/memory limits. Many instances that PI^X cannot solve are solved by $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$ and vice versa. This is already the case without computing certificates, as the structure of a benchmark affects the performance of the algorithms differently [29,30]. Thus, as in the case without computing certificates, there is no "best algorithm", and both PI^X and variants of II can be considered. Overall, 396 out of 447 instances are correctly solved and certified by PI^X or $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$ (or both). We highlight that PI^X is not only a complete certifying algorithm, but also practically efficient, even though it uses exact arithmetic.

RQ2: Runtime overhead of certificate generation? Figure 4 (left/middle) reports the runtime overhead of generating a certificate for PI^X and $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$. For PI^X, the overhead is typically within a factor of 2, often significantly less. It is sometimes faster due to implementation differences in the certifying variant of PI^X. For $\mathrm{II}^{\circlearrowleft 0.05}_{\mathrm{rnd}}$, the overhead is slightly larger, typically around 1.5 to 4. This is partly due to the slower convergence caused by smoothing. Figure 4 (right) investigates

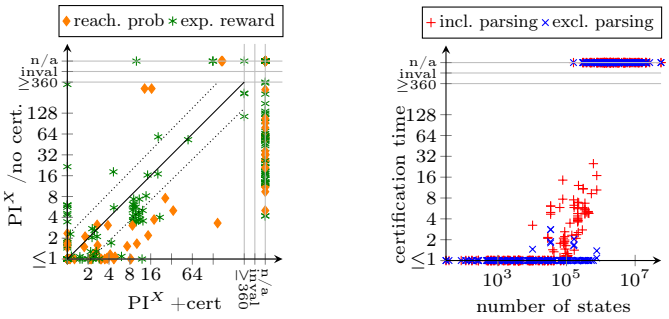

Fig. 5. RQ3: Runtime overhead of the certified pipeline/Runtime of certificate checking

the scalability of certificate generation with respect to the number of states. For MDPs with up to 10^5 states, certificate generation usually completes within a minute (often much less); for more than 10^7 states, it usually times out.

RQ3: Scalability of the formally verified certificate checker? Figure 5 (left) compares the runtime of the full pipeline including certificate generation and verification using our formally verified checker (PI^X +cert) with plain, uncertified MDP model checking based on PI^X. Compared to Figure 4 (left), the added verification of the certificates causes additional time/memory outs, and roughly doubles the runtime of the other instances. Figure 5 (right) reveals that parsing is currently a major bottleneck in the verified checker. Nonetheless, the checker completes within a few seconds on MDPs with up to $\approx 10^5$ states, and usually within 30 s for instances with up to $\approx 10^6$ states.

8 Conclusion and Future Work

We proposed *fixed point certificates* as a new standard for certified model checking of reachability and expected reward properties in MDPs. The soundness of these certificates was formalized in Isabelle/HOL, increasing their trustworthiness and enabling us to generate a formally verified certificate checker, applicable to non-trivial practically relevant instances. Our certificates can be generated with moderate overhead via minor, yet careful, modifications of established algorithms like II or PI. This allows tool developers and competitions [15]—for which our certificates provide formally verified reference results—to adopt our proposal with relatively low effort. Future work is to develop a more efficient certificate format. Further, we plan to extend our theory to other quantitative verification settings [3], e.g., stochastic games and ω-regular properties, and make it amenable to techniques such as *symbolic model checking* and *partial exploration*.

Data availability statement. The models, tools, and scripts to reproduce our experimental evaluation are available at DOI 10.5281/zenodo.14626585 [18].

