HAL
archives-ouvertes.fr

Compositional Abstraction Techniques for Probabilistic Automata

Falak Sher, Joost-Pieter Katoen

▶ **To cite this version:**

Falak Sher, Joost-Pieter Katoen. Compositional Abstraction Techniques for Probabilistic Automata. Jos C. M. Baeten; Tom Ball; Frank S. Boer. 7th International Conference on Theoretical Computer Science (TCS), Sep 2012, Amsterdam, Netherlands. Springer, Lecture Notes in Computer Science, LNCS-7604, pp.325-341, 2012, Theoretical Computer Science. <10.1007/978-3-642-33475-7_23>. <hal-01556222>

HAL Id: hal-01556222
https://hal.inria.fr/hal-01556222

Submitted on 4 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

Compositional Abstraction Techniques for Probabilistic Automata*

Falak Sher and Joost-Pieter Katoen

Software Modeling and Verification Group, RWTH Aachen University, Germany

Abstract. We present aggressive abstraction techniques for probabilistic automata (PA), a state-based model involving discrete probabilistic and nondeterministic branching. Our abstractions yield *abstract* PA in which transitions are typed "possible" or "required"—as in modal transition systems—and have constraint functions as target. The key idea is to focus on identifying common combined-transitions from concrete states and putting them as required ones in the abstract state. We prove the correctness of our abstraction techniques, study their relationship, and show that they are *compositional* w.r.t. parallel composition. We also show the preservation of probabilistic and expected reachability properties for PA.

1 Introduction

Segala's probabilistic automata [16] (PA) are important state-based models for modeling complex systems that involve discrete probabilistic and nondeterministic branching. PA generalize labelled transition systems (LTSs) [14] in that the target of a transition is a probability distribution over the states rather than just a single state. Nondeterminism occurs if several (possibly equally) labelled transitions emanate from a given state. PA are *compositional*—a model of a complex system can be obtained by modeling its components and putting them in parallel, e.g., by synchronizing actions in a CSP-based manner. Thus, PA are an adequate modeling formalism for asynchronously concurrent systems with discrete probabilistic choice such as randomized distributed algorithms. They have been used as semantic model for probabilistic process algebras and the PIOA language.

This paper presents aggressive abstraction techniques for PA that aim at applying abstraction in a component-based,i.e., compositional manner. Rather than focusing on obtaining equivalent behaviour (as for bisimulation), we focus on refinement, a pre-order relation between abstract and concrete models, that assures every *required* transition of an abstract model is mimicked by a combined-transition in the concrete model; and any transition of a concrete model needs to be matched by an *optional* combined-transition in the abstraction. The kernel

* This research is supported by the EU FP7 MoVeS Project (Modeling, Verification and Control of Complex Systems) and the German-Dutch bilateral project ROCKS.

of our refinement yields a notion of strong bisimulation for PA. It is shown to be a pre-congruence w.r.t. parallel composition of abstract PA. This result is a cornerstone for the remaining results in this paper as it facilitates *compositional abstraction*.

As in [8, 7], our abstraction techniques yield *abstract* PA in which action-labelled transitions are typed either "possible" or"required" and have constraint functions as targets representing sets of distributions as in constraint Markov chains [5]. As in modal transition systems [9], this distinction is exploited to consider abstract PA as specifications representing a set of concrete PA, its implementations. The key idea of our abstraction techniques is to find out common combined-transitions from concrete states and put them as required transitions in the abstract state. We define two different notions of abstraction and prove their correctness, i.e., show that the concrete models are indeed refinements of the abstract models. Using the fact that our refinement is a pre-congruence for parallel composition, this enables the compositional abstraction of PA.

For the formal analysis and verification of probabilistic systems two quantitative properties are of interest: *probabilistic reachability* (the probability of reaching a set of goal states) and *expected reachability* (the expected number of transitions before reaching a set of goal states). Due to the presence of nondeterminism in PA, it is not always possible to have one unique value for these measures; instead, the *maximum* and the *minimum* values are obtained. As every abstraction technique induces additional nondeterminism, the analysis of the abstract PA gives *lower* and *upper* bounds on the maximum/minimum values for the concrete model. In our abstraction techniques, the distinction of required and possible transitions help achieving this goal of finding bounds on the maximum/minimum values as in [12]. It also helps finding how precise the abstraction is; the better the abstraction is, the tighter the bounds will be. Had there been just one type of transitions in abstract PA, there would have been only single maximum/minimum value for each property. The contributions of the paper are summarized as follows:

- a new notion of a transition relation, called *multi transition*, that is used in defining a refinement relation among PA,
- a new notion of a refinement relation that is a pre-congruence w.r.t. parallel composition,
- a novel compositional abstraction technique, based on common combined-transitions from concrete states, that helps defining *lower* and *upper* bounds on the maximum/minimum values of probabilistic and expected reachability measures, and
- the preservation of probabilistic and expected reachability properties for PA.

Organization. Section 2 sets the ground for this paper and introduces Markov chains, probabilistic automata and abstract PA. Sections 3 and 4 define a satisfaction and a refinement relation respectively. Section 5 discusses two different abstraction techniques whereas section 6 discusses reachability analysis of PA. Section 7 considers parallel composition of abstract PA and presents our compo-

sitionality results. Section 8 concludes the paper and provides pointers for future work. The complete paper with proofs of theorems can be found at http://www-i2.informatik.rwth-aachen.de/i2/publications/.

