Tight Game Abstractions of Probabilistic Automata*

Falak Sher and Joost-Pieter Katoen

Software Modeling and Verification
RWTH Aachen University
D-52056 Aachen, Germany

Abstract. We present a new game-based abstraction technique for probabilistic automata (PA). The key idea is to use *distribution*-based abstraction – preserving novel distribution-based (alternating) simulation relations – rather than classical *state*-based abstraction. These abstractions yield (simple) probabilistic game automata (PGA), turn-based 2 player stochastic games in which moves of both players – as opposed to classical stochastic games – yield distributions over states. As distribution-based (alternating) simulation relations are pre-congruences for composite PGA, abstraction can be done compositionally. Our abstraction yields tighter upper and lower bounds on (extremal) reachability probabilities than state-based abstraction. This shows the potential superiority over state-based abstraction of PA and Markov decision processes.

1 Introduction

Probabilistic automata [1] (PA) extend labelled transition systems by allowing targets of transitions to be distributions over states rather than simply states. As transitions emanating from a state can be equally labelled, PA slightly generalize Markov decision processes (MDPs). This enables a natural way of putting PA in parallel. Due to the presence of non-determinism and discrete probabilistic branching, PA are convenient for modelling randomized distributed algorithms and security protocols. They are also popular semantic models for probabilistic process algebras and form the backbone of the PIOA language.

To combat the well-known state space explosion problem, abstractions of PA that go beyond bisimulation have received quite some attention. Whereas abstract PA [2–4] build upon concepts from modal transition systems and constraint functions, [5] uses three-valued abstraction yielding interval Markov chains, while [6] aggregates MDPs by separating the non-determinism in the MDPs from that introduced by abstraction. This naturally yields turn-based stochastic 2-player games [7, 8], where one player controls the non-determinism in the MDPs, whereas the other is in charge of the non-determinism from the abstraction. Game-based MDPs abstraction yields upper and lower bounds on reachability

* This research is supported by the EU FP7 SENSATION Project and the EU Marie-Curie Project MEALS.

probabilities, and significantly improves standard MDPs model checking as evidently shown by several case studies [6]. Besides, this game-based abstraction is optimal in the sense of abstract interpretation [9].

Although the aforementioned abstraction techniques are different in nature, they have in common that the abstraction is state-based. That is to say, abstract models simulate concrete models in a step-wise manner [10]. The key idea in this paper is to treat distributions rather than states as first-class citizens, and relax state-based simulation to *distribution*-based simulations. Our abstractions yield (simple) probabilistic game automata (PGA), turn-based 2-player stochastic games in which moves of both players – as opposed to classical stochastic games (SGs) [7, 8] – yield distributions over states. The new abstraction technique yields tighter upper and lower bounds on (extremal) reachability probabilities than state-based abstraction. This shows the potential superiority over *state*-based game-based MDP abstraction [6], and puts the optimality result of [9] in perspective.

Abstract models are probabilistic game automata (PGA), in fact simplified versions of the games in [11]. These games feature action-labelled transitions, in which every player non-deterministically makes a move and randomly picks the next state. We define two distribution-based pre-orders between abstract and concrete PGA: *simulation* and *alternating simulation* relations. Simulation relations are of interest when both players have identical objectives, whereas alternating simulation relations are useful for competitive objectives. Both relations are shown to be pre-congruences w.r.t. parallel composition of (a class of) PGA, enabling *compositional* abstraction of P(G)A. The pre-orders are the key to distribution-based abstraction, a technique distinguishing the non-deterministic behaviour of concrete distributions from that of the distributions induced by the abstraction. This enables merging concrete distributions having similar behaviour in the abstraction.

Put in a nutshell, the major contributions of this paper are: (1) a distribution-based abstraction framework of PA using a slight generalisation of stochastic games, (2) elementary results for distribution-based (alternating) simulation relations such as congruence properties and comparison to state-based simulation, and (3) distribution-based abstraction yields tighter bounds for extremal reachability probabilities than state-based abstraction.

Organization. Section 2 sets the ground for this paper and introduces SGs and PGA. Section 3 and 4 present (alternating) simulation relations for PGA. Section 5 treats two abstraction techniques. Section 6 presents game composition and the fact that abstraction is compositional. Section 7 presents results on bounding extremal reachability probabilities for PA. Section 8 discusses the special case of MDPs abstraction and compares to [6] while Section 9 concludes.

2 Preliminaries

Distributions. A *distribution* μ 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 given as $\text{Supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$; and its mass w.r.t. set $S' \subseteq S$ is given as $\mu(S') = \sum_{s \in S'} \mu(s)$. Let $|\mu| = \mu(S)$ denote the size of the distribution μ; μ is a *full distribution* iff $|\mu| = 1$, otherwise, it is a sub-distribution. Let $\text{Dist}(S)$ denote the set of distributions over S. Let $\iota_s \in \text{Dist}(S)$ denote the *Dirac* distribution for $s \in S$, i.e, $\iota_s(s) = 1$. A distribution μ'' can be split into sub-distributions μ and μ', say, represented as $\mu'' = \mu \oplus \mu'$, iff $\mu''(s) = \mu(s) + \mu'(s)$ for $s \in S$. Since $\oplus$ is associative and commutative, we use the notation $\bigoplus$ for finite sums. A distribution is sometimes represented as $\mu = [\![\mu(s)s \mid s \in \text{Supp}(\mu)]\!]$, where $[\![$ and $]\!]$ differentiate a set of probabilities from an ordinary set. For $0 \leq c \leq 1$, $c{\cdot}\mu$ denotes the distribution defined by: $(c{\cdot}\mu)(s) = c \cdot \mu(s)$. For a distribution μ, the conditional distribution w.r.t. a set $A \subseteq \text{Supp}(\mu)$ is given as: $\mu_{\downarrow_A}(s) = \frac{\mu(s)}{\mu(A)}$ for $s \in A$, and $\mu_{\downarrow_A}(s) = 0$ if $s \notin A$; if $A = \text{Supp}(\mu)$, we omit A and simply write $\mu_{\downarrow}$.

Probability measures and spaces. Let Ω be a non-empty set and $\mathcal{F} \subseteq 2^{\Omega}$. $\mathcal{F}$ is a σ-field on Ω iff: (1) $\emptyset \in \mathcal{F}$; (2) $A \in \mathcal{F} \Rightarrow \Omega \backslash A \in \mathcal{F}$; (3) $A_1, A_2, A_3, ... \in \mathcal{F} \Rightarrow \bigcup_{i \geq 1} A_i \in \mathcal{F}$. The elements of $\mathcal{F}$ are *measurable sets* and $(\Omega, \mathcal{F})$ is a *measurable space*. A function $\Pr : \mathcal{F} \to [0,1]$ is a *probability measure* on $(\Omega, \mathcal{F})$ iff $\Pr(\Omega) = 1$ and if $A_1, A_2, ...$ are disjoint elements in $\mathcal{F}$, then $\Pr(\bigcup_i A_i) = \sum_i \Pr(A_i)$. $(\Omega, \mathcal{F}, \Pr)$ is called a *measurable space*. For any $\mathcal{A} \subseteq \mathcal{F}$, there exists a unique smallest σ-field that contains $\mathcal{A}$ [12]; and given that $\mathcal{A}$ satisfies certain conditions [12], a *probability measure* defined on $\mathcal{A}$ can be uniquely extended to the σ-field containing $\mathcal{A}$.

Probabilistic Automata (PA). PA [1] is an extension of labelled transition systems (LTS) in which the target of any action-labelled transition is a distribution over states instead of a single state. Let UAct be a countable universe actions including the internal action τ. Formally,

Definition 1. *A* Probabilistic Automaton *is a tuple $\mathcal{M} = (S, A, \Delta, s_0)$ where S is a non-empty, countable set of states with initial state $s_0 \in S$; $A \subseteq \text{UAct}$; and $\Delta \subseteq S \times A \times \text{Dist}(S)$ is a set of transitions.*

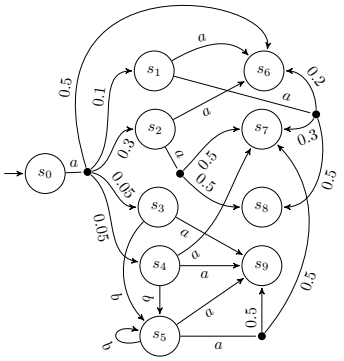

Fig. 1: A PA.

In the sequel, $\mathcal{M} = (S, A, \Delta, s_0)$ is assumed to be a finitely branching PA.

