# Modal Stochastic Games*

## —Abstraction-Refinement of Probabilistic Automata—

Joost-Pieter Katoen[1] and Falak Sher[2]

[1] RWTH Aachen University, Germany
[2] Information Technology University, Punjab, Pakistan

**Abstract.** This paper presents an abstraction-refinement framework for Segala's probabilistic automata (PA), a slight variant of Markov decision processes. We use Condon and Ladner's two-player probabilistic game automata extended with *possible* and *required* transitions — as in Larsen and Thomsen's modal transition systems — as abstract models. The key idea is to refine player-one and player-two states separately resulting in a nested abstract-refine loop. We show the adequacy of this approach for obtaining tight bounds on extremal reachability probabilities.

## 1 Introduction

Probabilistic automata (PAs) [1] extend Markov decision processes (MDPs) by allowing for states having more than one choice labeled with the same action. This extension is needed for parallel composition. Whereas in an MDP, each distribution (over states) is unique, this no longer holds for PA. PAs have been used as operational model for probabilistic process algebras, the PIOA language, and have served to reason about randomized distributed algorithms, see [2]. Segala [1] has studied several behavioral relations on PAs such as (weak and strong) bisimulation and simulation pre-orders, as well as trace inclusions. These relations form the basis for obtaining abstractions of PAs, i.e., smaller models that then can be used for further analysis. This includes for instance, determining extremal (minimal and maximal) reachability probabilities.

To obtain coarser abstractions, more aggressive abstraction schemes have been proposed in the literature. These include finite-state approximations [3], abstract probabilistic automata [4], game-based abstractions [5], abstractions that are based on distribution-based simulation pre-orders [6], and compositional abstraction [7]. This paper is a continuation of this line of research that is aimed at obtaining an automated abstraction-refinement framework for PAs that yields tight bounds on extremal reachability probabilities.

The first key ingredient of this paper is to use Condon and Ladner's two-player probabilistic game automata (PGAs) [8] and extend them with possible and required transitions as known from modal transition systems [9, 10]. There are two main differences with existing works on game-based abstraction of PAs:

* This work has been partially funded by the Excellence Initiative of the German federal and state government and the CDZ project CAP (GZ 1023).

*(1) both players are fully symmetric (and randomized), and (2) transitions have modalities.* We define satisfaction and refinement relations — much in the style of modal transition systems — on these models, define (alternating) simulation relations, and prove the special role of two specific implementations that provide (upper and lower) bounds on extremal reachability probabilities for competing and collaborating players.

The second key ingredient, and the major contribution of this paper, is an (nested) abstraction-refinement scheme. The main idea is separate refining player-one and player-two states. We formally define the notion of stable abstraction from the perspective of each player, prove that each refinement loop indeed yields a refinement, and that the iterative abstraction-refinement terminates for every PA with a finite bisimulation quotient.

Put shortly, the major contributions of this paper are: (1) generalizing two-player probabilistic game automata (by annotating transitions with modalities) and proposing them as abstractions of PAs, (2) showing that our abstractions yield *at most* as tight bounds on extremal reachability probabilities as game-based abstractions, however, they are *at most* the sizes of game-based abstractions, and (3) proposing an abstraction-refinement framework consisting of a nested loop – the inner-loop (outer-loop) refines player-one (player-two) states.

This paper is organized as follows. Section 2 sets the ground for this work. Sections 3 and 4 introduce abstract PGAs and the abstraction technique based on it, respectively. Section 5 proposes our abstraction-refinement framework for PAs. Section 6 discusses related work. Section 7 concludes the paper. Proofs of theorems can be found in the Ph.D. thesis [18].

## 2 Preliminaries

A *distribution* $\mu$ is a function on a countable set $S$ iff $\mu : S \to [0,1]$ and $0 < \sum_{s \in S} \mu(s) \leq 1$; its support set is $\text{supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$; and its mass w.r.t. set $S' \subseteq S$ is given as $\mu(S') = \sum_{s \in S'} \mu(s)$.A distribution $\mu$ is a *full-distribution* iff $\mu(S) = 1$, otherwise, it is a *sub-distribution*. Let $\text{Dist}(S)$ denote the set of full-distributions over $S$. Let $\iota_s \in \text{Dist}(S)$ denote the *Dirac* distribution for $s \in S$, i.e., $\iota_s(s) = 1$.

### 2.1 Probabilistic Game Automata

PGAs are used for modeling systems in which players, behaving probabilistically, compete for certain objectives, i.e., some players maximize whereas the others minimize the probability of reaching a set of goal states. In this paper, we deal with PGAs having only two players that make their moves alternatively. Intuitively, it is a game of chance played between two players, say, player one and player two. The game arena is a bipartite graph — having, say, $S_1$ and $S_2$ as sets of vertices — in which each player owns a specific set of vertices; say, the players one and two own $S_1$ and $S_2$ respectively. The

game is started by player one and evolves in a turn-based fashion. Starting from the initial state in $S_1$, player one non-deterministically chooses an action-distribution pair. Based on the selected distribution, a state in $S_2$, say $s_2$, is randomly selected and the control is passed to player two; who then behaves in the same way as player one and the control passes back to player one. This goes on until some goal is achieved either by player one or player two. Let UAct be a countable universe of actions including the internal action $\tau$.

**Definition 1 (Probabilistic game automaton [8]).** *A* PGA *is a tuple* $\mathcal{G} = (S, \{S_1, S_2\}, A, \Delta, s_0)$ *where* $S$ *is a non-empty, countable set of states, partitioned into* $S_1$ *and* $S_2$*, with* $s_0 \in S_1$*;* $A \subseteq \text{UAct}$*, and* $\Delta \subseteq (S_1 \times A \times \text{Dist}(S_2)) \cup (S_2 \times A \times \text{Dist}(S_1))$ *is a set of transitions.*

We denote $(s, a, \mu) \in \Delta$ by $s \xrightarrow{a} \mu$; $\text{Act}(s) = \{a \in A \mid s \xrightarrow{a} \mu\}$ as the set of enabled actions from state $s$; $\text{succ}(s) = \{u \in S \mid \exists (s, a, \mu) \in \Delta : \mu(u) > 0\}$ as the set of successor states of $s$; and $\Delta(s) = \{(s, a, \mu) \mid s \xrightarrow{a} \mu\}$ as the set of transitions emanating from $s$. PGAs are thus a generalization of SGs [11] in which both players are random; in SGs only one player is random. In the sequel, let $\mathcal{G} = (S, \{S_1, S_2\}, A, \Delta, s_0)$ be a finitely branching – each state has a finite number of transitions and each distribution has a finite support – PGA. To depict PGAs we represent states in $S_1$ and $S_2$ as rectangles and double rectangles respectively. Moreover, if a player-one state $s$ has a unique predecessor $t$, we show $s$ inside $t$ for simplicity. Fig. 1 illustrates a sample PGA with $S_1 = \{s_0, \ldots, s_4\}$, $S_2 = \{t_1, t_2, t_3\}$ and transitions $t_1 \xrightarrow{a} \mu$ with $\mu(s_1) = \mu(s_2) = \frac{1}{2}$. In order to analyze reachability properties on PGA $\mathcal{G}$, at each state non-determinism is resolved by means of a scheduler for each player, resulting in a Markov chain with a countable state space. The induced chain further reduces to a path once probabilistic choices are resolved. A set of paths obtained thus is measurable, see e.g., [12, Ch. 10]. Let $\Pr^{\kappa_1}_{\kappa_2}(T)$ be the probability of the set of paths from the initial state $s_0$ in $\mathcal{G}$ that reach some set of states $T \subseteq S$ under schedulers $(\kappa_1, \kappa_2)$ for players one and two respectively. Let

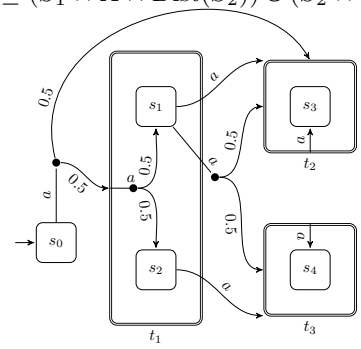

Fig. 1: A sample PGA $\mathcal{G}$

$$\begin{aligned}
\Pr^{+-}(T) &= \sup_{\kappa_1} \inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T) \\
\Pr^{++}(T) &= \sup_{\kappa_1} \sup_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T) \\
\Pr^{--}(T) &= \inf_{\kappa_1} \inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T) \\
\Pr^{-+}(T) &= \inf_{\kappa_1} \sup_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)
\end{aligned}$$

be the *optimal* (i.e., maximum and minimum) probabilities for reaching states in $T$. They can be achieved by *deterministic memoryless* schedulers [8], and computed through value iteration, policy iteration or by linear programming for games with finite state spaces.