2 Background

Sub-distributions. A function μ is a *sub-distribution* on a finite set S iff $\mu : S \to [0,1]$ and $\sum_{s \in S} \mu(s) \leq 1$. Let $supp(\mu) = \{s \in S \mid \mu(s) > 0\}$ denote the support of μ, and the probability of a set $S' \subseteq S$ w.r.t. μ be given as $\mu(S') = \sum_{s \in S'} \mu(s)$. Let $|\mu| = \mu(S)$ denote the size of the sub-distribution μ. A sub-distribution is a *(full) distribution* iff $|\mu| = 1$. Let $Dist(S)$ denote the set of distributions over S. For $s \in S$, let $\iota_s \in Dist(S)$ denote the *Dirac* distribution, i.e, $\iota_s(s) = 1$. For sub-distributions μ and μ', the point-wise product $(\mu \cdot \mu') : S \times S \to [0,1]$ is given as: $(\mu \cdot \mu')(s,s') = \mu(s) \cdot \mu'(s')$ for $s, s' \in S$. A sub-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 sub-distribution is sometimes represented as $\mu = [\![\mu(s) \cdot s \mid s \in supp(\mu)]\!]$, where $[\![$ and $]\!]$ differentiate a set of probabilities from an ordinary set. For $0 \leq c \leq 1$, $c \cdot \mu$ denotes the sub-distribution defined by: $(c \cdot \mu)(s) = c \cdot \mu(s)$.

Constraint functions. Let φ be an arithmetic expression on variables over S denoting probabilities over S and let $sat(\varphi)$, referred to as the *satisfaction set* of φ, denote the set of distributions that satisfy φ. We call φ a *constraint function* and let $C(S)$ denote the set of constraint functions over S. For simplicity we use φ and $sat(\varphi)$ interchangeably. As every distribution is a constraint function, thus $Dist(S) \subseteq C(S)$. The satisfaction set of the product of two sub-constraint functions φ and φ' is denoted as $sat(\varphi \cdot \varphi') = sat(\varphi) \cdot sat(\varphi') = \{(\mu \cdot \mu') \mid \mu \in sat(\varphi), \mu' \in sat(\varphi')\}$.

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) $\{\} \in \mathcal{F}$ (2) $A \in \mathcal{F} \Rightarrow \Omega \backslash A \in \mathcal{F}$ (3)$A_1, A_2, A_3, ... \in \mathcal{F} \Rightarrow A_1 \cup A_2 \cup A_3 \cup ... \in \mathcal{F}$. The elements of $\mathcal{F}$ are *measurable sets* and $(\Omega, \mathcal{F})$ is a *measurable space*. A function $Prob : \mathcal{F} \to [0,1]$ is a *probability measure* on $(\Omega, \mathcal{F})$ iff $Prob(\Omega) = 1$ and if $A_1, A_2, ...$ are disjoint elements in $\mathcal{F}$, then $Prob(\bigcup_i A_i) = \sum_i Prob(A_i)$. $(\Omega, \mathcal{F}, Prob)$ is called a *measurable space*. For any $\mathcal{A} \subseteq \mathcal{F}$, there exists a unique smallest σ-field that contains $\mathcal{A}$ [2]; and given that $\mathcal{A}$ satisfies certain conditions [2], a *probability measure* defined on $\mathcal{A}$ can be uniquely extended to the σ-field generated from $\mathcal{A}$.

We recap our basic models: *Markov chains* (MC), *probabilistic automata* (PA) [15] and *abstract probabilistic automata* (APA) [8, 7]; the first two will act as implementations while the latter as specifications.

Markov Chain (MC). MC extend labelled transition systems (LTS) by annotating transitions with probabilities (instead of actions) that indicate the likelihood of their occurrences.

Definition 1. *A* Markov chain *(MC) is a tuple* $L = (S, \mathbf{P}, AP, \mathbf{V}, s_0)$ *where:*

– *S is a non-empty, finite set of states with initial state s_0,*
– *$\mathbf{P} : S \times S \to [0,1]$ is a transition probability function,*
– *AP is a finite set of valuations, and*
– *$\mathbf{V} : S \to 2^{AP}$ is a state-labeling function.*

The probability of going from state s to s' is given by $\mathbf{P}(s, s')$. We assume that the number of transitions emanating from a state is finite. A *path* of a MC represents an execution of the system it models. It is a non-empty, finite or infinite sequence of states i.e. $\pi = s_0 s_1 s_2 ...$ such that $\mathbf{P}(s_i, s_{i+1}) > 0$ for $i \geq 0$. For a finite path π_{fin}, let $|\pi_{fin}|$ denote its length (number of transitions) and $last(\pi_{fin})$ its final state. Let $\pi(i)$ denote state s_i of path π; and $\pi^{(i)}$ denote its *prefix* of length i i.e. $\pi(0)...\pi(i)$. The probability that a finite path π_{fin} is executed is given as: $\mathbf{P}(\pi_{fin}) = 1$ if $|\pi_{fin}| = 0$, otherwise $\mathbf{P}(\pi_{fin}) = \mathbf{P}(\pi_{fin}(0), \pi_{fin}(1)) \cdot ... \cdot \mathbf{P}(\pi_{fin}(n-1), \pi_{fin}(n))$, where $|\pi_{fin}| = n$. Finally, let $Path_{fin}(s)$ and $Path(s)$ denote all finite and infinite paths emanating from state s respectively. In the rest of the paper, $L = (S, \mathbf{P}, AP, \mathbf{V}, s_0)$ is assumed to be a MC.