Stochastic Games (SGs). A 2-player SG [7, 8] is a game of chance played between two players, say, player 1 and player 2. 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 1 and 2 own S_1 and S_2 respectively. The game is

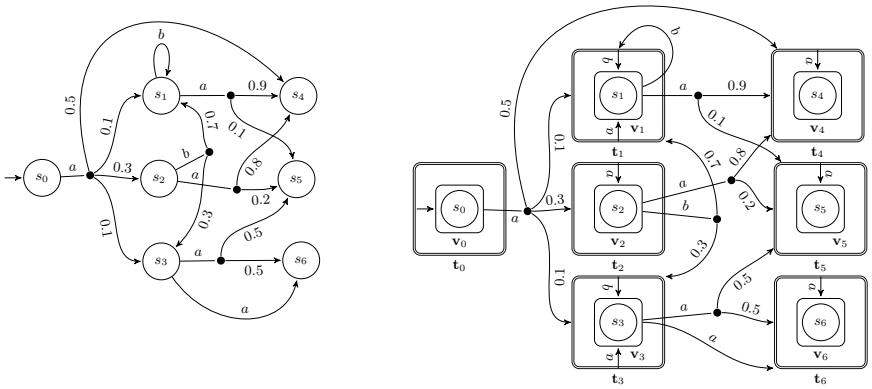

Fig. 2: A PA $\mathcal{M}$ (left) and its embedding $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M})$ (right)

started by player 1 and evolves in a turn-based fashion. Starting from the initial state in S_1, player 1 non-deterministically chooses an action-distribution pair. Based on the selected distribution a state in S_2, say s_2, is randomly selected and control is passed to player 2. Player 2 non-deterministically selects an enabled action in s_2, uniquely picks a successor of s_2 and passes control back to player 1. This goes on until some goal is achieved either by player 1 or player 2.

Definition 2. *A* Stochastic Game *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* $s_0 \in S_1$*;* $A \subseteq \mathrm{UAct}$*; and* $\Delta \subseteq (S_1 \times A \times \mathrm{Dist}(S_2)) \cup (S_2 \times A \times \mathrm{D}(S_1))$ *is a set of probabilistic transitions where* $\mathrm{D}(S_1) \subseteq \mathrm{Dist}(S_1)$ *is the set of Dirac distributions over* S_1*.*

We often denote $(s, a, \mu) \in \Delta$ by $s \xrightarrow{a} \mu$ and $\mathrm{Act}(s)$ as the set of enabled actions from state s, i.e., $\mathrm{Act}(s) = \{a \in A \mid \exists \mu \in \mathrm{Dist}(S) : s \xrightarrow{a} \mu\}$. We assume that a game is started by player 1 and $|\mathrm{Act}(s)| > 0$ for $s \in S$. Note that PA are SGs in which $\forall s \in S_2,\ a, b \in A : (s \xrightarrow{a} \mu \wedge s \xrightarrow{b} \nu)$ implies $\mu = \nu$ and $|\mathrm{Supp}(\mu)| = 1$.

Simple Probabilistic Game Automata (PGA). In SGs, player 1 moves yield distributions over states, while player 2 moves yield states. In PGA, player 2 moves also yield distributions over states.

Definition 3. *A* Simple Probabilistic Game Automaton *is a tuple* $\mathcal{G} = (S, \{S_1, S_2\}, A, \Delta, s_0)$ *where* S*,* S_1*,* S_2*,* A *and* s_0 *are as for* SGs*, and* $\Delta \subseteq S_{1+x} \times A \times \mathrm{Dist}(S_{2-x})$ *where* x *is a bit.*

Our PGA are simplified versions of the probabilistic game automata in [11]. SGs are a subclass of PGA in which $\mathrm{Dist}(S_1)$ is a set of Dirac distributions. In the sequel, let $\mathcal{G} = (S, \{S_1, S_2\}, A, \Delta, s_0)$ be a finitely branching PGA. For depicting PGA we represent states in S_1 and S_2 as rectangles and double rectangles respectively. In case of PA, states are circles. In the following, we show how a PA can be embedded into a PGA. For a state $s \in S$, let $\overline{s}$ be a copy of s.

Definition 4. *For* PA $\mathcal{M}$*, the bijective* embedding *function* $\alpha : S \to S'_2$ *induces the* PGA $\alpha(\mathcal{M}) = \mathcal{G}' = (S', \{S'_1, S'_2\}, A', \Delta', s'_0)$ *where* $A' = A$*,* $S'_1 = \{\overline{s'} \mid s' \in S'_2\}$ *—* S'_1 *is a copy of* S'_2 *—,* $s'_0 = \alpha(s_0)$ *and for every* $s' \in S'_2$*:*

1. $\overline{s'} \xrightarrow{a} \mu'$ *iff* $\alpha^{-1}(s') \xrightarrow{a} \mu$ *and* $\mu'(u') = \mu(\alpha^{-1}(u'))$ *for all* $u' \in S'_2$*,*
2. $s' \xrightarrow{a} \iota_{\overline{s'}}$ *iff* $\alpha^{-1}(s') \in \mathrm{Supp}(\mu)$ *for some* $u \in S$ *such that* $(u, a, \mu) \in \Delta$ *in* $\mathcal{M}$*.*

In the sequel, α_{PA} *denotes an* embedding *function for* PA*.*

Example 1. Let $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M})$ (see Fig. 2) with $S_2 = \{t_0, \ldots, t_6\}$ and $S_1 = \{v_0, \ldots, v_6\}$, $\alpha^{-1}_{\mathrm{PA}}(t_i) = s_i$, and $\overline{t_i} = v_i$, for $i = 0$ to 6. For convenience, the s_i states are depicted inside the corresponding states v_i and t_i. We have e.g., $v_2 \xrightarrow{b} \mu'$ with $\mu'(t_1) = \frac{7}{10}$ and $\mu'(t_3) = \frac{3}{10}$ and $t_1 \xrightarrow{b} v_1$ and $t_3 \xrightarrow{b} v_3$, as in PA $\mathcal{M}$ we have $s_2 \xrightarrow{b} \mu$ with $\mu(s_1) = \frac{7}{10}$ and $\mu(s_3) = \frac{3}{10}$.

Paths. If $|\mathrm{Act}(s)| > 1$ for state s, a non-deterministic choice among the enabled actions in s occurs. A path in a PGA represents a particular resolution of non-determinism by players 1 and 2 at each state, as well as a resolution of the probabilistic choices. Formally, a path from $s_{1_0} \in S_1$ is given as: $\pi = s_{1_0} \xrightarrow{a_{1_0}, \mu_{1_0}} s_{2_0} \xrightarrow{a_{2_0}, \mu_{2_0}} s_{1_1} \ldots$ where $s_{i_k} \in S_i$, $a_{i_k} \in \mathrm{Act}(s_{i_k})$, $\mu_{i_k} \in \mathrm{Dist}(S_i)$, $\mu_{1_k}(s_{2_k}) > 0$ and $\mu_{2_k}(s_{1_{k+1}}) > 0$ for all $i \in \{1, 2\}$ and $k \geq 0$; if $k < k'$ for some $k' \in \mathbb{N}^+$, then π is called *finite* path, otherwise *infinite* path. For a finite path π_{fin}, let $\mathrm{last}_i(\pi_{\mathrm{fin}})$ denote the last state in S_i for $i \in \{1, 2\}$ in π_{fin}. Let $\mathrm{Path}_{\mathrm{fin}}(\mathcal{G})$ and $\mathrm{Path}_{\mathrm{inf}}(\mathcal{G})$ denote the set of finite and infinite paths in a PGA $\mathcal{G}$ respectively, and $\mathrm{Paths}(\mathcal{G}) = \mathrm{Path}_{\mathrm{fin}}(\mathcal{G}) \cup \mathrm{Path}_{\mathrm{inf}}(\mathcal{G})$.

Schedulers. In order to analyse reachability properties on $\mathcal{G}$, we resolve non-determinism at all game states by means of a *scheduler* (also known as *policy, strategy or adversary*). Let κ_i be the scheduler for S_i, $i \in \{1, 2\}$. We consider *deterministic memoryless* (DM) schedulers as they suffice for reachability probabilities on PGA [11]. DM-schedulers select an action-distribution pair only on the basis of the current state. More specifically, for bit x, a *deterministic* scheduler $\kappa_{(1+x)}$ maps a finite path π_{fin} to a pair in $\mathrm{Act}(last_{(1+x)}(\pi_{\mathrm{fin}})) \times \mathrm{Dist}(S_{(2-x)})$; and a *memoryless* scheduler $\kappa_{(1+x)}$ assures that for finite paths π_{fin} and π'_{fin}, $last_{(1+x)}(\pi_{\mathrm{fin}}) = last_{(1+x)}(\pi'_{\mathrm{fin}})$ implies $\kappa_{(1+x)}(\pi_{\mathrm{fin}}) = \kappa_{(1+x)}(\pi'_{\mathrm{fin}})$.