Let $w_T : S \to [0,1]$ be a *probability valuation function* mapping a state $s$ to the probability of reaching $T \subseteq S$ from $s$ under a given pair of deterministic memoryless schedulers. We omit the subscript $T$ whenever $T$ is clear from the context. The probability valuation functions $W_T = \{w \mid w : S \to [0,1]\}$ form a complete lattice $(W_T, \leq, \bot, \top)$ with order $\leq \subseteq W_T \times W_T$, bottom element $\bot \in W_T$ and top element $\top \in W_T$. We write $w \leq w'$ iff $\forall s \in S : w(s) \leq w'(s)$; $\bot(s) = 0$ and $\top(s) = 1$ for $s \in S$. For a set $M \subseteq W_T$, the least upper bound is given as $\bigsqcup M(s) = \sup_{w \in M} w(s)$, and the greatest lower bound as $\bigsqcap M(s) = \inf_{w \in M} w(s)$ for $s \in S$. Let $w(\mu) = \sum_{s \in S} \mu(s) \cdot w(s)$ for $\mu \in \mathrm{Dist}(S)$. For PGA $\mathcal{G}$, let $\tau(\mathcal{G})$ be the closed PGA, a PGA $\mathcal{G}$ in which all actions of $\mathcal{G}$ are changed into $\tau$.[3]

**Definition 2 (Probability valuation transformer [8]).** *Let $T \subseteq S$ be the set of goal states in PGA $\tau(\mathcal{G})$. For reachability objectives $\mathbf{1}, \mathbf{2} \in \{\min, \max\}$ for players one and two respectively, the* probability valuation transformer $\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}} : W_T \to W_T$ *is defined for $w \in W_T$ and $s \in S$ as:*

$$\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(w)(s) = \begin{cases} 1 & \text{if } s \in T \\ \mathbf{1} = \max?\ 0 : 1 & \text{if } s \in S_1 \cap T_0 \\ \mathbf{2} = \max?\ 0 : 1 & \text{if } s \in S_2 \cap T_0 \\ \mathbf{1}\{w(\mu) \mid s \xrightarrow{\tau} \mu\} & \text{if } s \in S_1 \setminus (T \cup T_0) \\ \mathbf{2}\{w(\mu) \mid s \xrightarrow{\tau} \mu\} & \text{if } s \in S_2 \setminus (T \cup T_0) \end{cases}$$

*where $T_0 \subseteq S$ is the set of all states without outgoing transitions.*

$\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}$ is a monotonic function over the complete lattice $W$. By Tarski's theorem [14], it has a least **Fix** $\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\bot)$ and a greatest **Fix** $\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\top)$ fixed point. For finite-state PGA, they can be computed through e.g., value iteration [13].

### 2.2 Probabilistic automata

PAs [1] extend labeled transition systems (LTSs) in which the target of any action-labeled transition is a distribution over states instead of a single state. A *probabilistic automaton* (PA) is a quadruple $\mathcal{M} = (S, A, \Delta, s_0)$ where $S$, $A$, and $s_0$ are as before, and $\Delta \subseteq S \times A \times \mathrm{Dist}(S)$ is a set of transitions. A PA can be embedded into a PGA (where player-two states have one emanating transition) in a straightforward manner. Fig. 2 depicts a sample PA. Its embedding as PGA is provided in Fig. 4 (left, page 8).

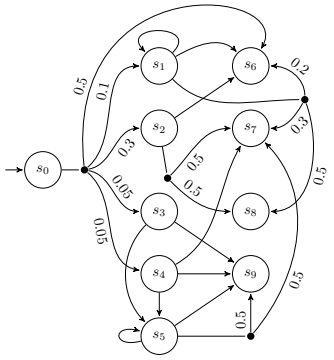


Fig. 2: A sample PA $\mathcal{M}$

[3] As this paper does not cover parallel composition all PGAs are closed. For modeling PGAs in a compositonal manner though, the distinction between internal and other actions is important, see [7].

**Definition 3 (Embedding a PA into an PGA[6]).** *PA* $\mathcal{M} = (S, A, \Delta, s_0)$ induces *the PGA* $\alpha_{\mathrm{PA}}(\mathcal{M}) = (S', \{S'_1, S'_2\}, A, \Delta', (s_0, 1))$ *with* $S'_1 = S \times \{1\}$, $S'_2 = S \times \{2\}$ *and for every* $s \in S$*:*

1. $(s,1) \xrightarrow{a} \mu'$ *iff* $s \xrightarrow{a} \mu$ *and* $\mu'(u,2) = \mu(u)$*, and*
2. $(s,2) \xrightarrow{a} \mu'$ *with* $\mu'(s,1) = 1$ *iff for some* $u \in S$ *with* $u \xrightarrow{a} \mu$ *and* $s \in \mathrm{supp}(\mu)$.

### 2.3 Simulation relations

Simulation relations for probabilistic systems are pre–orders requiring that whenever state $u$ simulates state $s$, then $u$ can at least mimic the stepwise behaviour of $s$. They can be computed for finite models by reducing them to network max-flow problems [15]. They are lifted to distributions over states as follows:

**Definition 4 (Simulation relation [16]).** *Let* $S$ *be a countable, non-empty set of states, and let* $\mu, \mu' \in \mathrm{Dist}(S)$*. For* $R \subseteq S \times S$*,* $\mu'$ *simulates* $\mu$ *w.r.t.* $R$*, denoted* $\mu R \mu'$*, iff there exists a function* $\delta : S \times S \to [0,1]$ *such that for all* $u, v \in S$*: (1)* $\delta(u,v) > 0 \Rightarrow uRv$*, (2)* $\sum_{s \in S} \delta(u,s) = \mu(u)$*, and (3)* $\sum_{s \in S} \delta(s,v) = \mu'(v)$.

We define two simulation relations on PGAs: *simulation* and *alternating simulation*. Simulation relations compare reachability probabilities in case of collaborating players (i.e., both players want to maximize/minimize reachability probabilities), whereas alternating simulation relations do so in case of competing players.

**Definition 5 (Simulation on PGAs [6]).** $R \subseteq \bigcup_{j \in \{1,2\}} S_j \times S_j$ *is a* simulation *relation on PGA* $\mathcal{G}$ *iff for every* $sRs'$*,* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a} \mu'$ *with* $\mu R \mu'$*. Let* $\prec$ *be the largest simulation relation.*

**Definition 6 (Alternating simulation on PGAs [6]).** $R \subseteq \bigcup_{j \in \{1,2\}} S_j \times S_j$ *is an* alternating simulation *relation on PGA* $\mathcal{G}$ *iff for every* $sRs'$ *the following holds: (1) if* $s, s' \in S_1$*, then* $s' \xrightarrow{a} \mu'$ *implies* $s \xrightarrow{a} \mu$ *such that* $\mu R \mu'$*, (2) if* $s, s' \in S_2$*, then* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a} \mu'$ *such that* $\mu R \mu'$*. Let* $\preccurlyeq$ *be the largest alternating simulation relation. We write "*$s'$ alt-simulates $s$*" iff* $s \preccurlyeq s'$.

Intuitively, in case of player-one states, the behaviour of $s'$ is mimicked by that of $s$; whereas in case of player-two states, it is the other way round.

We write $\mathcal{G} \prec \mathcal{G}'$ ($\mathcal{G} \preccurlyeq \mathcal{G}'$) if $s_0 \prec s'_0$ ($s \preccurlyeq s'_0$), where $\prec$ ($\preccurlyeq$) is taken on the disjoint union of $\mathcal{G}$ and $\mathcal{G}'$. By the following theorem, $\mathcal{G} \prec \mathcal{G}'$ ($\mathcal{G} \preccurlyeq \mathcal{G}'$) implies that $\mathcal{G}'$ bounds $\Pr^{++}$ ($\Pr^{+-}$) and $\Pr^{--}$ ($\Pr^{-+}$) values of $\mathcal{G}$ from above (below) and below (above) in case of collaborating (competing) players.

**Theorem 1.** *For* PGA $\mathcal{G}$ *and* $\mathcal{G}'$*, and* $T \subseteq S$*. Then:*

1. $\mathcal{G} \prec \mathcal{G}'$ *implies* $\Pr^{--}(T') \leq \Pr^{--}(T)$ *and* $\Pr^{++}(T) \leq \Pr^{++}(T')$*, and*
2. $\mathcal{G} \preccurlyeq \mathcal{G}'$ *implies* $\Pr^{-+}(T'') \geq \Pr^{-+}(T)$ *and* $\Pr^{+-}(T) \geq \Pr^{+-}(T'')$

*where* $T' = \{s' \in S' \mid \exists s \in T : s \prec s'\}$ *and* $T'' = \{s' \in S' \mid \exists s \in T : s \preccurlyeq s'\}$.

## 3 Modal Stochastic Games

This section presents an extension of PGAs by annotating their transitions with *required* (must) and *possible* (may) modalities as in modal transition systems [17]. This results in *abstract* probabilistic game automata (APGAs, for short). The semantics of an APGA is a *set* of PGAs, namely all PGAs that have at least all required transitions and zero or more possible transitions. These games are called *implementations* of APGA.

**Definition 7 (Abstract PGA).** *An* abstract PGA *(APGA) is a tuple* $\mathcal{H} = (S, \{S_1, S_2\}, A, \Delta_{\mathrm{r}}, \Delta_{\mathrm{p}}, s_0)$ *with* $S$, $S_1$, $S_2$, $A$, *and* $s_0$ *as in PGA,* $\Delta_{\mathrm{p}} \subseteq S_{1+x} \times A \times \mathrm{Dist}(S_{2-x})$ *is a set of* possible *transitions and* $\Delta_{\mathrm{r}} \subseteq S_{1+x} \times A \times \mathrm{Dist}(S_{2-x})$ *is a set of* required *transitions with* $\Delta_{\mathrm{r}} \subseteq \Delta_{\mathrm{p}}$, *where* $x \in \{0, 1\}$.

We denote $(s, a, \mu) \in \Delta_y$ by $s \xrightarrow{a}_y \mu$, and transitions emanating from a state $s$ as $\Delta_y(s) = \{(s, a, \mu) \mid s \xrightarrow{a}_y \mu\}$ for $y \in \{\mathrm{p}, \mathrm{r}\}$. Every PGA is an APGA with $\Delta_{\mathrm{r}} = \Delta_{\mathrm{p}}$. We depict required transitions as solid lines, and others as dotted lines (see Fig. 3). Let closed APGA be defined in a similar way as closed PA (page 4). In the sequel, let $\mathcal{H} = (S, \{S_1, S_2\}, A, \Delta_{\mathrm{r}}, \Delta_{\mathrm{p}}, s_0)$ be a finitely branching APGA.