Let a probability measure space $(Paths(s), \mathcal{F}_s, Prob_s)$ over the (infinite) paths emanating from state s be defined as follows. Let the *cylinder set* for finite path $\pi_{fin} \in Paths_{fin}(s)$ be $cyl(\pi_{fin}) = \{\pi \in Paths(s) \mid \pi_{fin} \text{ is a prefix of } \pi\}$. The probability of the cylinder set $cyl(\pi_{fin})$ is defined to be that of π_{fin}, i.e. $\mathbf{P}(cyl(\pi_{fin})) = \mathbf{P}(\pi_{fin})$. Let $Cyl(s) = \{cyl(\pi_{fin}) \mid \pi_{fin} \in Paths_{fin}(s)\}$ be the set of all cylinder sets defined on finite paths emanating from s; and let $\mathcal{F}_s$ be the smallest σ-field on $Paths(s)$ such that $Cyl(s) \subseteq \mathcal{F}_s$. $\mathbf{P}$ can be uniquely extended for $\mathcal{F}_s$ as $Prob_s$, thus, completing the definition of the probability measure space $(Paths(s), \mathcal{F}_s, Prob_s)$. Based on this construction, we can now define two measures for every MC, *probabilistic reachability* and *expected reachability*, that form the basis of probabilistic model checking [6, 3].

Probabilistic reachability refers to the probability of eventually reaching a state in L satisfying an element of $T \subseteq 2^{AP}$ starting from a given state s. Expected reachability refers to the expected number of transitions before reaching a state satisfying an element of T. Let $Paths_{fin}(s, T) = \{\pi_{fin} \in Paths_{fin}(s) \mid \exists i : \mathbf{V}(\pi(i)) \in T \wedge \forall j < i : \mathbf{V}(\pi(j)) \notin T\}$. Formally,

Definition 2. *For MC L, let $s \in S$ and $T \subseteq 2^{AP}$. Then, the reachability probability of a state satisfying an element in T from s is:*

$$p(L, s, T) = \sum_{\pi_{fin} \in Paths_{fin}(s,T)} Prob_s(cyl(\pi_{fin}))$$

and the expected reachability is: if $p(L, s, T) < 1$, then $e(L, s, T) = \infty$; otherwise

$$e(L, s, T) = \sum_{\pi_{fin} \in Paths_{fin}(s,T)} |\pi_{fin}| \cdot Prob_s(cyl(\pi_{fin})).$$

Probabilistic Automata (PA). PA extend labelled transition systems (LTS) by specifying the target of action-labelled transitions as *distributions over states*

instead of single states. Let *U-Act* be a countable universe of actions ranged over by a, b, etc; and let $Act(s)$ denote the set of enabled actions from state s.

Definition 3. *A* Probabilistic automaton *(PA) is a tuple* $M = (S, A, \Delta, AP, \mathbf{V}, s_0)$ *where:*

- S *is a non-empty, finite set of states with initial state* s_0*,*
- $A \subseteq$ *U-Act is a set of actions,*
- $\Delta \subseteq S \times A \times Dist(S)$ *is a set of transitions,*
- AP *is a finite set of valuations, and*
- $\mathbf{V} : S \to 2^{AP}$ *is a state-labeling function.*

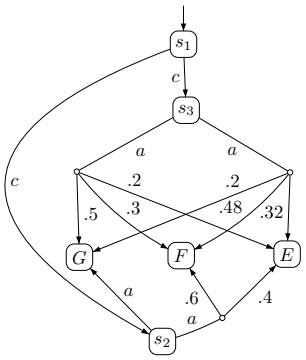

Fig. 1: A PA

We denote $(s, a, \mu) \in \Delta$ by $s \xrightarrow{a} \mu$ and assume that the number of transitions emanating from a state is finite. As $|Act(s)| \geq 1$ for a state $s \in S$, we may have a non-deterministic choice among the enabled actions in s. Therefore, a path in PA represents a particular resolution of non-determinism. Formally, a path from s_0 is given as: $\pi = s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} s_2 ...$ where $a_i \in Act(s_i)$ and $\mu_i(s_{i+1}) > 0$ for all $i \geq 0$.

A PA with $|supp(\mu)| = 1$ for every transition $s \xrightarrow{a} \mu$ is an LTS. Similarly, a Markov chain (MC) is a PA in which $|Act(s)| = 1$ for every state $s \in S$. In the rest of the paper, $M = (S, A, \Delta, AP, \mathbf{V}, s_0)$ is assumed to be a PA.

Example 1. Consider the PA given in Fig. 1. Note that the target of the transitions $s_2 \xrightarrow{a} [\![.4E, .6F]\!]$, $s_3 \xrightarrow{a} [\![.2E, .3F, .5G]\!]$ and $s_3 \xrightarrow{a} [\![.32E, .48F, .2G]\!]$ are distributions over states instead of single states. The target of other transitions are Dirac distributions.