A path π under a pair of DM-schedulers (κ_1, κ_2) is of the form $\pi = s_{1_0} \xrightarrow{a_{1_0}, \mu_{1_0}} s_{2_0} \xrightarrow{a_{2_0}, \mu_{2_0}} s_{1_1} \ldots$ where $\kappa_i(s_{i_k}) = (a_{i_k}, \mu_{i_k})$ for $i \in \{1, 2\}$ and $k \geq 0$. Let $\mathrm{Paths}^{\kappa_1}_{\kappa_2}(\mathcal{G})$ be the set of paths of PGA $\mathcal{G}$ under DM-schedulers (κ_1, κ_2). The DM-schedulers (κ_1, κ_2) on PGA $\mathcal{G}$ induce a countably infinite Markov chain. This allows us to construct a measurable space $(\mathrm{Paths}^{\kappa_1}_{\kappa_2}(\mathcal{G}), \mathcal{F}^{\kappa_1}_{\kappa_2}, \Pr^{\kappa_1}_{\kappa_2})$ over the (infinite) paths of $\mathcal{G}$ under (κ_1, κ_2). The problem of computing reachability probabilities on $\mathcal{G}$ reduces to a *stochastic shortest path problem* [13, 14] (for details, see Section 7). As reachability analysis is performed on closed versions of systems, we define a function that yields closed versions of PGA.

Definition 5. *For* PGA $\mathcal{G}$*, let* PGA $\tau(\mathcal{G}) = \mathcal{G}' = (S', \{S'_1, S'_2\}, A', \Delta', s'_0)$ *with* $S' = S$*,* $s'_0 = s_0$*,* $A' = \{\tau\}$ *and* $\Delta' = \{(s, \tau, \mu) \mid (s, a, \mu) \in \Delta\}$*.*

Combined hyper transitions. We adapt hyper and combined transitions – convex combinations of sets of transitions – for PA [1, 15] to PGA which are later on used in definitions.

Definition 6. *For* PGA $\mathcal{G}$ *with* $s \in S$ *and* $\mu \in \mathrm{Dist}(S)$*, we write:*

- $\mu \xrightarrow{a} \eta$ *is a* hyper transition *iff* $\eta = \bigoplus\{\mu(s) \cdot \rho \mid \exists s \in \mathrm{Supp}(\mu) : s \xrightarrow{a} \rho\}$*. Let* $\Delta(\mu, a) = \{\eta \mid \exists \eta \in \mathrm{Dist}(S) : \mu \xrightarrow{a} \eta\}$*.*
- $s \xrightarrow{a}_{\mathrm{c}} \eta$ *is a* combined transition *iff there is a finite indexed set* $\{(c_i, \eta_i)\}_{i \in I}$ *such that* $s \xrightarrow{a} \eta_i$ *and* $c_i \in \mathbb{R}_{\geq 0}$ *for all* $i \in I$*,* $\sum_{i \in I} c_i = 1$ *and* $\eta = \bigoplus_{i \in I} c_i \cdot \eta_i$*.*
- $\mu \xrightarrow{a}_{\mathrm{c}} \eta$ *is a* combined hyper transition *iff* $\eta = \bigoplus\{\mu(s) \cdot \rho \mid \exists s \in \mathrm{Supp}(\mu) : s \xrightarrow{a}_{\mathrm{c}} \rho\}$*.*

Simulation. The notion of simulation for probabilistic processes [10] is a pre-order on the state space requiring that whenever state u simulates state s, then u can mimic the stepwise behaviour of s but may have more behaviour. This notion can be lifted to distributions over states using weight functions [10]:

Definition 7. *Let* S *be a finite, non-empty set of states, and let* $\mu, \mu' \in \mathrm{Dist}(S)$*. For* $R \subseteq S \times S$*,* μ *is* simulated *by* μ' *w.r.t.* R*, denoted* $\mu R \mu'$*, iff there exists a weight 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 now recall Segala's probabilistic simulation relation [1] for PA.

Definition 8. $R \subseteq S \times S$ *is a* simulation *relation for* PA $\mathcal{M}$ *iff for every* sRs'*,* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a}_{\mathrm{c}} \mu'$ *with* $\mu R \mu'$*. Let* $\prec_{\mathrm{pa}}$ *be the largest simulation relation.*

We can lift $\prec_{\mathrm{pa}}$ to PA in the usual way: $\mathcal{M} \prec_{\mathrm{pa}} \mathcal{M}'$ for PA $\mathcal{M}$ and $\mathcal{M}'$, with initial states s_0 and s_0', iff $s_0 \prec_{\mathrm{pa}} s_0'$ in the disjoint union of $\mathcal{M}$ and $\mathcal{M}'$. In the sequel, we will adopt this convention to all simulation relations.

3 Simulation Relations on Stochastic Games

Simulation relations are typically defined over the states of models; however, in the probabilistic setting, coarser relations have been considered over the distributions over states [1, 16, 17]. We define simulation relations for PGA, that are state-based as well as distribution-based, and prove them to be pre-orders. Later on, these relations form the basis to compare a PGA with its abstraction.

Definition 9. $R \subseteq \bigcup_{j \in \{1,2\}} S_j \times S_j$ *is a* state-based simulation (SBS) *relation on* PGA $\mathcal{G}$ *iff for every* sRs'*,* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a}_{\mathrm{c}} \mu'$ *with* $\mu R \mu'$*. Let* $\prec_{\mathrm{sb}}$ *be the largest* SBS *relation.*

This asserts that for sRs', an a-transition from s implies a combined a-transition from s' such that the resulting distributions are related as by Def. 7. It is not difficult to show that $\prec_{\mathrm{sb}}$ is a preorder. Moreover, $\prec_{\mathrm{sb}} = \prec_{\mathrm{pa}}$ for PA.

Definition 10. $R \subseteq \bigcup_{j\in\{1,2\}} \mathrm{Dist}(S_j) \times \mathrm{Dist}(S_j)$ *is a* distribution-based simulation (DBS) *relation on* PGA $\mathcal{G}$ *iff for every* $\mu R \mu'$, *(1)* $\mu = \bigoplus_{s'\in \mathrm{Supp}(\mu')} \mu_{s'} : \mu'(s') = |\mu_{s'}|$ *and* $\mu_{s'}{\downarrow} R \iota_{s'}$, *(2)* $\mu \xrightarrow{a} \rho$ *implies* $\mu' \xrightarrow{a}_{\mathrm{c}} \rho'$ *such that* $|\rho| \leq |\rho'|$ *and* $\rho{\downarrow} R \rho'{\downarrow}$. *Let* $\prec_{\mathrm{db}}$ *be the largest* DBS *relation. We write* $s \prec_{\mathrm{db}} s'$ *iff* $\iota_s \prec_{\mathrm{db}} \iota_{s'}$.

By constraint (1), μ splits into sub-distributions as per the support of μ', i.e., for every $s' \in \mathrm{Supp}(\mu')$, there exists a sub-distribution μ'_s of μ such that the conditional distribution of μ'_s is related to $\iota_{s'}$. By constraint (2), an a-transition from μ to *some* ρ implies a combined a-transition from μ' to ρ' such that the mass of ρ' is at least that of ρ and their conditional distributions are related.

Example 2. In Fig. 3, $\mu = [\![0.3s_3, 0.3s_4, 0.4s_5]\!] \prec_{\mathrm{db}} \iota_{s_0}$ as $R = \{(\iota_{s_1}, \iota_{s_1}), (\iota_{s_2}, \iota_{s_2}), ([\![0.3s_3, 0.3s_4, 0.4s_5]\!], \iota_{s_0}), ([\![0.5s_1, 0.5s_2]\!], [\![0.5s_1, 0.5s_2]\!])\}$ is a DBS relation. Let us check the conditions of Def. 10 for μ and ι_{s_0}. The constraint (1) trivially holds for μ and ι_{s_0}. For the a-transition from μ to $\rho = [\![0.3s_1, 0.3s_2]\!]$, there is an a-transition from ι_{s_0} to $\rho' = [\![0.5s_1, 0.5s_2]\!]$ such that $|\rho| \leq |\rho'|$ and $\rho{\downarrow} R \rho'$. The same holds for the b-transitions from μ and ι_{s_0}, thus fulfilling constraint (2). Note that no SBS relation exists associating s_0 with any other state in Fig. 3.