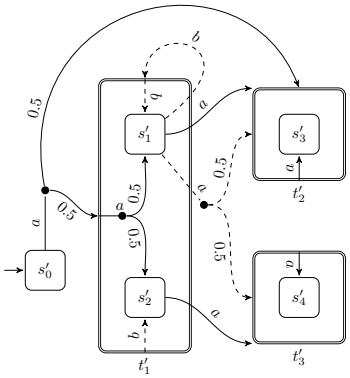


Fig. 3: A sample APGA $\mathcal{H}$

APGAs are compared using *refinement relations*. Intuitively, when a state $s$ refines a state $s'$, then $s'$ mimics at least the step-wise *possible* behaviour of $s$, whereas $s$ mimics at least the step-wise *required* behaviour of $s'$. A special class of refinement relations, called *satisfaction relations*, relates implementations (concrete models, i.e., PGAs) with APGA (specifications). In the sequel, let $S_j$ be the set of states of player $i$ in PGA $\mathcal{G}$ (APGA $\mathcal{H}$), and $S'_j$ be its set of states in the APGA $\mathcal{H}'$.

**Definition 8 (Satisfaction relation).** $R \subseteq \bigcup_{j \in \{1,2\}} S_j \times S'_j$ *is a* satisfaction *relation between PGA* $\mathcal{G}$ *and APGA* $\mathcal{H}'$ *iff for* $sRs'$, *(1)* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a}_{\mathrm{p}} \mu'$ *such that* $\mu R \mu'$, *and (2)* $s' \xrightarrow{a}_{\mathrm{r}} \mu'$ *implies* $s \xrightarrow{a} \mu$ *such that* $\mu R \mu'$. *Let* $\models$ *be the largest satisfaction relation.*

The set of implementations of APGA $\mathcal{H}$ is defined by $\wr \mathcal{H} \wr = \{\mathcal{G} \mid \mathcal{G} \models \mathcal{H}\}$.

**Definition 9 (Refinement relation).** $R \subseteq \bigcup_{j \in \{1,2\}} S_j \times S'_j$ *is a* refinement *relation between APGA* $\mathcal{H}$ *and* $\mathcal{H}'$ *iff for* $sRs'$, *(1)* $s \xrightarrow{a}_{\mathrm{p}} \mu$ *implies* $s' \xrightarrow{a}_{\mathrm{p}} \mu'$ *such that* $\mu R \mu'$, *and (2)* $s' \xrightarrow{a}_{\mathrm{r}} \mu'$ *implies* $s \xrightarrow{a}_{\mathrm{r}} \mu$ *such that* $\mu R \mu'$. *Let* $\preceq$ *be the largest refinement relation.*

The conditions (1) and (2) are the same as in Def. 8 except that in (1) the transition from $s$ is a *possible* transition, whereas in (2) the transition from $s$ is a *required* transition.

*Example 1.* $R = \bigcup_{i=1\ldots3}(t_i, t'_i) \cup \bigcup_{i=0\ldots4}(s_i, s'_i)$ is a refinement (in fact, a satisfaction) relation between PGA $\mathcal{G}$ (Fig. 1) and APGA $H$ (Fig. 3).

**Proposition 1.** *$\preceq$ is a pre–order.*

**Extremal implementations.** We focus on two special implementations of APGA $\mathcal{H}$, denoted $\mathcal{G}^{\mathrm{p}}$ and $\mathcal{G}^{\mathrm{r}}$, and show that they bound the optimal reachability probabilities of every implementation of $\mathcal{H}$. We call $\mathcal{G}^{\mathrm{p}}$ and $\mathcal{G}^{\mathrm{r}}$ *extreme* PGAs (EPGAs, for short). Both $\mathcal{G}^{\mathrm{p}}$ and $\mathcal{G}^{\mathrm{r}}$ inherit the player-two transitions from its possible transitions in $\mathcal{H}$. They differ for player one, though. EPGA $\mathcal{G}^{\mathrm{p}}$ inherits its player-one transitions (denoted by the superscript) from the possible transitions of player one in $\mathcal{H}$, whereas $\mathcal{G}^{\mathrm{r}}$ inherits its player-one transitions from the required transitions in $\mathcal{H}$.

**Definition 10 (Extremal PGAs implementations).** *For $y \in \{\mathrm{p}, \mathrm{r}\}$, $\mathcal{G}^y$ is an* EPGA *of $\mathcal{H}$ iff $S$, $S_1$, $S_2$, $A$ and $s_0$ in $\mathcal{G}^y$ are as in $\mathcal{H}$, $\Delta(s) = \Delta_{\mathrm{p}}(s)$ for $s \in S_2$, and $\Delta(s) = \Delta_y(s)$ for $s \in S_1$.*

In the sequel, $\mathcal{G}^y = (S, \{S_1, S_2\}, A, \Delta, s_0)$ is an EPGA of $\mathcal{H}$ for $y \in \{\mathrm{p}, \mathrm{r}\}$.

**Proposition 2.** *For every $\mathcal{G} \in \wr\mathcal{H}\wr$, it holds $\mathcal{G} \prec \mathcal{G}^{\mathrm{p}}$ and $\mathcal{G} \preccurlyeq \mathcal{G}^{\mathrm{r}}$.*

By Th. 1 and Prop. 2, EPGAs suffice for the optimal reachability analysis of $\mathcal{H}$. Note that the two extreme implementations by considering the required (as opposed to the possible) transitions of player two are simulated and alt-simulated by $\mathcal{G}^{\mathrm{p}}$ and $\mathcal{G}^{\mathrm{r}}$ respectively.).

**Proposition 3.** *$\mathcal{H}_1 \preceq \mathcal{H}_2$ implies (1) $\mathcal{G}_1^{\mathrm{p}} \prec \mathcal{G}_2^{\mathrm{p}}$ and (2) $\mathcal{G}_1^{\mathrm{r}} \preccurlyeq \mathcal{G}_2^{\mathrm{r}}$.*

It follows from Prop. 2, 3 and Th. 1 that if $\mathcal{H}_1 \preceq \mathcal{H}_2$, then $\mathcal{H}_2$ bounds the extremal reachability probabilities in $\mathcal{H}_1$.

## 4 Abstraction

This section presents our abstraction technique, a combination of abstraction of PA using modalities [7] and game-based abstraction [5]. It is based on partitioning the state space such that player-one and player-two states are kept separate. The key principle is that player-one states that have the same set of transitions (after abstraction) must at least be assigned to the same abstract state. Every transition from a concrete state, either belonging to player-one or two, becomes a *possible* transition from its corresponding abstract state. For player-one states we additionally apply the following approach. An abstract player-one state is equipped with a required $a$-transition to distribution $\mu'$ iff every of its concrete states has a required $a$-transition to $\mu$ such that $\mu'$ is the abstract counterpart

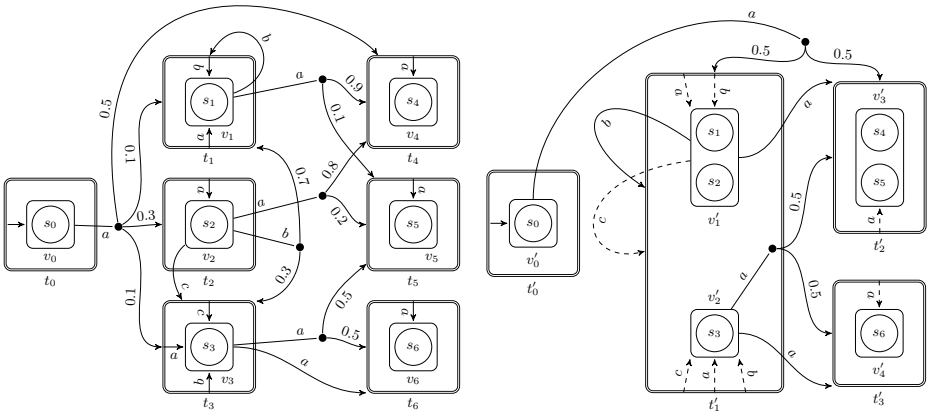


Fig. 4: The embedding $\mathcal{H}$ (left) of the PA in Fig. 2, and its abstraction $\mathcal{H}' = \alpha(\mathcal{H})$ (right)

of $\mu$. Required transitions for player-two states are not detailed further, as they play no role in the analysis of optimal reachability probabilities (see Prop. 2).

Let $\alpha : S \to S'$ be an abstraction (a surjective function) and $\gamma : S' \to 2^S$ be the corresponding concretization function. That is, $\alpha(s)$ is the abstract state of $s$ whereas $\gamma(s')$ is the set of concrete states abstracted by $s'$. The abstraction of distribution $\mu$ is given as $\alpha(\mu)(s') = \mu(\gamma(s'))$. The functions $\alpha$ and $\gamma$ are lifted to sets of states or sets of distributions in a point-wise manner.

**Definition 11 (Abstraction).** *For APGA $\mathcal{H}$, the* abstraction function $\alpha : S \to S'$ *induces the APGA $\mathcal{H}' = \alpha(\mathcal{H})$ if the following conditions are satisfied: $A' = A$; $S'_i = \alpha(S_i)$ for $i \in \{1,2\}$; $\forall s,u \in S_1 : \alpha(\Delta_y(s)) = \alpha(\Delta_y(u))$ for $y \in \{\mathrm{p},\mathrm{r}\}$ implies $\alpha(s) = \alpha(u)$; and for every $s' \in S'$:*

1. *$s' \in S'_1$ implies $s' \xrightarrow{a}_{\mathrm{r}} \mu'$ iff $\forall s \in \gamma(s') : s \xrightarrow{a}_{\mathrm{r}} \mu$ such that $\alpha(\mu) = \mu'$,*
2. *$\exists s \in \gamma(s') : s \xrightarrow{a}_{\mathrm{p}} \mu$ implies $s' \xrightarrow{a}_{\mathrm{p}} \mu'$ such that $\alpha(\mu) = \mu'$, and*
3. *$s' \xrightarrow{a}_{\mathrm{p}} \mu'$ implies $\exists s \in \gamma(s') : s \xrightarrow{a}_{\mathrm{p}} \mu$ such that $\alpha(\mu) = \mu'$.*

In the sequel, $\alpha$ denotes an *abstraction* function. Our framework considers abstractions of APGAs. For simplicity, all examples consider the abstractions of PAs.