To resolve a non-deterministic choice among enabled actions in a state s, a *scheduler* (also known as *policy, strategy or adversary*) is used that makes its decision based on the history of the execution up to that point. In this work, we only consider *deterministic memoryless schedulers* that map a finite path π_{fin} to one pair in $Act(last(\pi_{fin})) \times Dist(S)$. A scheduler κ is called *memoryless* iff $last(\pi_{fin}) = last(\pi'_{fin}) \Rightarrow \kappa(\pi_{fin}) = \kappa(\pi'_{fin})$ for any finite paths π_{fin} and π'_{fin}. A path π under a memoryless scheduler κ is of the form $\pi = s_0 \xrightarrow{a_0,\mu_0} s_1 \xrightarrow{a_1,\mu_1} s_2 ...$ where $(a_i, \mu_i) = \kappa(s_i)$ for $i \geq 0$. For $s \in S$ and scheduler κ, let $Paths^{\kappa}_{fin}(s)$ and $Paths^{\kappa}(s)$ denote the sets of finite and infinite paths emanating from s under κ. The behaviour of a PA under a scheduler κ is a MC. For PA M and scheduler κ, let M^{κ} be the induced MC. We can construct a measurable space $(Paths^{\kappa}(s_0), \mathcal{F}^{\kappa}, Prob^{\kappa})$ over the (infinite) paths of PA under scheduler κ. To reason about the probabilistic and expected reachability for PA, we consider their *minimum* and *maximum* values under all schedulers κ.

Definition 4. *For PA* M*, let* $T \subseteq 2^{AP}$ *and* $s \in S$*. The minimum and maximum values of probabilistic and expected reachability to a state satisfying an element in* T *from* s *are given as:*

– $p^{min}(M,s,T) = \inf_\kappa p(M^\kappa,s,T)$ *and* $p^{max}(M,s,T) = \sup_\kappa p(M^\kappa,s,T)$,
– $e^{min}(M,s,T) = \inf_\kappa e(M^\kappa,s,T)$ *and* $e^{max}(M,s,T) = \sup_\kappa e(M^\kappa,s,T)$.

As proved in [4, 1], there exist deterministic memoryless schedulers that optimize the values of probabilistic and expected reachability; and these values can be computed through *value iteration* [4, 1, 12] which gradually refines them to the required values, or by linear programming.

Abstract PA (APA) [8, 7]. Abstract PA act as specifications and aim to represent a set of PA. They extend PA by categorizing transitions/valuations into two types: *required* and *possible* ones. This distinction, adopted from modal transition systems (MTS) [9], is standard for abstract models of labelled transition systems. In addition, the target of transitions are *constraint* functions representing a set of distributions [5]. Altogether, this yields:

Definition 5. *An* abstract PA *is a tuple* $N = (S, A, \Delta_r, \Delta_p, AP, \mathbf{V}_r, \mathbf{V}_p, s_0)$ *with* S *(possibly empty),* A*,* AP *and* s_0 *as before, and:*

– $\Delta_r \subseteq S \times A \times C(S)$ *is a set of* required *transitions,*
– $\Delta_p \subseteq S \times A \times C(S)$ *is a set of* possible *transitions with* $\Delta_r \subseteq \Delta_p$,
– $\mathbf{V}_r : S \to 2^{AP}$ *maps a state to a set of* required *valuations, and*
– $\mathbf{V}_p : S \to 2^{AP}$ *maps a state to a set of* possible *valuations with* $\mathbf{V}_r(s) \subseteq \mathbf{V}_p(s)$ *for* $s \in S$.

When $(s, a, \varphi) \in \Delta_p$, we write $s \xrightarrow{a}_p \varphi$; and similarly when $(s, a, \mu) \in \Delta_r$, we write $s \xrightarrow{a}_r \varphi$. $s \xrightarrow{a}_p \varphi$ basically represents a set of a-transitions $s \xrightarrow{a}_p \mu$ from s with $\mu \in sat(\varphi)$. In figures, we subscript the actions of required transitions by "r", and those of possible transitions by "p". Each state in an APA is labelled with two sets of atomic propositions: required and possible ones. The required set of atomic propositions is satisfied as a whole by a state in an implementation, where a possible one may be satisfied and that too partially. As the distinction between possible and required propositions is similar to that for transitions, we will ignore propositions in examples. An APA with $\Delta_r = \Delta_p$, $\mathbf{V}_r(s) = \mathbf{V}_p(s)$ for $s \in S$, and $|sat(\varphi)| = 1$ for every required transition $s \to_r \varphi$ is a PA. Similarly, an APA in which for every $s \in S$, $\mathbf{V}_r(s) = \mathbf{V}_p(s)$ and for $s \xrightarrow{a}_p \varphi$, $sat(\varphi)$ is a Dirac distribution, is an MTS [9]. PA and MTS are thus proper sub-models of APA. In fact, PA will be used as implementations and APA as specifications—finite representations of a possibly infinite set of PA. Later on we will see that for the analysis of an APA, a finite set of its implementations suffices. In the rest of the paper, $N = (S, A, \Delta_r, \Delta_p, AP, \mathbf{V}_r, \mathbf{V}_p, s_0)$ is assumed to be an APA.