References

1. Abate, A., Giacobbe, M., Roy, D.: Stochastic omega-regular verification and control with supermartingales. In: CAV (3). Lecture Notes in Computer Science, vol. 14683, pp. 395–419. Springer (2024). https://doi.org/10.1007/978-3-031-65633-0_18
2. de Alfaro, L.: Formal verification of probabilistic systems. Ph.D. thesis, Stanford University, USA (1997)
3. Andriushchenko, R., Bork, A., Budde, C.E., Češka, M., Hahn, E.M., Hartmanns, A., Israelsen, B., Jansen, N., Jeppson, J., Junges, S., Köhl, M.A., Könighofer, B., Křetínský, J., Meggendorfer, T., Parker, D., Pranger, S., Quatmann, T., Ruijters, E., Taylor, L., Volk, M., Weininger, M., Zhang, Z.: Tools at the frontiers of quantitative verification: QComp 2023 competition report. In: International TOOLympics Challenge, pp. 90–146. Springer (2024)
4. Azeem, M., Evangelidis, A., Kretínský, J., Slivinskiy, A., Weininger, M.: Optimistic and topological value iteration for simple stochastic games. In: ATVA. Lecture Notes in Computer Science, vol. 13505, pp. 285–302. Springer (2022). https://doi.org/10.1007/978-3-031-19992-9_18
5. Baier, C., de Alfaro, L., Forejt, V., Kwiatkowska, M.: Model checking probabilistic systems. In: Handbook of Model Checking, pp. 963–999. Springer (2018). https://doi.org/10.1007/978-3-319-10575-8_28
6. Baier, C., Chau, C., Klüppelholz, S.: Certificates and witnesses for multi-objective queries in Markov decision processes. In: QEST+FORMATS. Lecture Notes in Computer Science, vol. 14996, pp. 1–18. Springer (2024). https://doi.org/10.1007/978-3-031-68416-6_1
7. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
8. Baier, C., Klein, J., Leuschner, L., Parker, D., Wunderlich, S.: Ensuring the reliability of your model checker: Interval iteration for Markov decision processes. In: CAV (1). Lecture Notes in Computer Science, vol. 10426, pp. 160–180. Springer (2017). https://doi.org/10.1007/978-3-319-63387-9_8
9. Balyo, T., Heule, M., Iser, M., Järvisalo, M., Suda, M.: Proceedings of SAT Competition 2023: Solver, benchmark and proof checker descriptions (2023), https://researchportal.helsinki.fi/en/publications/proceedings-of-sat-competition-2023-solver-benchmark-and-proof-ch
10. Batz, K., Biskup, T.J., Katoen, J., Winkler, T.: Programmatic strategy synthesis: Resolving nondeterminism in probabilistic programs. Proc. ACM Program. Lang. **8**(POPL), 2792–2820 (2024). https://doi.org/10.1145/3632935
11. Bellman, R.: Dynamic programming and stochastic control processes. Inf. Control. **1**(3), 228–239 (1958). https://doi.org/10.1016/S0019-9958(58)80003-0
12. Beyer, D.: Competition on software verification and witness validation: SV-COMP 2023. In: TACAS (2). Lecture Notes in Computer Science, vol. 13994, pp. 495–522. Springer (2023). https://doi.org/10.1007/978-3-031-30820-8_29
13. Brázdil, T., Chatterjee, K., Chmelik, M., Forejt, V., Kretínský, J., Kwiatkowska, M.Z., Parker, D., Ujma, M.: Verification of Markov decision processes using learning algorithms. In: ATVA. Lecture Notes in Computer Science, vol. 8837, pp. 98–114. Springer (2014). https://doi.org/10.1007/978-3-319-11936-6_8
14. Budde, C.E., Dehnert, C., Hahn, E.M., Hartmanns, A., Junges, S., Turrini, A.: JANI: quantitative model and tool interaction. In: TACAS (2). Lecture Notes in Computer Science, vol. 10206, pp. 151–168 (2017). https://doi.org/10.1007/978-3-662-54580-5_9