*Example 2.* Let $\mathcal{H}' = \alpha(\mathcal{H})$ in Fig. 4 (right) be the induced abstract model of APGA $\mathcal{H}$ (left) with $\gamma(t'_0) = \{t_0\}$, $\gamma(t'_1) = \{t_1, t_2, t_3\}$, $\gamma(t'_2) = \{t_4, t_5\}$ and $\gamma(t'_3) = \{t_6\}$ as well as $\gamma(v'_0) = \{v_0\}$, $\gamma(v'_1) = \{v_1, v_2\}$, $\gamma(v'_2) = \{v_3\}$, $\gamma(v'_3) = \{v_4, v_5\}$ and $\gamma(v'_4) = \{v_6\}$. Let us consider the abstract state $v'_1$, it has a *required* $a$-transition to $t'_2$ because both of its concrete states ($v_1$ and $v_2$) have *required* $a$-transitions with target distributions over $t_4$ and $t_5$ (the concrete states of $t'_2$). By a similar reason there exists a *required* $b$-transition from $v'_1$ to $t'_1$. However, only $v_2$ has a *required* $c$-transition to $t_3$, therefore, $v'_1$ has a *possible* $c$-transition

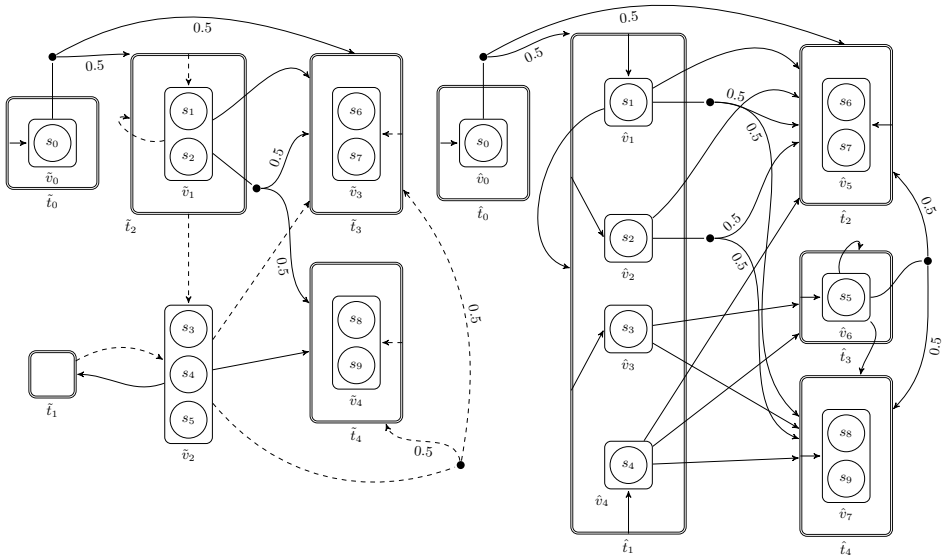


Fig. 5: For PA $\mathcal{M}$ (Fig. 2), APGA-based abstraction $\tilde{\mathcal{H}} = \alpha(\alpha_{\mathrm{PA}}(\mathcal{M}))$ (left) with $|\tilde{\Delta}| = 17$, $|\tilde{S}_1| = 5$ and $|\tilde{S}_2| = 5$; and game-based abstraction $\hat{\mathcal{H}} = \alpha(\alpha_{\mathrm{PA}}(\mathcal{M}))$ (right) with $|\hat{\Delta}| = 26$, $|\hat{S}_1| = 8$ and $|\hat{S}_2| = 5$.

to $t_1'$ (the abstract state of $t_3$). Note that the incoming transition of state $v_0$ indicates that $v_0$ is initial; there is no transition from $t_0$ to $v_0$. The rest of the example is self-explanatory.

The proposition below establishes that concrete models refine their abstractions; therefore, by Th. 1 and Prop. 3, their reachability probabilities are bounded by those of their abstractions.

**Proposition 4.** $\mathcal{H} \preceq \alpha(\mathcal{H})$.

Prop. 4 and the corollary below (that follows from Def. 11) prove that APGA-based abstractions yield *at most as tight bounds* as SG-based abstractions, whereas they are *at most* the sizes of SG-based abstractions. (Note that in order to compare concrete with abstract models (in terms of their sizes), we take the sizes of probabilistic transitions equal to the cardinality of the support sets of their target distributions, e.g., the size of a transition $s \xrightarrow{a} \mu$ is equal to $|\mathrm{supp}(\mu)|$.)

**Corollary 1.** *Let $\mathcal{H}_{sg}$ be an SG-based abstraction and $\mathcal{H}_{apga}$ be an APGA-based abstraction of PA $\mathcal{M}$ with $S_2^{sg} = S_2^{apga}$. Then: (1) $|S_1^{sg}| \geq |S_1^{apga}|$, and (2) $\mathcal{H}_{sg} \preceq \mathcal{H}_{apga}$.*

*Example 3.* Consider the game-based abstraction $\hat{\mathcal{H}}$ (Fig. 5 right) of PA $\mathcal{M}$ (Fig. 2). The maximum probability to reach states $\{s_6, s_7\}$ lies in $[0.75, 1]$ in $\hat{\mathcal{H}}$

whereas in APGA-based abstraction $\tilde{\mathcal{H}}$ (Fig. 5 left), it lies in $[0.5, 1]$. Note that both $\tilde{S}_2$ and $\hat{S}_2$ represent the same partitioning of the concrete state space, the reachability probability bounds of $\tilde{S}_2$ states contain that of $\hat{S}_2$ states, and $\tilde{\mathcal{H}}$ is smaller than $\hat{\mathcal{H}}$.

Sher [18] defines a composition operator in a TCSP-like manner for the class of APGAs representing abstract models of PAs, and shows that our abstraction technique is compositional.

## 5 Iterative Abstraction-Refinement

The key idea of our abstract-refine framework (see Fig. 6) is to separate the *iterative* refinement of player-one and player-two states. It *automatically* generates APGA-based abstractions of (closed) PAs with a finite bisimulation quotient in which the bounds on probabilities for reaching a set of goal states are within the allowed range.

The input is a closed PA, a reachability property (max/min probability to goal states) and an error bound $\epsilon \in \mathbb{R}_{(0,1)}$. Starting from an initial abstraction (obtained by partitioning $S_1$ and $S_2$ states in the embedding of PA), we incrementally refine player-one states (yielding a new partitioning for the player-one state space) until the reachability probability bounds of player-two states stabilize. Next, we check whether the probability bounds of a set of player-two states (that are of interest) are within the allowed range $\epsilon$. If not, some of the player-two states are refined yielding a new partitioning of the concrete state space — recall that $S_1$ states having the same set of transitions under a given partitioning of $S_2$ states are at least assigned to the same abstract state (see Def. 11). The first step is then repeated for the new abstract model. The above two steps form the inner and the outer loop, that refine player-one and player-two states respectively, of our abstract-refine framework.

Our *refinement strategy* is based on optimal probability valuation functions. It induces a strictly finer partition in each iteration, and thus makes the nested loop eventually terminate for PA having a finite bisimulation quotient.

Let $\mathcal{M}$ be a closed finite PA, having a finite bisimulation quotient, with its embedding $\mathcal{H}' = \alpha_{\mathrm{PA}}(\mathcal{M})$ and set of goal states $T' \subseteq S'_2$. Let $\Pr^x(T')$ be the probability for reaching states in $T'$ from the initial state $s_0$, where $x \in \{\min, \max\}$. Let $Abst(\mathcal{H}')$ be the set of abstraction functions defined on $\mathcal{H}'$ such that $\gamma(\alpha(T')) = T'$ for all $\alpha \in Abst(\mathcal{H}')$, i.e., $\alpha$ does not merge $T'$ states with $S'_2 \backslash T'$ states. Let $\mathcal{H} = \alpha(\mathcal{H}')$ and $T = \alpha(T')$ for $\alpha \in Abst(\mathcal{H}')$.

Depending on the property $\Pr^x(T')$, let $\mathbf{1}, \mathbf{2} \in \{\min, \max\}$ with $\mathbf{1} \neq \mathbf{2}$ [4]. Let $w_{\mathbf{11}}, w_{\mathbf{12}} \in W$ be the probability valuation functions (see Def. 2) such that $w_{\mathbf{12}} = \mathbf{Fix}\ \mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\bot)$ and $w_{\mathbf{11}} = \mathbf{Fix}\ \mathrm{Prt}^{\mathbf{1}}_{\mathbf{1}}(\bot)$ (both players have the same objective, i.e., $\mathbf{1}$) are defined on EPGA $\mathcal{G}^{\mathrm{p}}$ of $\mathcal{H}$ for the set of goal states $T$. Thus, $w_{\mathbf{12}}/w_{\mathbf{11}}$ maps a state $s \in S$ to the probability of reaching $T$ in case of competing/collaborating

[4] For example, let $x = \max$ in $\Pr^x(T')$ then $\mathbf{1} = \max$ and $\mathbf{2} = \min$ (player-one maximizes whereas the player-two minimizes the probability) or vice versa.