4 Alternating Simulation Relations

To compare two-player stochastic games with competitive objectives (e.g., if player 1 maximises the probability to reach a certain goal state, her opponent (player 2) will try to minimize this quantity), we use *alternating* simulation relations. Our state-based alternating simulation relations are inspired by the notions of alternating simulation [18] and strong probabilistic game simulation [19].

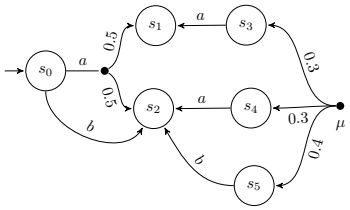

Fig. 3: $[\![0.5s_3, 0.5s_4]\!] \prec_{\mathrm{db}} \iota_{s_0}$ but $s_i \not\prec_{\mathrm{sb}} s_0$ for $i \in \{3, 4\}$.

Definition 11. $R \subseteq \bigcup_{j\in\{1,2\}} S_j \times S_j$ *is a* state-based alternating simulation (SBAS) *relation for* PGA $\mathcal{G}$ *iff for every* sRs' *the following holds: (1) if* $s, s' \in S_1$, *then* $s' \xrightarrow{a} \mu'$ *implies* $s \xrightarrow{a}_{\mathrm{c}} \mu$ *such that* $\mu R \mu'$, *(2) if* $s, s' \in S_2$, *then* $s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a}_{\mathrm{c}} \mu'$ *such that* $\mu R \mu'$. *Let* $\preccurlyeq_{\mathrm{sb}}$ *be the largest* SBAS *relation.*

Intuitively, in case of player 1 states, the behaviour of s' is simulated by that of s; whereas in case of player 2 states, it is the other way round. The first constraint asserts that if $s, s' \in S_1$, then an a-transition from s' implies a combined a-transition from s and the resulting distributions are related with each other by Def. 7. The second constraint asserts that if $s, s' \in S_2$, the similar conditions as in (1) hold for every transition from s. It is easy to show that $\preccurlyeq_{\mathrm{sb}}$ is a preorder.

Remark 1. The strong probabilistic game simulation relation in [19] [Def. 6.10] is obtained by merging Def. 9 and 11 and lifting them to player 2 states.

Definition 12. $R \subseteq \bigcup_{j\in\{1,2\}} \mathrm{Dist}(S_j) \times \mathrm{Dist}(S_j)$ *is a* distribution-based alternating simulation (DBAS) *relation for* PGA $\mathcal{G}$ *iff for every* $\mu R \mu'$*: (1)* $\mu = \bigoplus_{s'\in\mathrm{Supp}(\mu')} \mu_{s'} : |\mu_{s'}| = \mu'(s')$ *and* $\mu_{s'\downarrow} R \iota_{s'}$*, (2) if* $\mu, \mu' \in \mathrm{Dist}(S_1)$*,* $\mu' \xrightarrow{a} \rho'$ *implies* $\mu \xrightarrow{a}_{\mathrm{c}} \rho$ *such that* $|\rho| \geq |\rho'|$ *and* $\rho_\downarrow R \rho'_\downarrow$*, (3) if* $\mu, \mu' \in \mathrm{Dist}(S_2)$*,* $\mu \xrightarrow{a} \rho$ *implies* $\mu' \xrightarrow{a}_{\mathrm{c}} \rho'$ *such that* $|\rho| \leq |\rho'|$ *and* $\rho_\downarrow R \rho'_\downarrow$*. Let* $\preccurlyeq_{\mathrm{db}}$ *be the largest* DBAS *relation. We write* $s \preccurlyeq_{\mathrm{db}} s'$ *iff* $\iota_s \preccurlyeq_{\mathrm{db}} \iota_{s'}$*.*

The constraint (1) is the same as in Def. 10. By constraint (2), if $\mu, \mu' \in \mathrm{Dist}(S_1)$, then an a-transition from μ' to ρ' implies a combined a-transition from μ to ρ such that the mass of ρ is at least the mass of ρ' and the conditional distribution of ρ is in relation R with that of ρ'. And by constraint (3), if $\mu, \mu' \in \mathrm{Dist}(S_2)$, the similar conditions as in (2) hold for every transition from μ. Note that if the state space is not partitioned (as for PA), then simulation relations coincide with alternating simulation relations:

Proposition 1. $\prec_x = \preccurlyeq_x^{-1}$ *for* PA*, where* $x \in \{\mathrm{sb}, \mathrm{db}\}$*.*

Theorem 1. $\prec_{\mathrm{db}}$ *and* $\preccurlyeq_{\mathrm{db}}$ *are preorders.*

Although a state-based (alternating) simulation relation can be lifted from states to distributions over states (by Def. 7), an example can be constructed showing state-based (lifted to distributions over states) and distribution-based (alternating) simulation relations are not ordered in general but for closed PGA.

Proposition 2. $\prec_{\mathrm{sb}}/\preccurlyeq_{\mathrm{sb}}$ *and* $\prec_{\mathrm{db}}/\preccurlyeq_{\mathrm{db}}$ *are incomparable for* PGA*; and* $\prec_{\mathrm{sb}}/\preccurlyeq_{\mathrm{sb}} \subseteq \prec_{\mathrm{db}}/\preccurlyeq_{\mathrm{db}}$ *for closed* PGA*.*

At the end of this section, we highlight that if PGA are in a state-based or a distribution-based relation, their closed versions are also in that relation.

5 Game Abstraction

In this section, we show that PGA can act as appropriate abstract models for PA. We do so by considering abstractions of PGA that are embeddings of PA. Let $\mathcal{G}$ be a PGA with $S = \{S_1, S_2\}$. Intuitively, the state space S_2 of $\mathcal{G}$ is partitioned and each partition is represented by a single state in the abstract state space S'_2. This step induces a partition of S_1. We propose two different techniques for the partition of S_1: (a) S_1 states having similar behaviour under the player 2 partition S'_2 are grouped (*state-based abstraction*); (b) the (sub-)distributions (over S_1) that have similar behaviour are grouped (*distribution-based abstraction*). In the sequel, we show that the latter technique is more precise as well as concise than the former one.

Let (α, γ) be an abstraction-concretization pair such that $\alpha : S \to S'$ is a surjection and $\gamma : S' \to 2^S$ is 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 μ is given as $\alpha(\mu)(s') = \mu(\gamma(s'))$. The functions α and γ are lifted to sets of states or sets of distributions in a pointwise manner.

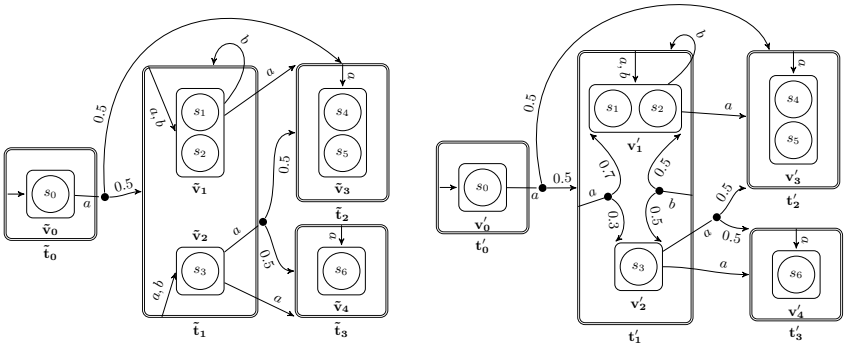

Fig. 4: For game $\mathcal{G}$ (Fig. 2 right), $\tilde{\mathcal{G}} = \alpha_{\text{sb}}(\mathcal{G})$ (left) and $\mathcal{G}' = \alpha_{\text{db}}(\mathcal{G})$ (right).

Definition 13. *For* PGA $\mathcal{G}$*, the* state-based abstraction *function* $\alpha : S \to S'$ *induces the* PGA $\alpha(\mathcal{G}) = \mathcal{G}' = (S', \{S'_1, S'_2\}, A', \Delta', s'_0)$ *where* $A' = A$*;* $S'_i = \alpha(S_i)$ *for* $i \in \{1, 2\}$*;* $\forall u', v' \in S'_1$*:* $\Delta'(u') = \Delta'(v')$ *implies* $u' = v'$*;* $s'_0 = \alpha(s_0)$ *and for every* $s' \in S'$*:*

1. *if* $s' \in S'_1$*, then: (a)* $s' \xrightarrow{a} \mu'$ *iff* $\forall s \in \gamma(s') : s \xrightarrow{a} \mu$ *such that* $\alpha(\mu) = \mu'$*, (b)* $\exists s \in \gamma(s') : s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a} \mu'$ *such that* $\alpha(\mu) = \mu'$*,*
2. *if* $s' \in S'_2$*, then: (a)* $s' \xrightarrow{a} \mu'$ *implies* $\exists s \in \gamma(s') : s \xrightarrow{a} \mu$ *such that* $\alpha(\mu) = \mu'$*, (b)* $\exists s \in \gamma(s') : s \xrightarrow{a} \mu$ *implies* $s' \xrightarrow{a}_{\text{c}} \mu'$ *such that* $\alpha(\mu) = \mu'$*.*

In the sequel, $(\alpha_{\text{sb}}, \gamma_{\text{sb}})$ *denotes a pair of* state-based abstraction-concretization *functions for* PGA*.*

By constraint (1) all player 1 states in the concrete model whose transitions become similar after abstraction — that can be found by considering their ordinary transitions instead of combined transitions — are aggregated; thus every state in S'_1 has a unique set of transitions enabled from it. Besides, (2) transitions of player 2 abstract states are derived from their concrete states, whose convex combination simulate the (abstract) transitions of concrete states.