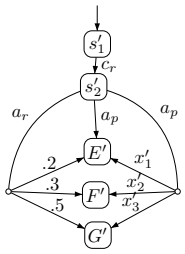

Fig. 2: An APA

Example 2. Fig. 2 represents an APA. Note that state s'_2 has one required transition, $s'_2 \xrightarrow{a}_r [\![.2E', .3F', .5G']\!]$ and two possible transitions, $s'_2 \xrightarrow{a}_p \varphi'_{x'}$ with $sat(\varphi'_{x'}) = \{[\![.4E', .6F']\!], [\![G']\!]\}$ and $s'_2 \xrightarrow{a}_p [\![\iota_{E'}]\!]$.

For the sake of convenience, we introduce the following notations and definitions. The notion of a *combined* transition, denoted $\xrightarrow{a}_C$, arises as a convex combination of a set of transitions with the same label $a \in A$.

Definition 6. (Combined transition) *For PA M, let $s \in S$ and $\mu \in Dist(S)$. We write $s \xrightarrow{a}_C \mu$, if there is a finite indexed set $\{(c_i, \mu_i)\}_{i \in I}$, $c_i \in \mathbb{R}_{\geq 0}$ and $\mu_i \in Dist(S)$, such that $s \xrightarrow{a} \mu_i$ for each $i \in I$, $\sum_{i \in I} c_i = 1$, and $\mu = \bigoplus_{i \in I} c_i \cdot \mu_i$.*

For APA we define $s \xrightarrow{a}_{pC} \varphi$ and $\mu \xrightarrow{a}_{rC} \varphi$ in a similar way for combined possible and required transitions respectively. For a set of distributions $B = \{\mu_1, \mu_2, ..., \mu_n\}$, let $B^C = \{\mu \mid \mu = \bigoplus_{i=1}^{n} c_i \cdot \mu_i \wedge \sum_i^n c_i = 1\}$. Similarly, we have φ^C for a constraint function φ. Moreover, we write $s \overset{a}{\rightrightarrows}_p \varphi'$, called *multi a-transition* from s, iff $s \xrightarrow{a}_p \varphi_1$, $s \xrightarrow{a}_p \varphi_2$,..., $s \xrightarrow{a}_p \varphi_n$ and $\varphi' = \bigvee_{i=1}^{n} \varphi_i$. It is important to note that $sat(\bigvee_{s \xrightarrow{a}_{pC} \varphi} \varphi) \subseteq sat(\bigvee_{s \xrightarrow{a}_p \varphi} \varphi)^C$. The multi-transitions extend for PA by default.

Simulation/refinement [10] is a preorder on the state space requiring that whenever state u simulates state s, then u can mimic at least the stepwise behaviour of s. This can be lifted to distributions over states as:

Definition 7. (Simulation) *Let S be a finite, non-empty set of states, and let $\mu, \mu' \in Dist(S)$. 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) $\sum_{s \in S} \Delta(s, v) = \mu'(v)$.*

3 Satisfaction

A satisfaction relation formally relates a PA, i.e., an implementation, with an APA, i.e., a specification.

Definition 8. (Satisfaction) *Let PA M and APA N' have identical sets of actions A and atomic propositions AP. A relation $R \subseteq S \times S'$ is a* satisfaction *relation if for every sRs':*

1. *$s' \overset{a}{\rightrightarrows}_r \varphi'$ implies $s \overset{a}{\rightrightarrows} \varphi$ and for every $\mu' \in sat(\varphi')$, there exists $\mu \in sat(\varphi)^C$ such that $\mu R \mu'$.*
2. *$s \overset{a}{\rightrightarrows} \varphi$ implies $s' \overset{a}{\rightrightarrows}_p \varphi'$ with $\varphi R {\varphi'}^C$,*
3. *$\mathbf{V}'_r(s') \subseteq \mathbf{V}(s)$ and $\mathbf{V}(s) \subseteq \mathbf{V}'_p(s')$.*

Here, $\varphi R {\varphi'}^C$ iff for every $\mu \in sat(\varphi)$, there exists $\mu' \in sat(\varphi')^C$ with $\mu R \mu'$. PA M satisfies *or* implements *APA N', denoted $M \models N'$, iff there exists a satisfaction relation relating s_0 and s'_0. The set of implementations of N' is given as $\wr N' \wr = \{M \mid M \models N'\}$.*

Intuitively, if a state s satisfies a state s', then (1) whenever s' performs a required multi a-transition to a constraint function φ', then s can perform a multi a-transition to a constraint function φ such that every distribution in $sat(\varphi')$ is satisfied by some distribution in $sat(\varphi)^C$; (2) whenever s performs a multi a-transition to a constraint function φ, this can be mimicked by s' by possibly moving to a constraint function φ' such that every $\mu \in sat(\varphi)$ satisfies one distribution in $sat(\varphi')^C$; and (3) s should at least satisfy all required valuations of s' and its all valuations should be derived from that of s'.