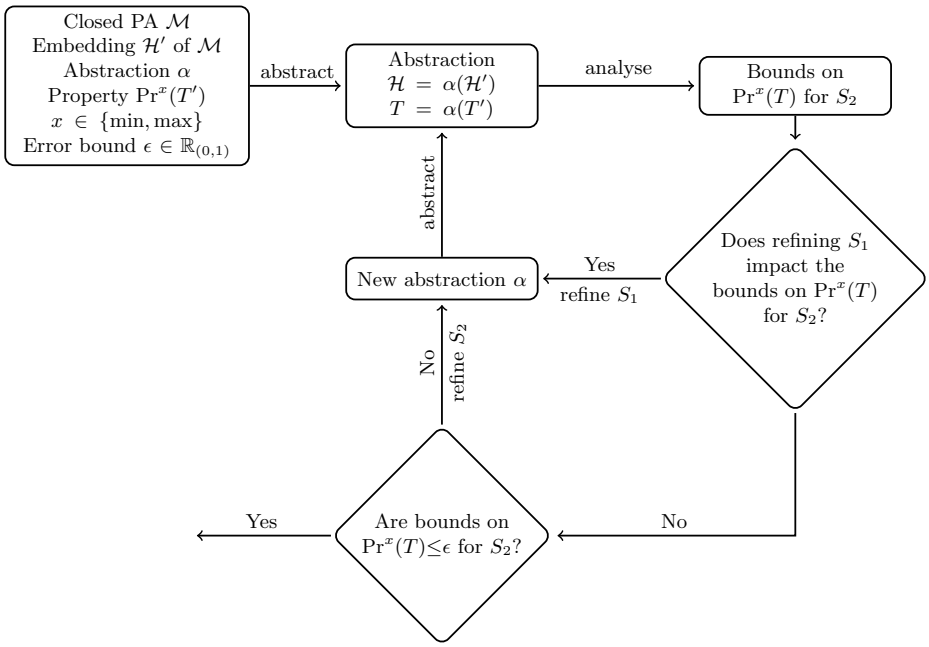


Fig. 6: Abstraction-refinement framework for closed PAs

players; and therefore define bounds on $\Pr^x(T')$ for the initial state in $\mathcal{H}'$. In the sequel, we assume $(\alpha, \gamma)$, $\mathcal{H}'$, $\mathcal{H}$, $\mathbf{1}$, $\mathbf{2}$, $w_{\mathbf{11}}$ and $w_{\mathbf{12}}$ are given; unless stated otherwise. Moreover, let $\Delta_y(s) = \{\mu \mid s \xrightarrow{\tau}_y \mu\}$ for $y \in \{\mathrm{p}, \mathrm{r}\}$.

### 5.1 Stable abstractions

We now explain our abstract-refine framework (Fig. 6). We only consider states $s$ with $\Delta_{\mathrm{p}}(s) \neq \emptyset$ for refinement, as only their refinement can affect the reachability probabilities.

We first check whether the probabilities for reaching goal states from player-two states in $\mathcal{H}$ depend on the non-determinism induced by the abstraction process in their successor (player-one) states. Alternatively, we check whether the splitting of player-one states (alone) affects the reachability probabilities of their corresponding player-two states. (Recall we allow to merge concrete player-one states even if their behaviour after abstraction is not the same (see Def. 11), therefore their splitting may change the reachability probabilities of their corresponding player-two states.). Let us first define some notions.

State $t \in S_2$ in APGA $\mathcal{H}$ is called *stable* whenever the value $w_{\mathbf{12}}(t)$ (a) coincides with that of one of its direct successors that obtains it via a *required* transition, and (b) remains unchanged

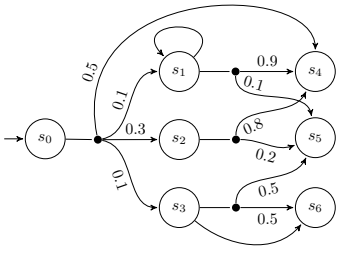


Fig. 7: A PA $\mathcal{M}$.

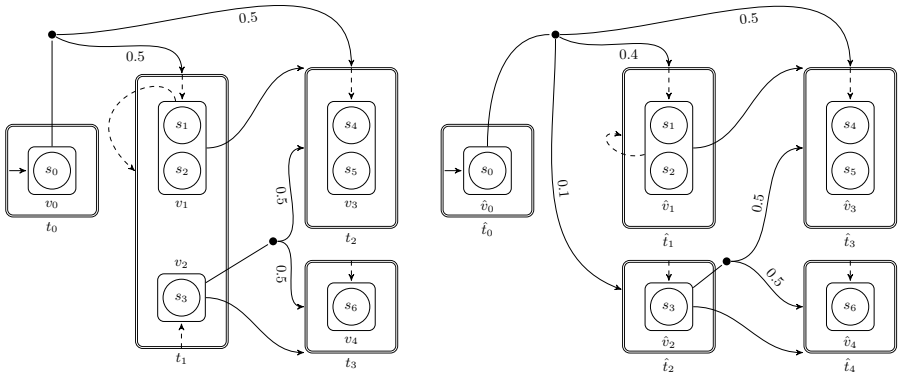


Fig. 8: For $\mathbf{1} = \min$ and $\mathbf{2} = \max$, APGA $\mathcal{H}$ (left) is a stable abstraction of PA $\mathcal{M}$ (Fig. 7) w.r.t. $T = \{t_3\}$; and APGA $\hat{\mathcal{H}}$ (right) is a bounded abstraction w.r.t. $T = \{\hat{t}_4\}$ and $\epsilon = 0.4$.

after splitting its direct successor states. To formally define this notion, we first define the notion of a *stable* player-one state. A state $s \in S_1$ is *stable* if its reachability probability $w_{\mathbf{12}}(s)$ is obtained via some of its *required* transitions.

**Definition 12 (Stable player-one states).** *State $s \in S_1$ is* stable *iff $w_{\mathbf{12}}(s) = w_{\mathbf{12}}(\mu)$ for some $\mu \in \Delta_{\mathrm{r}}(s)$. States that are not stable are unstable.*

*Example 4.* APGA $\mathcal{H}$ (Fig. 8 left) is an abstraction of PA $\mathcal{M}$ (Fig. 7). Let $\mathbf{1} = \min$, $\mathbf{2} = \max$, $T = \{t_3\}$ with $w = \mathbf{Fix}\ \mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\bot)$ where $w(v_0) = 0.25$, $w(v_1) = 0$, $w(v_2) = 0.5$, $w(v_3) = 0$, $w(v_4) = 0$, $w(t_0) = 0$, $w(t_1) = 0.5$, $w(t_2) = 0$ and $w(t_3) = 1$. Note that $\Delta_{\mathrm{r}}(v_1) \neq \emptyset$ and $\Delta_{\mathrm{p}}(v_1) \neq \emptyset$. As $w(v_1) = w(\iota_{t_2}) = 0$, and $\iota_{t_2} \in \Delta_{\mathrm{r}}(v_1)$, $v_1$ is stable.

**Proposition 5.** *Refining stable player-one states preserves reachability probabilities.*

Intuitively, if the reachability probability (w.r.t. $w_{\mathbf{12}}$) of a player-two state, say $t$, depends on one of its stable successors, it remains unchanged if any of them is split. This is because a stable (player-one) state, say $s$, obtains its reachability probability via a *required* transition. And if $s$ is split, then the partitions of $s$ inherit the *required* transitions of $s$; as a result the reachability probabilities of partitions of $s$ remain unchanged – because they obtain them via the same *required* transition as $s$. Thus, in the refined model the reachability probability of $t$ again depends on one of its stable successors, and remains unchanged. This is not ensured if an unstable successor, say $u$, of $t$ is split. Because in this case different partitions of $u$ might have different sets of *required* and *possible* transitions, possibly resulting in different reachability probabilities in the refined model. Now if the reachability probability of $t$ depends on one of them, it might be different from that in the abstract model.

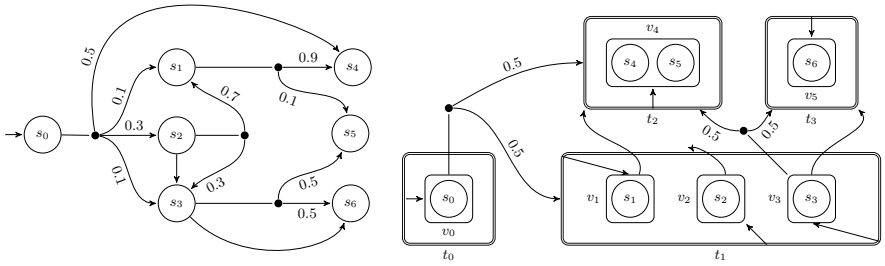


Fig. 9: PA $\mathcal{M}$ (left) with its SG-based abstraction $\mathcal{H}$ (right).

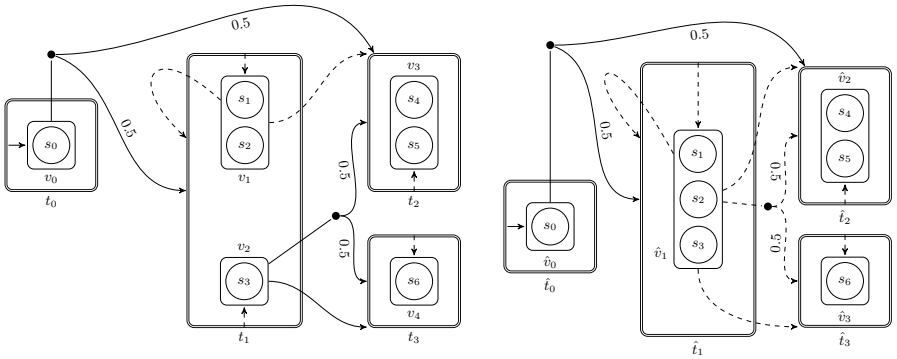


Fig. 10: $\mathcal{H}$ (left) and $\hat{\mathcal{H}}$ (right) are abstract models of PA $\mathcal{M}$ in Fig. 9 with $\mathcal{H} \preceq \hat{\mathcal{H}}$.