Example 3. Let $\tilde{\mathcal{G}} = \alpha_{\text{sb}}(\mathcal{G})$ (Fig. 4 left) where $\mathcal{G}$ is the PGA of Fig. 2 (see page) with $\gamma_{\text{sb}}(\tilde{t}_0) = \{t_0\}$, $\gamma_{\text{sb}}(\tilde{t}_1) = \{t_1, t_2, t_3\}$, $\gamma_{\text{sb}}(\tilde{t}_2) = \{t_4, t_5\}$ and $\gamma_{\text{sb}}(\tilde{t}_3) = \{t_6\}$. Consider v_1, v_2 and v_4, v_5 in S_1; the transitions of v_1 and v_2 are the same after abstraction; therefore, they are merged into $\tilde{v}_1$. The same applies to v_4 and v_5. Now consider the transitions from $\tilde{t}_1$; for each concrete transition from t_1, t_2 and t_3, there is a corresponding abstract transition from $\tilde{t}_1$; thus, $\tilde{t}_1$ simulates (according to Def. 9) t_1, t_2 and t_3 (after abstraction).

Theorem 2. *For* PGA $\mathcal{G}$*,* $\mathcal{G} \prec_{\text{sb}} \alpha_{\text{sb}}(\mathcal{G})$ *and* $\mathcal{G} \preccurlyeq_{\text{sb}} \alpha_{\text{sb}}(\mathcal{G})$*.*

Definition 14. *For* PGA $\mathcal{G}$*, the* distribution-based abstraction *function* $\alpha : S \to S'$ *induces the* PGA $\alpha(\mathcal{G}) = \mathcal{G}' = (S', \{S'_1, S'_2\}, A', \Delta', s'_0)$ *where* $A' = A$*;* $S'_i = \alpha(S_i)$ *for* $i \in \{1, 2\}$*;* $\forall u', v' \in S'_1$*:* $\Delta'(u') = \Delta'(v')$ *implies* $u' = v'$*;* $s'_0 = \alpha(s_0)$ *and for all* $\mu' \in \text{Dist}(S')$*:*

1. $\forall \mu \in \gamma(\mu') : \mu = \bigoplus_{s' \in \mathrm{Supp}(\mu')} \mu_{s'} : \mu'(s') = |\mu_{s'}| \wedge \alpha(\mu_{s'})_{\downarrow} = \iota_{s'}$,
2. *if* $\mu' \in \mathrm{Dist}(S'_1)$, *then:*
   (a) $\mu' \xrightarrow{a} \eta'$ *iff* $\forall \mu \in \gamma(\mu') : \mu \xrightarrow{a} \eta$ *such that* $|\eta| = |\eta'|$ *and* $\alpha(\eta)_{\downarrow} = \eta'_{\downarrow}$,
   (b) $\exists \mu \in \gamma(\mu') : \mu \xrightarrow{a} \eta$ *implies* $\mu' \xrightarrow{a} \eta'$ *such that* $|\eta| = |\eta'|$ *and* $\alpha(\eta)_{\downarrow} = \eta'_{\downarrow}$,
3. *if* $\mu' \in \mathrm{Dist}(S'_2)$, *then:*
   (a) $\mu' \xrightarrow{a} \eta'$ *implies* $\exists \mu \in \gamma(\mu') : \mu \xrightarrow{a} \eta$ *such that* $|\eta| \leq |\eta'|$ *and* $\alpha(\eta)_{\downarrow} = \eta'_{\downarrow}$,
   (b) $\exists \mu \in \gamma(\mu') : \mu \xrightarrow{a} \eta$ *implies* $\mu' \xrightarrow{a}_{\mathrm{c}} \eta'$ *such that* $|\eta| \leq |\eta'|$ *and* $\alpha(\eta)_{\downarrow} = \eta'_{\downarrow}$.

In the sequel, $(\alpha_{\mathrm{db}}, \gamma_{\mathrm{db}})$ *denotes a pair of* distribution-based abstraction-concretization *functions for* PGA.

As in a state-based abstraction, all player 1 states in a distribution-based abstraction of a PGA have a unique set of transitions enabled from them. However, the distribution-based abstraction differs from the state-based one in several ways: (1) asserts the splitting of every concrete distribution μ of μ' into sub-distributions as per the support of μ', i.e., $\mu = \bigoplus_{s' \in \mathrm{Supp}(\mu')} \mu_{s'}$, and the conditional distribution of $\mu_{s'}$ is abstracted by $\iota_{s'}$. By (2a), when μ' is defined over S'_1, then μ' has an a-transition to *some* distribution η' if its every concrete distribution μ has an a-transition to some distribution η such that the masses of η and η' coincide and (the conditional distribution of) η is abstracted by (that of) η'; moreover, (2b) all transitions from μ are present (after abstraction) from μ'. In fact, all concrete distributions of μ' have similar behaviour after abstraction, that can be asserted by considering their ordinary transitions instead of combined transitions. By (3a), when μ' is defined over S'_2, then μ' has an a-transition to *some* distribution η' if a concrete distribution μ (of μ') has an a-transition to some distribution η such that the mass of η' is at least that of η and (the conditional distribution of) η is abstracted by (that of) η'. Moreover, (3b) the transitions of concrete distributions of μ' are simulated by the convex combination of transitions of μ'.

Example 4. Let $\mathcal{G}' = \alpha_{\mathrm{db}}(\mathcal{G})$ (Fig. 4 right) for $\mathcal{G}$ in Fig.2 with $\gamma_{\mathrm{db}}(t'_0) = \{t_0\}$, $\gamma_{\mathrm{db}}(t'_1) = \{t_1, t_2, t_3\}$, $\gamma_{\mathrm{db}}(t'_2) = \{t_4, t_5\}$ and $\gamma_{\mathrm{db}}(t'_3) = \{t_6\}$. As the abstract state space is the same as for the state-based abstraction in the previous example, the transitions of concrete states v_1 and v_2 are the same after abstraction; therefore, they are merged into v'_1. The same applies to v_4 and v_5. Now, consider the transition $v'_0 \xrightarrow{a} [\![0.5t'_1, 0.5t'_2]\!]$ and its corresponding concrete transition $v_0 \xrightarrow{a} [\![0.1t_1, 0.3t_2, 0.1t_3, 0.5t_4]\!]$; note that $[\![0.1t_1, 0.3t_2, 0.1t_3, 0.5t_4]\!]$ can be split into sub-distributions as per the support of $[\![0.5t'_1, 0.5t'_2]\!]$. Consider the abstract distribution $[\![0.5t'_1]\!]$ and its concrete distribution $[\![0.1t_1, 0.3t_2, 0.1t_3]\!]$; for $[\![0.1t_1, 0.3t_2, 0.1t_3]\!]_{\downarrow} \xrightarrow{a} [\![0.1v_1, 0.3v_2, 0.1v_3]\!]_{\downarrow}$, there is a $[\![0.5t'_1]\!]_{\downarrow} \xrightarrow{a}_{\mathrm{c}} [\![0.4v'_1, 0.1v'_2]\!]_{\downarrow}$ and $[\![0.1v_1, 0.3v_2, 0.1v_3]\!]_{\downarrow} \in \gamma_{\mathrm{db}}([\![0.4v'_1, 0.1v'_2]\!]_{\downarrow})$; and for $[\![0.1t_1, 0.3t_2, 0.1t_3]\!]_{\downarrow} \xrightarrow{b} [\![0.1v_1, 0.1v_3]\!]_{\downarrow}$, there is a $[\![0.5t'_1]\!]_{\downarrow} \xrightarrow{b} [\![0.25v'_1, 0.25v'_2]\!]_{\downarrow}$ and $[\![0.1v_1, 0.1v_3]\!]_{\downarrow} \in \gamma_{\mathrm{db}}([\![0.25v'_1, 0.25v'_2]\!]_{\downarrow})$. Now consider the b-transition from v'_1 to $\iota_{t'_1}$; we have two concrete b-transitions: from v_1 to ι_{t_1} and from v_2 to $[\![0.7t_1, 0.3t_3]\!]$. For $\iota_{t_1} \xrightarrow{b} \iota_{v_1}$, there is a $\iota_{t'_1} \xrightarrow{b} \iota_{v'_1}$; and for $[\![0.7t_1, 0.3t_3]\!] \xrightarrow{b} [\![0.7v_1, 0.3v_3]\!]$, there