Example 3. The PA M in Fig. 1 is an implementation of the AMA N in Fig. 2 as $R = \{(s_1, s_1'), (s_2, s_2'), (s_3, s_2'), (G, G'), (E, E'), (F, F')\}$ is a satisfaction relation. Let us check whether (s_2, s_2') fulfils the conditions of Def. 8. For the required a-transition from s_2' to $[\![.2E', .3F', .5G']\!]$, there is a corresponding multi a-transition from s_2 to a constraint function with satisfaction set $\{[\![.4E, .6F]\!], \iota_G\}$, and $[\![.2E, .3F, .5G]\!]$ is a convex combination of distributions in $\{[\![.4E, .6F]\!], \iota_G\}$. As $[\![.2E, .3F, .5G]\!]$ satisfies $[\![.2E', .3F', .5G']\!]$ ($[\![.2E, .3F, .5G]\!] R [\![.2E', .3F', .5G']\!]$), condition (1) is fulfilled. For the a-transitions from s_2 to ι_G and $[\![.4E, .6F]\!]$, there are corresponding a-transitions from s_2' to $\iota_{G'}$ and $[\![.4E', .6F']\!]$ respectively. Thus, condition (2) is also fulfilled.
Now let us check how the pair (s_3, s_2') fulfils condition (2). Note that there is no simple or combined a-transition from s_2' that satisfies a-transition from s_3 to $[\![.32E, .48F, .2G]\!]$. However, if we consider a multi a-transition from s_2' to $\{[\![.4E, .6F]\!], \iota_E, \iota_G\}$, we get $[\![.32E', .48F', .2G']\!]$, a convex combination of distributions in $\{[\![.4E, .6F]\!], \iota_E, \iota_G\}$, that satisfies $[\![.32E, .48F, .2G]\!]$. Had we just simple transitions in Definition 8, state s_3 would not have been an implementation of state s_2'. (Recall that valuations are not considered in our examples.)

4 Refinement

In this section, we discuss *refinement* that is used to compare APA. Intuitively, a refinement relation compares two APA w.r.t. their sets of implementations. If APA N refines APA N', then we aim at $\wr N \wr \subseteq \wr N' \wr$. Refinement takes a similar view as satisfaction.

Definition 9. (Refinement) *Let APA N and N' have identical sets of actions A and atomic propositions AP. A relation $R \subseteq S \times S'$ is a* refinement *relation if for every sRs':*

1. $s' \overset{a}{\rightrightarrows}_r \varphi'$ *implies* $s \overset{a}{\rightrightarrows}_r \varphi$ *and for every* $\mu' \in sat(\varphi')$, *there exists* $\mu \in sat(\varphi)^C$ *such that* $\mu R \mu'$.
2. $s \overset{a}{\rightrightarrows}_p \varphi$ *implies* $s' \overset{a}{\rightrightarrows}_p \varphi'$ *with* $\varphi R {\varphi'}^C$,
3. $\mathbf{V}_r'(s') \subseteq \mathbf{V}(s)$ *and* $\mathbf{V}(s) \subseteq \mathbf{V}_p'(s')$.

Let $\preceq$ be the largest refinement relation. N refines N', *denoted $N \preceq N'$, iff there exists a refinement relation relating s_0 and s_0'.*

The above definition is a simple generalization of that of satisfaction. Condition (1) is similar with that of Def. 8 except that the multi a-transition from s now must be a required transition. Condition (2) adds that for every possible multi a-transition from s, there should be a corresponding possible multi a-transition from s'. The similar addition is found in condition (3). Evidently, a satisfaction relation is a special case of a refinement relation.

Definition 10. (Bisimulation) $\sim = \preceq \cap \preceq^{-1}$.

Let us recall the definitions of *Segala's strong probabilistic simulation* and *bisimulation* [15, 13] and see how Segala's bisimulation is related to $\sim$.

Definition 11. *Let PA M and M' have identical sets of actions A. A relation $R \subseteq S \times S'$ is a* strong simulation *relation if for every sRs':*

$s \xrightarrow{a} \mu$ implies $s' \xrightarrow{a} \mu'$ with $\mu R \mu'$.

M is simulated by M', denoted $M \sqsubseteq M'$, iff there exists a simulation relation relating s_0 and s'_0.

A relation $R \subseteq S \times S'$ is a strong bisimulation *relation if for every sRs':*

$s \xrightarrow{a} \mu$ implies $s' \xrightarrow{a} \mu'$ with $\mu R \mu'$ and $s' \xrightarrow{a} \mu'$ implies $s \xrightarrow{a} \mu$ with $\mu' R \mu$.

M is strongly bisimilar to M', denoted $N \simeq M'$, iff there exists a strong bisimulation relation relating s_0 and s'_0.

Lemma 1. *$\sim$ coincides with $\simeq$ for any PA.*

Proposition 1. *For PA M,M' and APA N, N':*

- *$N \preceq N'$ implies $\wr N \int \subseteq \wr N' \int$.*
- *$M \sim M'$ implies for every N, $M \models N$ iff $M' \models N$.*

5 Abstraction

This section explains two techniques of abstracting an APA that mimics all behaviours of the concrete one. Intuitively, the state space S of APA N is partitioned and each partition is represented by a single state in the abstract APA N'. Formally, let abstraction function $\alpha : S \to S'$ be a surjection and its inverse a concretization function $\gamma : S' \to 2^S$. That is, $\alpha(s)$ is an abstract state of s whereas $\gamma(s')$ is the set of concrete states abstracted by s'. These notions can be lifted to distributions in a simple way: an abstract distribution $\alpha(\mu)$ of μ is given as $\alpha(\mu)(s') = \mu(\gamma(s'))$. α and γ are lifted to sets of states or sets of distributions in a point-wise manner. Thus, $\varphi' = \alpha(\varphi)$ iff $sat(\varphi') = \alpha(sat(\varphi))$.

In the sequel, we assume w.l.o.g. that for APA N, the abstraction function $\alpha : S \to S'$ induces the APA $N' = \alpha(N)$ with the same set of actions A and atomic propositions AP.