In the following definition, we state conditions that guarantee the preservation of reachability probability w.r.t. $w_{\mathbf{12}}$ of a player-two state irrespective of whether its stable or unstable successor is refined.

**Definition 13 (Stable player-two states).** *State $t \in S_2$ is* stable *iff (1) $w_{\mathbf{12}}(t) = w_{\mathbf{12}}(v)$ for a stable $v \in \mathrm{succ}(t)$, and (2) $\forall u \in \mathrm{succ}(t)$: $w_{\mathbf{12}}(t) = \mathbf{2}\{w_{\mathbf{12}}(v), w_{\mathbf{12}}(\eta)\}$ for every $\eta \in \Delta_{\mathrm{p}}(u)$. States that are not stable are unstable.*

Condition (1) assures that the reachability probability of $t$ depends on a stable successor. Condition (2) assures that $w_{\mathbf{12}}(t)$ will remain unchanged even if the successor states of $t$ are split into their constituent states.

*Example 5.* Let $\mathbf{1} = \min$ and $\mathbf{2} = \max$ for APGA $\mathcal{H}$ (left in Fig. 10) with $w = \mathbf{Fix}\ \mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\bot)$ for $T = \{t_3\}$, where $w(v_0) = 0.25$, $w(v_1) = 0$, $w(v_2) = 0.5$, $w(v_3) = 0$, $w(v_4) = 0$, $w(t_0) = 0$, $w(t_1) = 0.5$, $w(t_2) = 0$ and $w(t_3) = 1$. (Note that this APGA is a copy of Fig. 8 left, except that $v_1 \to t_2$ is now a *possible* transition.) The reachability probability of $t_1$ depends on a stable successor $v_2$, i.e., $w(t_1) = 0.5 = \max\{w(v_1) = 0, w(v_2) = 0.5\}$ (fulfilling condition (1) of Def. 13). Moreover, as $w(v_2) = 0.5 = \max\{w(\iota_{t_1}) = 0.5, w(\iota_{t_2}) = 0, w(v_2) = 0.5\}$ (fulfilling condition (2) of Def. 13), therefore the *possible* transitions of unstable state $v_1$ have no impact on the reachability probability of $t_1$ in any refinement

of $v_1$. Thus, $t_1$ is stable. Note that in APGA $\hat{\mathcal{H}}$ (right in Fig. 10), the state $\hat{t}_1$ is not stable w.r.t. the above objectives as the reachability probability of $\hat{t}_1$ does not depend on a stable successor.

**Proposition 6.** *The reachability probabilities of stable player-two states are invariant to the refinement of their direct successors.*

An APGA $\mathcal{H} = \alpha(\mathcal{H}')$ is *stable* if all player-two states $t$ with $\Delta_{\mathrm{p}}(t) \neq \emptyset$ are stable; we call $\alpha$ a *stable abstraction function*. Any refinement of a stable abstraction, with the same player-two state space, preserves reachability probabilities. Therefore, if further tightening of probability bounds is required, we should consider refining player-two states (see Fig. 6). First we discuss the refinement of player-one states.

### 5.2 Refining player-one states

We consider unstable successors of unstable player-two states for refinement.

**Definition 14 (Effective unstable).** *State $s \in S_1$ is* effectively unstable *iff (1) $s$ is unstable, and (2) there exists an unstable $t \in S_2$ with $s \in \mathrm{succ}(t)$.*

Let $\mathrm{eus}(\mathcal{H})$ be the set of effectively unstable states. We define how to split an effectively unstable state in $\mathcal{H}$ into two blocks yielding a new partitioning of the state space of $\mathcal{H}'$. (Recall $\mathcal{H}' = \alpha_{\mathrm{PA}}(\mathcal{M})$, and $\mathcal{H} = \alpha(\mathcal{H}')$)

**Definition 15.** *For $s \in \mathrm{eus}(\mathcal{H})$, let $\mu \in \Delta_{\mathrm{p}}(s) : w_{\mathbf{12}}(s) = w_{\mathbf{12}}(\mu)$. Then, $\mathrm{B}_1(s) = \{s' \in \gamma(s) \mid \exists \rho' \in \Delta'(s') : \alpha(\rho') = \mu\}$ and $\mathrm{B}_2(s) = \gamma(s) \backslash \mathrm{B}_1(s)$.*

This is the basis for the inner-loop of our abstract-refine framework (Fig. 6).

**Definition 16 (Inner abstraction).** *The* inner abstraction transformer *function* $\mathrm{IAT} : \mathit{Abst}\ (\mathcal{H}') \to \mathit{Abst}(\mathcal{H}')$ *is defined for $\alpha \in \mathit{Abst}(\mathcal{H}')$ with $\mathcal{H} = \alpha(\mathcal{H}')$ and $s' \in S'$ as:*

$$\mathrm{IAT}(\alpha)(s') = \begin{cases} \alpha(s') & \text{if } \alpha(s') \in S_2, \text{ or } \alpha(s') \in S_1 \backslash \mathrm{eus}(\mathcal{H}) \\ \mathrm{B}_1(\alpha(s')) & \text{if } \alpha(s') \in \mathrm{eus}(\mathcal{H}) \text{ and } s' \in \mathrm{B}_1(\alpha(s')) \\ \mathrm{B}_2(\alpha(s')) & \text{if } \alpha(s') \in \mathrm{eus}(\mathcal{H}) \text{ and } s' \in \mathrm{B}_2(\alpha(s')) \end{cases}$$

Note that $\mathrm{IAT}(\alpha)$ maps $s'$ to the same partition block as $\alpha$ does if either $\alpha(s')$ is a player-two or a stable player-one state. In case $\alpha(s') = s$ is an effectively unstable state, it is either mapped to the partition block $\mathrm{B}_1(s)$ or $\mathrm{B}_2(s)$.

*Example 6.* APGA $\hat{\mathcal{H}}$ (right Fig. 10) is an abstraction of PA $\mathcal{M}$ (left in Fig. 9). Let $\mathbf{1} = \min$ and $\mathbf{2} = \max$ for $\hat{\mathcal{H}}$ with $\hat{w} = \mathbf{Fix}\ \mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}(\bot)$ for $\hat{T} = \{\hat{t}_2\}$, where $\hat{w}(\hat{v}_0) = 0.5$, $\hat{w}(\hat{v}_1) = 0$, $\hat{w}(\hat{v}_2) = 0$, $\hat{w}(\hat{v}_3) = 0$, $\hat{w}(\hat{t}_0) = 0$, $\hat{w}(\hat{t}_1) = 0$, $\hat{w}(\hat{t}_2) = 1$ and $\hat{w}(\hat{t}_3) = 0$. Note that $\hat{t}_1$ has only one successor, i.e. $\hat{v}_1$, having only *possible* transitions. Therefore, $\hat{\mathcal{H}}$ is not a stable abstraction of PA $\mathcal{M}$.

Let us refine $\hat{\mathcal{H}}$, and let $\mathcal{H}' = \alpha_{\mathrm{PA}}(\mathcal{M})$. For the successor state $\hat{v}_1$ of $\hat{t}_1$, we have $\hat{v}_1 \to \iota_{\hat{t}_3}$ with $\hat{w}(\iota_{\hat{t}_3}) = \hat{w}(\hat{v}_1) = 0$. We separate the concrete states of $\hat{v}_1$ that have a transition (after abstraction) to $\iota_{\hat{t}_3}$, which is $v'_3$. Therefore, $\hat{v}_1$ is

partitioned into two blocks $v_1 = \{v'_1, v'_2\}$ and $v_2 = \{v'_3\}$; and $\mathcal{H}$ (left in Fig. 10) is the APGA induced by the new partitioning of the state space of $\mathcal{H}'$. Note that $\mathcal{H}$ is a stable abstraction w.r.t. objectives **1**, **2** and $T = \{t_2\}$; and moreover $\mathcal{H} \preceq \hat{\mathcal{H}}$.

Instead of refining all states in $\text{eus}(\mathcal{H})$ in one step, one may pick some of them. In this way, unnecessary refinements of some states in $\text{eus}(\mathcal{H})$ in the next iteration may be avoided (because of splitting of states in the current step).

**Proposition 7.** $\text{IAT}(\alpha) \preceq \alpha$ *for* $\alpha \in Abst(\mathcal{H}')$

The fixpoint of the function IAT is guaranteed to exist for abstraction functions defined (on the embedding of) PAs with finite bisimulation quotient. Intuitively, because of the finite number of player-one states and transitions, the refinement process (in the worst case) will eventually result in a model having only *required* transitions from player-one states. At that point, all player-one states will be stable, thus, making their further partitioning impossible (see Def. 16). This provides the basis to iteratively refine player-one states in $\alpha(\mathcal{H}')$ resulting in a stable abstraction **Fix** $\text{IAT}(\alpha)(\mathcal{H}')$ of $\mathcal{H}'$.

**Theorem 2. Fix** $\text{IAT}(\alpha)(\mathcal{H}')$ *is a stable abstraction.*

The following corollary follows from Th. 2, and shows that for a given partitioning of states of PA $\mathcal{M}$, the SG-based abstraction [5] is as precise as the APGA-based abstraction when refined to a stable abstraction. However, the size of the latter is at most that of the former.