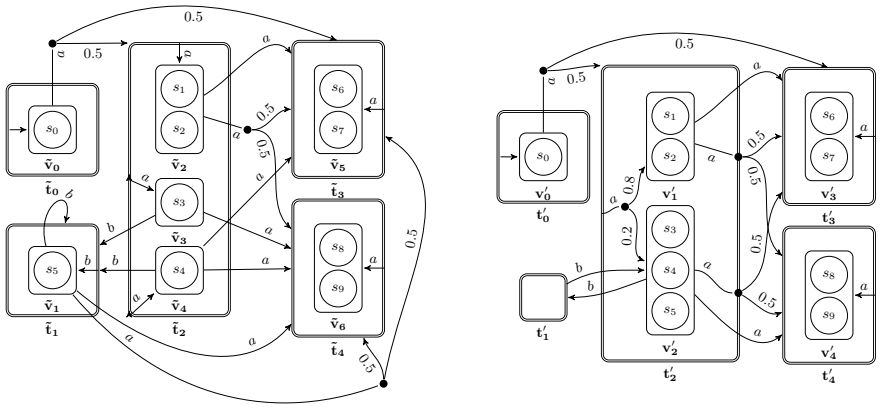

Fig. 5: For PA $\mathcal{M}$ (Fig. 1), $\tilde{\mathcal{G}} = \alpha_{\mathrm{sb}}(\alpha_{\mathrm{PA}}(\mathcal{M}))$ (left) and $\mathcal{G}' = \alpha_{\mathrm{db}}(\alpha_{\mathrm{PA}}(\mathcal{M}))$ (right). Considering each probabilistic transition as two transitions, $|\tilde{\Delta}| = 20$ and $|\tilde{S}_1| = 7$; whereas $|\Delta'| = 14$ and $|S'_1| = 5$

is a $\iota_{t'_1} \xrightarrow{b}_{\mathrm{c}} [\![0.7v'_1, 0.3v'_2]\!]$. Same is the case with a-transitions from ι_{t_1} and $[\![0.7t_1, 0.3t_3]\!]$.

In the previous example, only those states in S_1 whose transitions became the same after abstraction were aggregated. The next example illustrates that S_1 states having different transitions after abstraction can also be aggregated.

Example 5. For PA $\mathcal{M}$ (Fig. 1), let $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M})$ be its induced game. Let $\tilde{\mathcal{G}} = \alpha_{\mathrm{sb}}(\mathcal{G})$ (Fig. 5 left) be the state-based abstract model of $\mathcal{G}$ with $\gamma_{\mathrm{sb}}(\tilde{t}_0) = \{t_0\}$, $\gamma_{\mathrm{sb}}(\tilde{t}_1) = \{t_5\}$, $\gamma_{\mathrm{sb}}(\tilde{t}_2) = \{t_1, t_2, t_3, t_4\}$, $\gamma_{\mathrm{sb}}(\tilde{t}_3) = \{t_6, t_7\}$ and $\gamma_{\mathrm{sb}}(\tilde{t}_4) = \{t_8, t_9\}$. Let $\mathcal{G}' = \alpha_{\mathrm{db}}(\mathcal{G})$ (Fig.5 right) be the distribution-based abstract model of $\mathcal{G}$ with the same partition as above. Consider the distribution $[\![0.8v'_1, 0.2v'_2]\!]$ such that $[\![0.1v_1, 0.3v_2]\!]_{\downarrow}$ and $[\![0.05v_3, 0.05v_4]\!]_{\downarrow}$ are the corresponding distributions for v'_1 and v'_2 respectively. Note that $v'_2 \xrightarrow{a} \iota_{t'_4}, [\![0.5t'_3, 0.5t'_4]\!]$ iff $[\![0.05v_3, 0.05v_4]\!]_{\downarrow} \xrightarrow{a} \iota_{t_9}, [\![0.5t_7, 0.5t_9]\!]$. Similarly, $v'_2 \xrightarrow{b} \iota_{t'_1}$ iff $[\![0.05v_3, 0.05v_4]\!]_{\downarrow} \xrightarrow{b} \iota_{t_5}$. Moreover, the concrete distribution ι_{v_5} has the same behaviour as $[\![0.05v_3, 0.05v_4]\!]_{\downarrow}$, therefore, v_3, v_4 and v_5 are merged into v'_2. This example shows that distribution-based abstraction induces more concise models than state-based abstraction. Note that for PGA $\mathcal{G}$, $R = \{(s, \alpha_{\mathrm{db}}(s) \mid s \in S\}$ is not an SBS relation.

Theorem 3. *For* PGA $\mathcal{G}$, $\mathcal{G} \prec_{\mathrm{db}} \alpha_{\mathrm{db}}(\mathcal{G})$ *and* $\mathcal{G} \preccurlyeq_{\mathrm{db}} \alpha_{\mathrm{db}}(\mathcal{G})$.

Both Th. 2 and 3 are of importance when showing (in Section 7) that abstraction provides upper- and lower-bounds on extremal reachability probabilities in PGA (and thus PA).

Distribution- vs. state-based abstraction. Like for simulation relations, state-based abstraction is not a special case of distribution-based abstraction.

We observe that for every possible partition of state space, we can have a state-based abstract model of PGA, but not a distribution-based abstract model; however, for closed versions of PGA — PGA having $A = \{\tau\}$ —, we can have state-based as well as distribution-based abstract models.

Proposition 3. *$\alpha_{\mathrm{db}}(\tau(\mathcal{G}))$ is well-defined for* PGA *$\mathcal{G}$.*

By the above proposition, we mean that for every partition of state space of a closed PGA, we can construct a distribution-based abstract model, which is not the case with other PGA (not closed). However, for some PGA (not closed) we can have partitions of state space that can define distribution-based abstract models by aggregating states. Moreover, although we do not aggregate any states when the partition is S, α_{db} is defined for this partition. In the sequel, we assume that $\alpha_{\mathrm{db}}(\mathcal{G})$ is defined for PGA $\mathcal{G}$.

Now we prove that distribution-based abstraction is more precise than state-based abstraction. In fact, when two abstract models, obtained by a state-based and a distribution-based abstraction, have the same state space; then the latter one is at least as precise as the former one. Formally,

Theorem 4. *For* PGA *$\mathcal{G}$, $\alpha_{\mathrm{sb}}(S) = \alpha_{\mathrm{db}}(S)$ implies $\alpha_{\mathrm{db}}(\mathcal{G}) \prec \alpha_{\mathrm{sb}}(\mathcal{G})$ where $\prec \in \{\prec_{\mathrm{sb}}, \preccurlyeq_{\mathrm{sb}}\}$.*

6 Composition

We define a composition operator for a class of PGA that can act as abstract models of PA. The operator is defined in a TCSP-like manner, i.e., it is parametrized by a set of actions that need to be performed simultaneously by both games; other actions occur autonomously. For distributions μ and μ', let the point-wise product $\mu \| \mu' : S \times S \to [0,1]$ be given as: $\mu \| \mu'(s,s') = \mu(s) \cdot \mu'(s')$ for $s, s' \in S$.