Definition 12. (Strong abstraction (SA)) *For APA N, the abstraction function $\alpha : S \to S'$ induces the APA $N' = \alpha(N)$ where for all $s' \in S'$:*

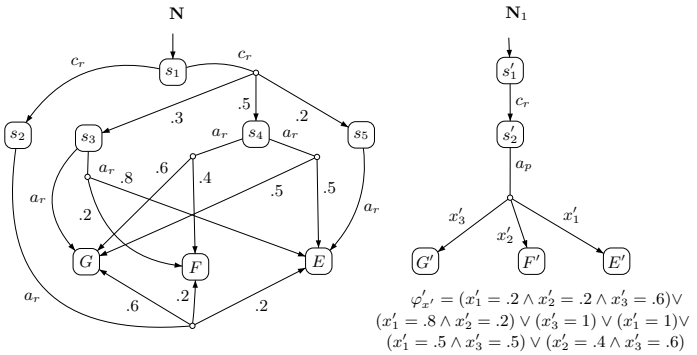

Fig. 3: $N_1 = \alpha(N)$

1. *for every* $a \in A$ *and* $\varphi' \in C(S')$,
   (a) $s' \xrightarrow{a}_r \varphi'$ *iff* $\forall s \in \gamma(s'), \varphi \in C(S)$*:* $s \xrightarrow{a}_r \varphi$ *and* $\varphi' = \alpha(\varphi)$,
   (b) $s' \xrightarrow{a}_p \varphi'$ *iff* $\exists s \in \gamma(s'), \varphi \in C(S)$*:* $s \xrightarrow{a}_p \varphi$*, and* $\varphi' = \bigvee_{u \in \gamma(s'): u \xrightarrow{a}_p \varphi} \alpha(\varphi)$.
2. $\mathbf{V}'_r(s') = \mathbf{V}_r(s)$ *for* $s \in \gamma(s')$*, and* $\mathbf{V}'_p(s') = \bigcup_{s \in \gamma(s')} \mathbf{V}_p(s)$.

(1a) Intuitively, if each state in $\gamma(s')$ has a required a-transition, then s' also has a required a-transition and vice versa. Moreover, the abstract behaviour of each concrete transition should be the same as that of the transition from s'. (1b) s' has a possible a-transition iff at least one state in $\gamma(s')$ has a possible a-transition. The accumulative abstract behaviour of all concrete possible transitions is the same as that of the a-transition from s'. (2) The set of required valuations of s' should be the same as that of each concrete state $s \in \gamma(s')$, whereas its set of possible valuations is the union of the sets of possible valuations of concrete states.

Example 4. For $N_1 = \alpha(N)$ (see Fig. 3), $\gamma(s'_1) = \{s_1\}$, $\gamma(s'_2) = \{s_2, s_3, s_4, s_5\}$, $\gamma(E') = \{E\}$, $\gamma(F') = \{F\}$ and $\gamma(G') = \{G\}$. Note that the abstract behaviour of both required c-transitions from s_1 is the same, and is represented by one required c-transition from s'_1. The required a-transitions from s_2, s_3, s_4 and s_5 behave differently; this is mimicked by the possible a-transition from s'_2.

Consider two states s_1 and s_2 with two required transitions from each: $s_1 \xrightarrow{a}_r \mu_1$, $s_1 \xrightarrow{a}_r \nu_1$ and $s_2 \xrightarrow{a}_r \mu_2$, $s_2 \xrightarrow{a}_r \nu_2$. Let $\alpha(\mu_1) \neq \alpha(\nu_1) \neq \alpha(\mu_2) \neq \alpha(\nu_2)$. It means there is no required a-transition from the abstract state $\alpha(s_1) = \alpha(s_2)$. Let $\varphi_1 = \{\mu_1, \nu_1\}^C$ and $\varphi_2 = \{\mu_2, \nu_2\}^C$ such that $\alpha(\varphi_1) \cap \alpha(\varphi_2) \neq \emptyset$. Then it is possible to have a required a-transition from the abstract state with $\varphi' = \alpha(\varphi_1) \cap \alpha(\varphi_2)$ as the target constraint function. As an example, observe that there are combined a-transitions from s_3 and s_4 (Fig. 3) that have common target distributions e.g. $[\![.2E, .2F, .6G]\!]$. We therefore adapt SA by exploiting these common target distributions of (only) combined required transitions as:

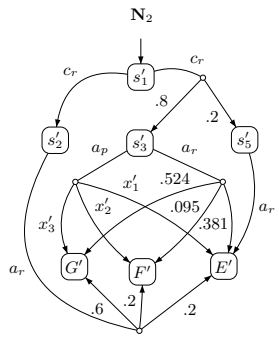

Fig. 4: $N_2 = \alpha_c(N)$

Definition 13. (Common-distribution Abstraction (CDA)) *For APA N, the abstraction function $\alpha_c : S \to S'$ induces the APA $N' = \alpha_c(N)$ where for all $s' \in S'$:*

1. *for every $a \in A$ and $\varphi' \in C(S')$,*
   (a) $s' \xrightarrow{a}_r \varphi'$ *iff* $\forall s \in \gamma_c(s'), \varphi \in C(S)$*:* $s \xrightarrow{a}_r \varphi$ *and* $\varphi' = \bigwedge_{u \in \gamma_c(s') : u \overset{a}{\rightrightarrows}_r \varphi} \alpha_c(\varphi)^C$,
   (b) $s' \xrightarrow{a}_p \varphi'$ *iff* $\exists s \in \gamma_c(s'), \varphi \in C(S)$*:* $s \xrightarrow{a}_p \varphi$ *and* $\varphi' = \bigvee_{u \in \gamma_c(s') : u \xrightarrow{a}_p \varphi} \alpha_c(\varphi)$
2. $\mathbf{V}'_r(s') = \mathbf{V}_r(s)$ *for* $s \in \gamma_c(s')$*, and* $\mathbf{V}'_p(s') = \bigcup_{s \in \gamma_c(s')} \mathbf{V}_p(s)$.