**Corollary 2.** *Let* $\mathcal{H}_{sg}$ *be an SG-based abstraction and* $\alpha(\mathcal{H}') = \hat{\mathcal{H}}_{apga}$ *be an APGA-based abstraction of APGA* $\mathcal{H}'$ *with* $S_2^{sg} = \hat{S}_2^{apga}$. *Let* $w_{\mathbf{12}}^{sg}$ *and* $w_{\mathbf{12}}^{apga}$ *be defined on* $\mathcal{H}_{sg}$ *and* $\mathcal{H}_{apga} =$ **Fix** $\text{IAT}(\alpha)(\mathcal{H}')$ *respectively. Then, (1)* $\forall t \in S_2^{sg}, u \in S_2^{apga} : w_{\mathbf{12}}^{sg}(t) = w_{\mathbf{12}}^{apga}(u)$, *and (2)* $|S_1^{sg}| \geq |S_1^{apga}| \geq |\hat{S}_1^{apga}|$.

*Example 7.* APGA $\mathcal{H}$ (right) is an SG-based abstraction [5] of PA $\mathcal{M}$ (left) in Fig. 9; whereas the left APGA in Fig. 10, say $\mathcal{H}''$, is a stable abstraction of $\mathcal{M}$ w.r.t. the objectives **1** = min, **2** = max and $T = \{t_3\}$. Note that both models have the same reachability probabilities to $t_3$ (i.e., 0.25) from the initial states. Moreover, $|S_1| = 6$ and $|S_1''| = 5$, and $|\Delta| = 12$ and $|\Delta''| = 11$.

### 5.3 Refining player-two states

We now discuss the outermost loop refining player-two states. This is, in principle, similar to *strategy-based refinement* in [5]. Let $\mathcal{H}$ be a *stable* abstraction of APGA $\mathcal{H}'$. If the reachability probabilities – w.r.t $w_{\mathbf{12}}$ and $w_{\mathbf{11}}$ – of $S_2$ states (that are of interest) are at most $\epsilon$-apart, we are done. Otherwise, we refine some of the player-two states.

**Definition 17 ($\epsilon$-boundedness).** *State* $s \in S$ *is* $\epsilon$-bounded *for* $\epsilon \in \mathbb{R}_{(0,1)}$ *iff* $|w_{\mathbf{12}}(s) - w_{\mathbf{11}}(s)| \leq \epsilon$. *Distribution* $\mu \in \text{Dist}(S)$ *is* $\epsilon$-bounded *iff* $|w_{\mathbf{12}}(\mu) - w_{\mathbf{11}}(\mu)| \leq \epsilon$. *APGA* $\mathcal{H}$ *is* $\epsilon$-bounded *iff all its states are bounded.*

**Lemma 1.** *In an unbounded APGA, the reachability probabilities of some player-two state — w.r.t. $w_{\mathbf{11}}$ and $w_{\mathbf{12}}$ — depend on two different successors.*

The lemma follows from the fact that if the reachability probabilities of each player-two state in an APGA depends on one of its successors, then the APGA represents the embedding of a PA that is 0-bounded — upper and lower bounds of reachability probabilites coincide for each player-two state.

The above lemma helps finding player-two states that can be refined. Let $\mathrm{ub}(\mathcal{H}) = \{t \in S_2 \mid \exists u, v \in \mathrm{succ}(t) : u \neq v \wedge w_{\mathbf{12}}(t) = w_{\mathbf{12}}(u) \wedge w_{\mathbf{11}}(t) = w_{\mathbf{11}}(v)\}$ be the set of player-two states in $\mathcal{H}$ whose reachability probability bounds depend on two different successors. A state in $\mathrm{ub}(\mathcal{H})$ can be refined as:

**Definition 18.** *State $t \in \mathrm{ub}(\mathcal{H})$ can be partitioned into $\mathrm{P}_1(t) = \{t' \in \gamma(t) \mid \exists \iota_{u'} \in \Delta'(t') : w_{\mathbf{12}}(t) = w_{\mathbf{12}}(\alpha(u'))\}$, and $\mathrm{P}_2(t) = \gamma(t) \backslash \mathrm{P}_1(t)$.*

Intuitively, concrete states (of $t$) whose player-one abstract states' reachability probabilities (w.r.t. $w_{\mathbf{12}}$) coincide with $w_{\mathbf{12}}(t)$ are separated from other concrete states. This is the basis for the outer-loop of our abstract-refine framework (Fig. 6).

**Definition 19 (Outer abstraction).** *The* outer abstraction transformer *function* $\mathrm{OAT} : \mathit{Abst}(\mathcal{H}') \to \mathit{Abst}(\mathcal{H}')$ *is defined for $\hat{\alpha} \in \mathit{Abst}(\mathcal{H}')$ with $\mathcal{H} = \mathbf{Fix}\ \mathrm{IAT}(\hat{\alpha})(\mathcal{H}')$ and $s' \in S'$ as:*

$$\mathrm{OAT}(\alpha = \mathbf{Fix}\ \mathrm{IAT}(\hat{\alpha}))(s') = \begin{cases} \alpha(s') & \text{if } \alpha(s') \in S_1 \text{ or } \alpha(s') \in S_2 \backslash \mathrm{ub}(\mathcal{H}) \\ \mathrm{P}_1(\alpha(s')) & \text{if } \alpha(s') \in \mathrm{ub}(\mathcal{H}) \text{ and } s' \in \mathrm{P}_1(\alpha(s')) \\ \mathrm{P}_2(\alpha(s')) & \text{if } \alpha(s') \in \mathrm{ub}(\mathcal{H}) \text{ and } s' \in \mathrm{P}_2(\alpha(s')) \end{cases}$$

Note that $\mathrm{OAT}(\alpha)$ maps $s'$ to the same partition block as $\alpha$ does if $\alpha(s')$ is a player-one state or a *bounded*-player-two state. Otherwise, it maps $s'$ either to $\mathrm{P}_1(s)$ or $\mathrm{P}_2(s)$.

*Example 8.* For $\epsilon = 0.4$, $\mathbf{1} = \min$, $\mathbf{2} = \max$ and $T = \{t_3\}$, the APGA $\mathcal{H}$ in Fig. 8 (left) is not an $\epsilon$-bounded abstraction of PA $\mathcal{M}$ in Fig. 7, as $|w_{\mathbf{12}}(t_1) - w_{\mathbf{11}}(t_1)| = |0.5 - 0| > \epsilon$ (0 is the reachability probability with $\mathbf{1} = \min$ and $\mathbf{2} = \min$). It is possible to refine $\mathcal{H}$ in order to have reachability probability bounds of $t_1$ at most $\epsilon$-apart. $\hat{\mathcal{H}}$ (Fig. 8 right) is an $\epsilon$-bounded abstraction of $\mathcal{M}$ obtained by partitioning the concrete states of $t_1$ in $\mathcal{H}$ into two blocks, i.e., $\mathrm{P}_1 = \{s_3\} = v_2'$ and $\mathrm{P}_2 = \{s_1, s_2\} = v_1'$. Note that $0 = |0 - 0| < \epsilon$ and $0 = |0.5 - 0.5| < \epsilon$ for $\hat{t}_1$ and $\hat{t}_2$ respectively.

The following theorem asserts that for $\tilde{\alpha} \in \mathit{Abst}(\mathcal{H}')$ with $\alpha = \mathbf{Fix}\ \mathrm{IAT}(\tilde{\alpha})$, the model induced by $\mathbf{Fix}\ \mathrm{IAT}(\mathrm{OAT}(\alpha))$ has at least as tight bounds on the reachability probabilities of player two states as the model induced by $\alpha$.

**Theorem 3.** *For $\tilde{\alpha} \in \mathit{Abst}(\mathcal{H}')$ with $\alpha = \mathbf{Fix}\ \mathrm{IAT}(\tilde{\alpha})$, $\mathbf{Fix}\ \mathrm{IAT}(\mathrm{OAT}(\alpha))(\mathcal{H}')$ has at least as tight bounds on the reachability probabilities of player-two states as $\alpha(\mathcal{H}')$.*

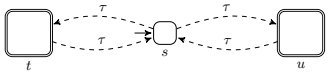


Fig. 11: $\alpha_\top(\mathcal{H}')$ for APGA $\mathcal{H}'$.

Like IAT, the fixpoint of the function OAT is guaranteed to exist for abstraction functions defined on the embedding of PAs with finite bisimulation quotient. Because in the worst case, the refinement of player-two states will eventually result in the embedding of PA that is 0-bounded, i.e., upper and lower bounds of reachability probabilites coincide for each player-two state. This therefore provides the basis to iteratively compute the partitioning of the state space of the model induced by $\alpha \in \mathit{Abst}(\mathcal{H}')$ such that the model induced by **Fix** $\mathrm{OAT}(\alpha)$ is an $\epsilon$-bounded abstraction.

**Theorem 4.** *For fixed $\epsilon \in \mathbb{R}_{(0,1)}$,* **Fix** $\mathrm{OAT}(\alpha)(\mathcal{H}')$ *is an $\epsilon$-bounded abstraction.*

In order to have an $\epsilon$-bounded abstraction, one can start with a coarsest abstraction $\mathcal{H}'$ given as $\alpha_\top(\mathcal{H}') = (\{s = \alpha(S'_1), t = \alpha_\top(T'), u = \alpha_\top(S'_2 \backslash T')\}, \{\{s\}, \{t, u\}\}, \{\tau\}, \emptyset, \{s \to_\mathrm{p} t, s \to_\mathrm{p} u, t \to_\mathrm{p} s, u \to_\mathrm{p} s\}, s)$ — recall that $T'$ is a set of goal states in $\mathcal{H}'$ — (see Fig. 11), and then refine it iteratively by Def. 19.