Definition 15. *The* parallel composition *of* PGA *$\mathcal{G}$ and $\mathcal{G}'$ w.r.t. synchronization set $\bar{A} \subseteq (A \cap A')\backslash\{\tau\}$ is given as: $\mathcal{G} \parallel_{\bar{A}} \mathcal{G}' = (S \times S', \{S_1 \times S_1', S \times S'\backslash S_1 \times S_1'\}, A \cup A', \tilde{\Delta}, (s_0, s_0'))$, where for all $a \in A \cup A'$ and $(s,s') \in S \times S'$, $(s,s') \xrightarrow{a}_{\mathrm{c}} \mu \| \mu'$ iff one of the following holds:*

1. *if $(s,s') \in S_1 \times S_1'$, then (i) $a \in \bar{A}$, $s \xrightarrow{a}_{\mathrm{c}} \mu$ and $s' \xrightarrow{a}_{\mathrm{c}} \mu'$, or (ii) $a \in A$, $s \xrightarrow{a}_{\mathrm{c}} \mu$ and $\iota_{s'} = \mu'$, or (iii) $a \in A'$, $s' \xrightarrow{a}_{\mathrm{c}} \mu'$ and $\iota_s = \mu$,*
2. *if $(s,s') \in S_2 \times S_2'$, then $a \in \bar{A}$, $s \xrightarrow{a}_{\mathrm{c}} \mu$ and $s' \xrightarrow{a}_{\mathrm{c}} \mu'$,*
3. *else, (i) $s \in S_2$, $s \xrightarrow{a}_{\mathrm{c}} \mu$ and $\iota_{s'} = \mu'$, or (ii) $s' \in S_2'$, $s' \xrightarrow{a}_{\mathrm{c}} \mu'$ and $\iota_s = \mu$.*

Note that the state space of our composite game is disjointly dividable based on the actions which are enabled. Although, we allow composition of $S_1(S_2)$ states with that of $S_2'(S_1')$ states, but only player 2 can make a move in such a state. (1) asserts that states in $S_1 \times S_1'$ can either synchronize with each other or act independently. Note that a state in $S_2 \times S_2'$ is only reached by a synchronizing action performed by players of type 1 in some $S_1 \times S_1'$ state; and (2) asserts that the next state is reached only by some synchronizing action. (3) tells that for a

state in $S_{(1+x)} \times S'_{(2-x)}$, where x is a bit, no synchronization occurs and only player 2 can make a move independently. Note that such a state can only be reached by a non-synchronizing action.

Theorem 5. *For any set $\bar{A}$ and $x \in \{\mathrm{sb}, \mathrm{db}\}$, $\prec_x$ and $\preccurlyeq_x$ are pre-congruences w.r.t. $||_{\bar{A}}$.*

Like for APA [4], our state-based and distribution-based abstractions for PGA are compositional. Intuitively, the composite PGA may be exponentially larger in size as compared to the composing ones. This problem could be avoided by applying abstraction prior to composition as illustrated by the following theorem.

Theorem 6. *For* PGA $\mathcal{G}$ *and* $\mathcal{G}'$*, synchronization set $\bar{A}$ and abstraction functions α_x, α'_x; $\alpha_x(\mathcal{G}) \,||_{\bar{A}}\, \alpha'_x(\mathcal{G}') = (\alpha_x \times \alpha'_x)(\mathcal{G}||_{\bar{A}}\mathcal{G}')$ up to isomorphism, where $x \in \{\mathrm{sb}, \mathrm{db}\}$ and $\alpha_x \times \alpha'_x$ is defined as $(\alpha_x \times \alpha'_x)(s, s') = (\alpha_x(s), \alpha'_x(s'))$.*

7 Preservation of Reachability Probabilities

This section presents how optimal (i.e., maximal and minimal) reachability probabilities are preserved under abstraction. We first define some notations and definitions. Let $\Pr^{\kappa_1}_{\kappa_2}(T)$ be the probability of the set of paths from the initial state s_0 that reach some state in $T \subseteq S$ under schedulers (κ_1, κ_2) for PGA $\mathcal{G}$.

Definition 16. [11] *For* PGA $\mathcal{G}$*, the* optimal probabilities *of reaching $T \subseteq S$ for players 1 and 2 are defined as:* $\sup_{\kappa_1}\inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$ *and* $\inf_{\kappa_1}\sup_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$.

Intuitively, the reachability probability to a set T of target states is optimal for player 1 under scheduler κ iff for every scheduler κ_2 of player 2, $\inf_{\kappa_2} \Pr^{\kappa}_{\kappa_2}(T) = \sup_{\kappa_1}\inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$. Similarly, we can define optimal reachability probability for player 2. For PGA $\mathcal{G}$ and $T \subseteq S$, we write:

- $\max^{\blacktriangledown}(T) = \sup_{\kappa_1}\inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$ and $\max^{\blacktriangle}(T) = \sup_{\kappa_1}\sup_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$
- $\min^{\blacktriangledown}(T) = \inf_{\kappa_1}\inf_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$ and $\min^{\blacktriangle}(T) = \inf_{\kappa_1}\sup_{\kappa_2} \Pr^{\kappa_1}_{\kappa_2}(T)$.

Note that the values $\max^{\blacktriangledown}(T)$ and $\min^{\blacktriangle}(T)$ are the optimal reachability probabilities for players 1 and 2 respectively, which can be achieved by DM-schedulers [11]. The values $\max^{\blacktriangle}(T)$ and $\min^{\blacktriangledown}(T)$ – for which both players collaborate with each other – can be obtained similarly. For games with finite state spaces these values can be computed through value iteration [13, 14] or by linear programming.

Let $w : S \to [0, 1]$ be a probability valuation function mapping a state s to the probability of reaching target states $T \subseteq S$ from s. The probability valuation functions $W = \{w \mid w : S \to [0, 1]\}$ form a complete lattice $(W, \leq, \bot, \top)$ with order $\leq$, bottom element $\bot \in W$ and top element $\top \in W$. We write $w \leq w'$ iff $w(s) \leq w'(s)$ for $s \in S$. $\bot(s) = 0$ and $\top(s) = 1$ for $s \in S$. Moreover, w can be lifted from states to distributions over states as $w(\mu) = \sum_{s \in S} \mu(s) \cdot w(s)$ for $\mu \in \mathrm{Dist}(S)$.

Definition 17. *Let* PGA $\tau(\mathcal{G})$ *and* $T \subseteq S$. *For reachability goals* $\mathbf{1}, \mathbf{2} \in \{\min, \max\}$ *for players 1, 2 respectively, the* probability valuation transformer $\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}} : W \to W$ *is defined for* $w \in W$, $s \in S$ *and* $n \in \mathbb{N}$ *as:*

$$(\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}})^n(w)(s) = \begin{cases} 1 & s \in T,\ n \geq 0 \\ 0 & n = 0,\ s \notin T \\ \mathbf{1}\{w(\mu) \mid s \xrightarrow{\tau} \mu\} & s \in S_1,\ n > 0 \\ \mathbf{2}\{w(\mu) \mid s \xrightarrow{\tau} \mu\} & s \in S_2,\ n > 0 \end{cases}$$

For $n > 0$, when $s \in S_1$, then for the next iteration the reachability probability from s is the optimal value of the set $\{w(\mu) \mid s \xrightarrow{\tau} \mu\}$ w.r.t. objective $\mathbf{1}$; whereas when $s \in S_2$, it is w.r.t. objective $\mathbf{2}$. Note that $\mathrm{Prt}^{\mathbf{1}}_{\mathbf{2}}$ is a monotonic function over W and, by Tarski's theorem [20], has a least and a greatest fixpoint. This definition provides the basis to compute reachability probabilities. A similar function has been defined in [11].

The next theorem shows that simulation/alternating simulation relations between PGA provide bounds on their reachability probabilities when players collaborate/compete with each other. In fact, simulation relations between PGA bound $\max^{\blacktriangle}$ and $\min^{\blacktriangledown}$ values, and alternating simulation relations $\max^{\blacktriangledown}$ and $\min^{\blacktriangle}$ values.

Theorem 7. *For* $x \in \{\mathrm{sb}, \mathrm{db}\}$, *let* PGA $\mathcal{G}$ *and* $\mathcal{G}'$ *with* $\mathcal{G} \prec_x \mathcal{G}'$ *and* $\mathcal{G} \preccurlyeq_x \mathcal{G}'$. *Let* $T \subseteq S$ *such that* $T' = \{s' \in S' \mid \exists s \in T : s \prec_x s'\}$ *and* $T'' = \{s' \in S' \mid \exists s \in T : s \preccurlyeq_x s'\}$, *then: (1)* $\min^{\blacktriangledown}(T') \leq \min^{\blacktriangledown}(T)$ *and* $\max^{\blacktriangle}(T) \leq \max^{\blacktriangle}(T')$, *(2)* $\min^{\blacktriangle}(T) \leq \min^{\blacktriangle}(T'')$ *and* $\max^{\blacktriangledown}(T'') \leq \max^{\blacktriangledown}(T)$

As abstractions of PGA preserve simulation and alternating simulation relations, their optimal probabilities are bounded by their abstract models. This is laid down in the following corollary, a direct consequence of Th. 2, 3 and 7:

Corollary 1. *Let* $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M})$ *for* PA $\mathcal{M}$, *and* $x \in \{\mathrm{sb}, \mathrm{db}\}$ *with* $\mathcal{G}' = \alpha_x(\mathcal{G})$. *Let* $T \subseteq S_2$ *such that* $T' = \alpha_x(T)$. *Then* $\min^{\blacktriangledown}(T') \leq \min(T) \leq \min^{\blacktriangle}(T')$ *and* $\max^{\blacktriangledown}(T') \leq \max(T) \leq \max^{\blacktriangle}(T')$.