15. Budde, C.E., Hartmanns, A., Klauck, M., Kretínský, J., Parker, D., Quatmann, T., Turrini, A., Zhang, Z.: On correctness, precision, and performance in quantitative verification - QComp 2020 competition report. In: ISoLA (4). Lecture Notes in Computer Science, vol. 12479, pp. 216–241. Springer (2020). https://doi.org/10.1007/978-3-030-83723-5_15
16. Chatterjee, K., Henzinger, T.A.: Value iteration. In: 25 Years of Model Checking. Lecture Notes in Computer Science, vol. 5000, pp. 107–138. Springer (2008). https://doi.org/10.1007/978-3-540-69850-0_7
17. Chatterjee, K., Quatmann, T., Schäffeler, M., Weininger, M., Winkler, T., Zilken, D.: Fixed point certificates for reachability and expected rewards in MDPs. CoRR **abs/2501.11467** (2025), https://arxiv.org/abs/2501.11467
18. Chatterjee, K., Quatmann, T., Schäffeler, M., Weininger, M., Winkler, T., Zillken, D.: Artifact: Fixed point certificates for reachability and expected rewards in mdps (2025). https://doi.org/10.5281/zenodo.14626585
19. Chen, T., Forejt, V., Kwiatkowska, M.Z., Parker, D., Simaitis, A.: Automatic verification of competitive stochastic systems. Formal Methods Syst. Des. **43**(1), 61–92 (2013). https://doi.org/10.1007/S10703-013-0183-7, https://doi.org/10.1007/s10703-013-0183-7
20. Debbi, H.: Counterexamples in model checking - A survey. Informatica (Slovenia) **42**(2) (2018), http://www.informatica.si/index.php/informatica/article/view/1442
21. Froleyks, N., Yu, E., Biere, A., Heljanko, K.: Certifying phase abstraction. In: IJCAR (1). Lecture Notes in Computer Science, vol. 14739, pp. 284–303. Springer (2024). https://doi.org/10.1007/978-3-031-63498-7_17
22. Funke, F., Jantsch, S., Baier, C.: Farkas certificates and minimal witnesses for probabilistic reachability constraints. In: TACAS (1). Lecture Notes in Computer Science, vol. 12078, pp. 324–345. Springer (2020). https://doi.org/10.1007/978-3-030-45190-5_18
23. Granlund, T., Team, G.D.: GNU MP 6.0 Multiple Precision Arithmetic Library. Samurai Media Limited (2015)
24. Haddad, S., Monmege, B.: Interval iteration algorithm for MDPs and IMDPs. Theor. Comput. Sci. **735**, 111–131 (2018). https://doi.org/10.1016/J.TCS.2016.12.003
25. Haftmann, F., Krauss, A., Kuncar, O., Nipkow, T.: Data refinement in Isabelle/HOL. In: ITP. Lecture Notes in Computer Science, vol. 7998, pp. 100–115. Springer (2013). https://doi.org/10.1007/978-3-642-39634-2_10
26. Hahn, E.M., Hartmanns, A., Hensel, C., Klauck, M., Klein, J., Kretínský, J., Parker, D., Quatmann, T., Ruijters, E., Steinmetz, M.: The 2019 comparison of tools for the analysis of quantitative formal models - (QComp 2019 competition report). In: TACAS (3). Lecture Notes in Computer Science, vol. 11429, pp. 69–92. Springer (2019). https://doi.org/10.1007/978-3-030-17502-3_5
27. Hartmanns, A.: Correct probabilistic model checking with floating-point arithmetic. In: TACAS (2). Lecture Notes in Computer Science, vol. 13244, pp. 41–59. Springer (2022). https://doi.org/10.1007/978-3-030-99527-0_3
28. Hartmanns, A., Hermanns, H.: The Modest Toolset: An integrated environment for quantitative modelling and verification. In: TACAS. LNCS, vol. 8413, pp. 593–598. Springer (2014). https://doi.org/10.1007/978-3-642-54862-8_51
29. Hartmanns, A., Junges, S., Quatmann, T., Weininger, M.: A practitioner's guide to MDP model checking algorithms. In: TACAS (1). Lecture Notes in Computer Science, vol. 13993, pp. 469–488. Springer (2023). https://doi.org/10.1007/978-3-031-30823-9_24

30. Hartmanns, A., Junges, S., Quatmann, T., Weininger, M.: The revised practitioner's guide to MDP model checking algorithms. Under submission, Preprint: https://sjunges.github.io/files/revised_practitioners_guide.pdf (2025)
31. Hartmanns, A., Kaminski, B.L.: Optimistic value iteration. In: CAV (2). Lecture Notes in Computer Science, vol. 12225, pp. 488–511. Springer (2020). https://doi.org/10.1007/978-3-030-53291-8_26
32. Hartmanns, A., Klauck, M., Parker, D., Quatmann, T., Ruijters, E.: The quantitative verification benchmark set. In: TACAS (1). Lecture Notes in Computer Science, vol. 11427, pp. 344–350. Springer (2019). https://doi.org/10.1007/978-3-030-17462-0_20
33. Hensel, C., Junges, S., Katoen, J., Quatmann, T., Volk, M.: The probabilistic model checker Storm. Int. J. Softw. Tools Technol. Transf. **24**(4), 589–610 (2022). https://doi.org/10.1007/S10009-021-00633-Z
34. Hölzl, J.: Markov chains and Markov decision processes in Isabelle/HOL. J. Autom. Reason. **59**(3), 345–387 (2017). https://doi.org/10.1007/S10817-016-9401-5
35. Jantsch, S.: Certificates and Witnesses for Probabilistic Model Checking. Ph.D. thesis, Dresden University of Technology, Germany (2022)
36. Jantsch, S., Harder, H., Funke, F., Baier, C.: SWITSS: Computing small witnessing subsystems. In: FMCAD. pp. 236–244. IEEE (2020). https://doi.org/10.34727/2020/ISBN.978-3-85448-042-6_31
37. Junges, S., Katoen, J., Pérez, G.A., Winkler, T.: The complexity of reachability in parametric Markov decision processes. J. Comput. Syst. Sci. **119**, 183–210 (2021). https://doi.org/10.1016/J.JCSS.2021.02.006
38. Kratsch, D., McConnell, R.M., Mehlhorn, K., Spinrad, J.P.: Certifying algorithms for recognizing interval graphs and permutation graphs. SIAM J. Comput. **36**(2), 326–353 (2006). https://doi.org/10.1137/S0097539703437855
39. Kretínský, J., Meggendorfer, T., Weininger, M.: Stopping criteria for value iteration on stochastic games with quantitative objectives. In: LICS. pp. 1–14. IEEE (2023). https://doi.org/10.1109/LICS56636.2023.10175771
40. Kupferman, O., Sickert, S.: Certifying inexpressibility. In: FoSSaCS. Lecture Notes in Computer Science, vol. 12650, pp. 385–405. Springer (2021). https://doi.org/10.1007/978-3-030-71995-1_20
41. Kupferman, O., Vardi, M.Y.: From complementation to certification. Theor. Comput. Sci. **345**(1), 83–100 (2005). https://doi.org/10.1016/J.TCS.2005.07.021
42. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer (2011). https://doi.org/10.1007/978-3-642-22110-1_47
43. Lechner, M., Zikelic, D., Chatterjee, K., Henzinger, T.A.: Stability verification in stochastic control systems via neural network supermartingales. In: AAAI. pp. 7326–7336. AAAI Press (2022). https://doi.org/10.1609/AAAI.V36I7.20695
44. McConnell, R.M., Mehlhorn, K., Näher, S., Schweitzer, P.: Certifying algorithms. Comput. Sci. Rev. **5**(2), 119–161 (2011). https://doi.org/10.1016/J.COSREV.2010.09.009
45. Namjoshi, K.S.: Certifying model checkers. In: CAV. Lecture Notes in Computer Science, vol. 2102, pp. 2–13. Springer (2001). https://doi.org/10.1007/3-540-44585-4_2
46. Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL - A Proof Assistant for Higher-Order Logic, Lecture Notes in Computer Science, vol. 2283. Springer (2002). https://doi.org/10.1007/3-540-45949-9, https://doi.org/10.1007/3-540-45949-9