**Corollary 3.** **Fix** $\mathrm{OAT}(\alpha_\top)(\mathcal{H}')$ *is an $\epsilon$-bounded abstraction.*

## 6 Related Work

Abstraction of probabilistic automata (PAs) and the strongly related MDPs has received considerable attention. Starting from initial work by D'Argenio et al. [19] in 2001, techniques such as three-valued abstraction [20], counterexample-guided abstraction refinement (CEGAR) [21], and game-based abstraction [5] have been tailored to these probabilistic models. For a recent overview of abstraction techniques of probabilistic models, see [22].

*Abstraction.* Our abstraction is closely related to game-based abstraction. We separate the non-determinism in the concrete model and the non-determinism introduced by the abstraction. For each source of non-determinism, one player is used. Whereas [5] uses Shapley's stochastic games [11] as abstract models, we use (1) a variant in which both players are symmetric, and (2) extend this with modal transitions. Our abstract models are thus a *modal variant of probabilistic game automata* [8]. Whereas [5] uses the principle "states *must* have the same step-wise behaviour after abstraction to be merged together [5]"; in our setting states having the same step-wise behaviour after abstraction are *at least* merged together. SG-abstractions are thus a special case of our abstractions.

*Modal games and probabilistic models.* Modal extensions of two-player games have been studied in [23]. De Alfaro *et al.* show that modal game abstraction preserves alternating $\mu$-calculus, and provide (amongst others) a completeness results for a safety fragment of that logic. Our abstract stochastic games can be considered as lifting their model to the stochastic setting. Modal transitions for probabilistic models have been advocated in our earlier work [7, 4].

*Tighter abstractions.* All aforementioned abstraction techniques (including the one in this paper) for probabilistic models are state-based. That is, the relation between the concrete and abstract model is given by a simulation relation

that relates groups of concrete states to an abstract state. This has been casted in a general abstract interpretation setting in [24]. The abstraction in [5] is optimal in the sense of abstract interpretation [25] when relating states. Our earlier work [6] showed that using simulation and refinement relations that relate probability distributions rather than states has the potential to provide more precise abstractions. Relating distributions has also been applied [26] so as to obtain a distribution-based variant of Larsen and Skou's notion of probabilistic bisimulation [27]. Applying this principle to our abstraction-refinement framework has been briefly described in [18].

*Refinement.* Depending on whether the two players join forces so as to maximize or minimize the reachability probability or they act as opponents, analyzing the abstract game yields a lower or upper bound on the minimal or maximal reachability probability. If these bounds are sufficiently precise, the satisfaction or refutation of the property on the original PA can be concluded. Otherwise, the abstraction is refined. The resulting game then yields more precise results and, similarly to CEGAR, the procedure may be iterated until the obtained bounds are precise enough. In contrast to other refinement techniques, the crux of our technique is to separate the refinement of the various players, resulting in a *nested* abstraction-refinement loop. Player-two refinement is a mild variant of that in [5] in which states are always split in two parts [5]. Player-one refinement heavily relies on exploiting the modal transitions in the abstract model, a concept that is absent in [5].

## 7 Conclusion

This paper presented a *nested abstraction-refinement framework* for Segala's probabilistic automata (PAs). It is complete in the sense that termination is guaranteed for every PA with a finite bisimulation quotient. The key to our technique is to use a *modal variant* of Condon and Ladner's two-player probabilistic game automata. Abstraction using this model yields (tight) upper and lower bounds on extremal reachability probabilities. We believe that modal stochastic games are of interest as such and deserve further investigation. This paper focused on the theoretical underpinnings of our abstraction-refinement technique. An implementation and experimental comparison to game-based abstraction [5] is needed to check its practical feasibility and performance.

**Acknowledgements.** This work is strongly inspired by and heavily builds upon the work of Kim G. Larsen. The idea of using possible (may) and required (must) transitions goes back to his seminal work with Thomsen [28]. Simulation and refinement relations for probabilistic models originated in his work with Jonsson [16]. Kim developed one of the first, if not the very first, abstraction-refinement technique for MDPs [19]. His work on constraint Markov chains [29] provided the basis for our joint work on abstract PAs [4]. The uncertainty of

[5] This may converge slower than allowing for coarser splittings (as in [5]), but yields smaller state spaces.

the non-deterministic choices in APA is modeled by modal transitions while uncertainty of the stochastic behavior is expressed—as in constraint Markov chains—by (underspecified) stochastic constraints. Besides the influence of all these work, Kim has always been extremely inspiring. This started in 1996 at the conference FTRTFT in Uppsala, when he stimulated us to use Uppaal—at those days in its very early stage of development [30]—to take up the challenge of modeling and verifying Philips' bounded retransmission protocol [31]. This relationship has continued over the years and has led to several joint EU projects. It has been a great pleasure and enormous honor to work with Kim. This paper is a salute to his 60th birthday.

## References


1. Segala, R., Lynch, N.A.: Probabilistic simulations for probabilistic processes. Nordic J. of Computing **2**(2) (1995) 250–273
2. Norman, G.: Analysing randomized distributed algorithms. In: Validation of Stochastic Systems. Volume 2925 of LNCS, Springer (2004) 384–418
3. Huth, M.: On finite-state approximants for probabilistic computation tree logic. Theoretical Computer Science **346**(1) (2005) 113–134
4. Delahaye, B., Katoen, J.P., Larsen, K.G., Legay, A., Pedersen, M.L., Sher, F., Wasowski, A.: Abstract probabilistic automata. Inf. Comput. **232** (2013) 66–116
5. Kattenbelt, M., Kwiatkowska, M.Z., Norman, G., Parker, D.: A game-based abstraction-refinement framework for Markov decision processes. Formal Methods in System Design **36**(3) (2010) 246–280
6. Sher, F., Katoen, J.P.: Tight game abstractions of probabilistic automata. In: CONCUR. Volume 8704 of LNCS (2014) 576–591
7. Sher, F., Katoen, J.P.: Compositional abstraction techniques for probabilistic automata. In: IFIP TCS. Volume 7604 of LNCS (2012) 325–341
8. Condon, A., Ladner, R.E.: Probabilistic game automata. J. Comput. Syst. Sci. **36**(3) (1988) 452–489
9. Antonik, A., Huth, M., Larsen, K.G., Nyman, U., Wasowski, A.: 20 years of modal and mixed specifications. Bulletin of the EATCS **95** (2008) 94–129
10. Huth, M., Jagadeesan, R., Schmidt, D.: Modal transition systems: A foundation for three-valued program analysis. In: ESOP. Volume 2028 of LNCS (2001) 155–169
11. Shapley, L.S.: Stochastic games. Proc. of the National Academy of Sciences of the United States of America **39**(10) (1953) 1095–1100
12. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press (2008)
13. Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. Mathematics of Operations Research **16** (1991) 580–595
14. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pacific J. of Math. **5**(2) (1955) 285–309
15. Baier, C., Engelen, B., Majster-Cederbaum, M.E.: Deciding bisimilarity and similarity for probabilistic processes. J. Comput. Syst. Sci. **60**(1) (2000) 187–231
16. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: LICS, IEEE Computer Society (1991) 266–277
17. Larsen, K.G., Thomsen, B.: Compositional proofs by partial specification of processes. In: MFCS. Volume 324 of LNCS, Springer (1988) 414–423

18. Sher, F.: Abstraction and Refinement of Probabilistic Automata using Modal Stochastic Games. PhD thesis, RWTH Aachen University (2015) Aachener Informatik-Berichte AIB–2015–10.
19. D'Argenio, P.R., Jeannet, B., Jensen, H.E., Larsen, K.G.: Reachability analysis of probabilistic systems by successive refinements. In: PAPM-PROBMIV. Volume 2165 of LNCS, Springer (2001) 39–56
20. Katoen, J.P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for probabilistic systems. J. Log. Algebr. Program. **81**(4) (2012) 356–389
21. Hermanns, H., Wachter, B., Zhang, L.: Probabilistic CEGAR. In: Computer-Aided Verification. Volume 5123 of LNCS (2008) 162–175
22. Dehnert, C., Gebler, D., Volpato, M., Jansen, D.N.: On abstraction of probabilistic systems. In: ROCKS Autumn School. Volume 8453 of LNCS, Springer (2014) 87–116
23. de Alfaro, L., Godefroid, P., Jagadeesan, R.: Three-valued abstractions of games: Uncertainty, but with precision. In: LICS, IEEE Computer Society (2004) 170–179
24. Cousot, P., Monerau, M.: Probabilistic abstract interpretation. In: ESOP. Volume 7211 of LNCS, Springer (2012) 169–193
25. Wachter, B., Zhang, L.: Best probabilistic transformers. In: VMCAI. Volume 5944 of LNCS, Springer (2010) 362–379
26. Hermanns, H., Krcál, J., Kretínský, J.: Probabilistic bisimulation: Naturally on distributions. In: CONCUR. Volume 8704 of LNCS, Springer (2014) 249–265
27. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Inf. Comput. **94**(1) (1991) 1–28
28. Larsen, K.G., Thomsen, B.: A modal process logic. In: LICS, IEEE Computer Society (1988) 203–210
29. Caillaud, B., Delahaye, B., Larsen, K.G., Legay, A., Pedersen, M.L., Wasowski, A.: Constraint Markov chains. Theoretical Computer Science **412**(34) (2011) 4373–4404
30. Bengtsson, J., Larsen, K.G., Larsson, F., Pettersson, P., Yi, W.: UPPAAL in 1995. In: TACAS. Volume 1055 of LNCS, Springer (1996) 431–434
31. D'Argenio, P.R., Katoen, J.P., Ruys, T.C., Tretmans, J.: The bounded retransmission protocol must be on time! In: TACAS. Volume 1217 of LNCS, Springer (1997) 416–431