Note that for every $s \in T$, we have $s \prec_x \alpha_x(s)$ and $s \preccurlyeq_x \alpha_x(s)$ for $x \in \{\mathrm{sb}, \mathrm{db}\}$. Moreover, the target states are only player 2 states as they represent the partitions of the concrete states of PA. Next, as one of the main results of this work, we show that distribution-based abstraction of PA is more precise than state-based abstraction. This result is a direct consequence Th. 4 and 7.

Corollary 2. *Let* $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M})$ *for* PA $\mathcal{M}$, $\mathcal{G}_{\mathrm{sb}} = \alpha_{\mathrm{sb}}(\mathcal{G})$ *and* $\mathcal{G}_{\mathrm{db}} = \alpha_{\mathrm{db}}(\mathcal{G})$ *with* $\alpha_{\mathrm{sb}}(S) = \alpha_{\mathrm{db}}(S)$. *Let* $T \subseteq S_2$ *such that* $T_{\mathrm{sb}} = \alpha_{\mathrm{sb}}(T)$ *and* $T_{\mathrm{db}} = \alpha_{\mathrm{db}}(T)$. *Then* $\min(T) \leq \min^{\blacktriangle}(T_{\mathrm{db}}) \leq \min^{\blacktriangle}(T_{\mathrm{sb}})$ *and* $\max^{\blacktriangledown}(T_{\mathrm{sb}}) \leq \max^{\blacktriangledown}(T_{\mathrm{db}}) \leq \max(T)$.

Example 6. The minimum probability in PA $\mathcal{M}$ (Fig. 2) to reach state s_6 is 0.05. By Corollary 1, this probability lies in $[0, 0.25]$ for $\alpha_{\mathrm{sb}}(\mathcal{G}) = \tilde{\mathcal{G}}$ (Fig. 4 left). Instead, $\alpha_{\mathrm{db}}(\mathcal{G}) = \mathcal{G}'$ (Fig. 4 right) yields $[0, 0.125]$.

8 Distribution-based Game Abstraction of MDP

In [6], abstract models of Markov decision processes (MDP) are given as stochastic games (SG). These abstractions coincide with our state-based abstractions. The abstract models of MDP induced by our *distribution-based abstraction* are PGA that generalize SG. By Th. 4, our distribution-based abstraction induces more precise abstract models than state-based abstraction. This shows the superiority of our distribution-based abstraction technique over [6]. The following corollary follows from Def. 5 and Th. 4. It asserts that our distribution-based abstraction induces more precise abstractions than [6].

Corollary 3. *For* PA $\mathcal{M}'$*, let* $\mathcal{G} = \alpha_{\mathrm{PA}}(\mathcal{M}')$*. If* $\alpha_{\mathrm{sb}}(S) = \alpha_{\mathrm{db}}(S)$*, then:* $\alpha_{\mathrm{db}}(\tau(\mathcal{G})) \prec_{\mathrm{sb}} \alpha_{\mathrm{sb}}(\tau(\mathcal{G}))$ *and* $\alpha_{\mathrm{db}}(\tau(\mathcal{G})) \preccurlyeq_{\mathrm{sb}} \alpha_{\mathrm{sb}}(\tau(\mathcal{G}))$.

One may argue that although PGA-based abstract models of MDP are at least as precise as SG-based ones this comes at the expense of larger games, — e.g. more space is required to store the target distributions of player 2 transitions. The following example shows that for abstracting $\mathcal{G}$ — an embedding on MDP — with $\alpha_{\mathrm{db}}(S_2) = \alpha_{\mathrm{sb}}(S_2)$, $\alpha_{\mathrm{db}}(\mathcal{G})$ is at least as precise as $\alpha_{\mathrm{sb}}(\mathcal{G})$ and $|\alpha_{\mathrm{db}}(S)| \leq |\alpha_{\mathrm{sb}}(S)|$. (Recall that the same partition of player 2 states does not imply the same partition for player 1 states, as shown in Example 5).

Example 7. The maximum probability in PA $\mathcal{M}$ (Fig. 1) to reach states $\{s_8, s_9\}$ equals 0.3. By Corollary 1, this probability lies in $[0.25, 0.5]$ for the state-based abstraction $\tilde{\mathcal{G}}$ (Fig. 5 left). Instead, distribution-based abstraction $\mathcal{G}'$ (Fig. 5 right) yields $[0.3, 0.3]$. Moreover, ignoring player 2 transitions — such that the successor states from player 2 states are decided non-deterministically as in [6] — yields $[0.25, 0.5]$ in $\tilde{\mathcal{G}}$ and $\mathcal{G}'$. However, in terms of number of transitions and states, the size of $\mathcal{G}'$ is smaller than $\tilde{\mathcal{G}}$ (see Fig. 5).

As a side result of our achievement, we put the result of [9][Th. 2] in perspective: game-based abstraction is the optimal *state-based* abstraction, but not the optimal abstraction preserving reachability probabilities.

9 Conclusion

We gave two abstraction techniques — state-based and distribution-based — for PA, and presented PGA as abstract models for PA. We defined a composition operator for a class of PGA that act as abstract models for PA; and gave two notions of simulation and alternating simulation relations for PGA that are precongruences w.r.t. composition. Our distribution-based abstraction is more precise as well as concise than the one in [19]. Future work includes the application of this work to practical case studies, and the extension of abstraction-refinement framework, in [19], for PA.

10 Acknowledgement

We thank the reviewers for the constructive feedback, and helping us improve the quality of the paper.

References

1. Segala, R.: Modeling and Verification of Randomized Distributed Real-Time Systems. PhD thesis, Massachusetts Institute of Technology (1995)
2. Delahaye, B., Katoen, J.P., Larsen, K., Legay, A., Pedersen, M., Sher, F., Wasowski, A.: Abstract probabilistic automata. Information and Computation **232** (2013) 66–116
3. Delahaye, B., Katoen, J.P., Larsen, K.G., Legay, A., Pedersen, M.L., Sher, F., Wasowski, A.: Abstract probabilistic automata. In: VMCAI. Volume 6538 of LNCS, Springer (2011) 324–339
4. Sher, F., Katoen, J.P.: Compositional abstraction techniques for probabilistic automata. In: TCS. Volume 7604 of LNCS, Springer (2012) 325–341
5. Katoen, J.P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for probabilistic systems. J. Log. Algebr. Program. **81**(4) (2012) 356–389
6. 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
7. Shapley, L.S.: Stochastic games. Proceedings of the National Academy of Sciences of the United States of America **39**(10) (1953) 1095–1100
8. Condon, A.: The complexity of stochastic games. Information and Computation **96** (1992) 203–224
9. Wachter, B., Zhang, L.: Best probabilistic transformers. In: VMCAI. Volume 5944 of LNCS, Springer Verlag (2010) 362–379
10. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: LICS, IEEE CS Press (1991) 266–277
11. Condon, A., Ladner, R.E.: Probabilistic game automata. Journal of Computer and System Sciences **36**(3) (1988) 452–489
12. Ash, R.B., Doléans-Dade, C.A.: Probability & Measure Theory, 2nd Edition. Academic Press (2000)
13. Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. Mathematics of Operations Research **16** (1991) 580–595
14. Alfaro, L.d.: Computing minimum and maximum reachability times in probabilistic systems. In: Concurrency Theory. Volume 1664 of LNCS, Springer (1999) 66–81
15. Lynch, N.A., Segala, R., Vaandrager, F.W.: Observing branching structure through probabilistic contexts. SIAM J. Comput. **37**(4) (2007) 977–1013
16. Eisentraut, C., Hermanns, H., Zhang, L.: On probabilistic automata in continuous time. In: LICS, IEEE CS Press (2010) 342–351
17. Doyen, L., Henzinger, T.A., Raskin, J.F.: Equivalence of labeled Markov chains. Int. J. Found. Comput. Sci. **19**(3) (2008) 549–563
18. Alur, R., Henzinger, T.A., Kupferman, O., Vardi, M.Y.: Alternating refinement relations. In: Concurrency Theory. Volume 1466 of LNCS, Springer (1998) 163–178
19. Kattenbelt, M.: Automated Quantitative Software Verification. PhD thesis, University of Oxford (2010)
20. Tarski, A., et al.: A lattice-theoretical fixpoint theorem and its applications. Pacific Journal of Mathematics **5**(2) (1955) 285–309