47. Peled, D.A., Pnueli, A., Zuck, L.D.: From falsification to verification. In: FSTTCS. Lecture Notes in Computer Science, vol. 2245, pp. 292–304. Springer (2001). https://doi.org/10.1007/3-540-45294-X_25
48. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics, Wiley (1994). https://doi.org/10.1002/9780470316887
49. Quatmann, T., Katoen, J.: Sound value iteration. In: CAV (1). Lecture Notes in Computer Science, vol. 10981, pp. 643–661. Springer (2018). https://doi.org/10.1007/978-3-319-96145-3_37
50. Schäffeler, M., Abdulaziz, M.: Formally verified solution methods for Markov decision processes. In: AAAI. pp. 15073–15081. AAAI Press (2023). https://doi.org/10.1609/AAAI.V37I12.26759
51. Takisaka, T., Zhang, L., Wang, C., Liu, J.: Lexicographic ranking supermartingales with lazy lower bounds. In: CAV (3). Lecture Notes in Computer Science, vol. 14683, pp. 420–442. Springer (2024). https://doi.org/10.1007/978-3-031-65633-0_19
52. Tan, Y.K., Yang, J., Soos, M., Myreen, M.O., Meel, K.S.: Formally certified approximate model counting. In: CAV (1). Lecture Notes in Computer Science, vol. 14681, pp. 153–177. Springer (2024). https://doi.org/10.1007/978-3-031-65627-9_8
53. White, D.J.: A survey of applications of Markov decision processes. Journal of the operational research society **44**(11), 1073–1096 (1993). https://doi.org/10.1057/jors.1993.181
54. Wimmer, R., Jansen, N., Ábrahám, E., Katoen, J., Becker, B.: Minimal counterexamples for linear-time probabilistic verification. Theor. Comput. Sci. **549**, 61–100 (2014). https://doi.org/10.1016/J.TCS.2014.06.020, https://doi.org/10.1016/j.tcs.2014.06.020
55. Winkler, T., Katoen, J.: Certificates for probabilistic pushdown automata via optimistic value iteration. In: TACAS (2). Lecture Notes in Computer Science, vol. 13994, pp. 391–409. Springer (2023). https://doi.org/10.1007/978-3-031-30820-8_24
56. Winkler, T., Katoen, J.: On certificates, expected runtimes, and termination in probabilistic pushdown automata. In: LICS. pp. 1–13. IEEE (2023). https://doi.org/10.1109/LICS56636.2023.10175714
57. Yu, E., Biere, A., Heljanko, K.: Progress in certifying hardware model checking results. In: CAV (2). Lecture Notes in Computer Science, vol. 12760, pp. 363–386. Springer (2021). https://doi.org/10.1007/978-3-030-81688-9_17
58. Yu, E., Froleyks, N., Biere, A., Heljanko, K.: Stratified certification for k-induction. In: FMCAD. pp. 59–64. IEEE (2022). https://doi.org/10.34727/2022/ISBN.978-3-85448-053-2_11
59. Yu, E., Froleyks, N., Biere, A., Heljanko, K.: Towards compositional hardware model checking certification. In: FMCAD. pp. 1–11. IEEE (2023). https://doi.org/10.34727/2023/ISBN.978-3-85448-060-0_12