(1a) Like in SA, if each state in $\gamma_c(s')$ has a required a-transition, then s' has a required a-transition and vice versa. However, the common target distributions (after abstraction) among all the combined required a-transitions from the concrete states should be the target of the required a-transition from s'. (1b) and (2) are the same as in SA.

Example 5. Let $N_2 = \alpha_c(N)$ as in Fig. 4 with $\gamma_c(s'_1) = \{s_1\}$, $\gamma_c(s'_2) = \{s_2\}$, $\gamma_c(s'_3) = \{s_3, s_4\}$, $\gamma_c(s'_5) = \{s_5\}$, $\gamma_c(E') = \{E\}$, $\gamma_c(F') = \{F\}$ and $\gamma_c(G') = \{G\}$. The common behaviour of combined a-transitions from s_3 and s_4 is given by the required a-transition from the abstract state s'_3. The possible a-transition from s'_3 represents all a-transitions from s_3 and s_4.

The following results show that the above notions of abstraction yield APA and preserve refinement. Finally, we show that our notions of abstraction are comparable w.r.t. refinement.

Lemma 2. *For APA N, $\alpha(N)$ and $\alpha_c(N)$ are APA.*

Theorem 1. *For APA N, $N \preceq \alpha(N)$ and $N \preceq \alpha_c(N)$.*

Lemma 3. *For APA N and $s \in S$, if $\alpha(s) = \alpha_c(s)$ then $N \preceq \alpha_c(N) \preceq \alpha(N)$.*

6 Reachability

In order to analyse the behaviour of APA, we need to resolve three different types of nondeterminism. The first one is due to possible transitions/valuations that may either be present or absent in an implementation. The second type of nondeterminism is the same as in PA, i.e., the nondeterministic choice among the enabled actions from each state. Like for PA, deterministic schedulers are used for APA to choose among the enabled actions. The third source of nondeterminism is the target constraint function of a transition; one of the distributions in the satisfaction set of the constraint function is nondeterministically chosen. As the satisfaction set of a constraint function may be infinite (and even uncountable), we approximate it by a finite set. We consider the approximation of polynomial constraint functions as they are closed under composition [8].

For $i \in \mathbb{N}^+$, let Θ_i and Φ_i be linear constraint functions in variables over S. Let $\varphi_\iota = \{\iota_s \mid s \in S\}$ be a linear constraint function characterizing Dirac distributions over S, and φ_μ be a linear constraint function characterizing only one distribution, i.e., μ.

Consider a polynomial constraint function ϕ representing a set of distributions over S; therefore, it at least contains a linear constraint $\sum_{s\in S} x_s = 1$ which implies that $\phi^C \subseteq \varphi_\iota^C$. We can now deduce a series of linear constraint functions $\varphi_0 = \varphi_\iota$, $\varphi_1 = \varphi_0 \wedge \Theta_1$, $\varphi_2 = \varphi_1 \wedge \Theta_2$ and so on such that $\phi^C \subseteq \varphi_{i+1}^C \subseteq \varphi_i^C$ for all $i \geq 0$ and $\phi^C = \lim_{i\to\infty} \varphi_i^C$. Every φ_i in the above series over-approximates ϕ, i.e., every μ in ϕ^C also exists in φ_i^C.

Now we under-approximate ϕ by a linear constraint function. Let $\mu \in \phi$ such that $\varphi_\mu = \mu$ is a linear constraint function. As in the above case there exists a series of constraint functions $\varphi_0 = \varphi_\mu$, $\varphi_1 = \varphi_0 \vee \Phi_1$, $\varphi_2 = \varphi_1 \vee \Phi_2$ and so on such that $\phi^C \supseteq \varphi_{i+1}^C \supseteq \varphi_i^C$ for all $i \geq 0$ and $\phi^C = \lim_{i\to\infty} \varphi_i^C$. Every φ_i in the above series under-approximates ϕ, i.e., every μ in φ_i^C also exists in ϕ^C.

Definition 14. *A constraint-approximating function $\varsigma : C(S) \to C(S)$ over-approximates a polynomial constraint function by a linear one iff for polynomial constraint functions $\phi, \phi_1, \phi_2 \in C(S)$:*

- *$\varsigma(\phi)$ is a linear constraint function and $\phi^C \subseteq \varsigma(\phi)^C$, and*
- *$\phi_1^C \subseteq \phi_2^C \Rightarrow \varsigma(\phi_1)^C \subseteq \varsigma(\phi_2)^C$.*

The inverse function ς^{-1} under-approximates a polynomial constraint function by a linear one, i.e., $\phi^C \supseteq \varsigma^{-1}(\phi)^C$ and $\phi_1^C \supseteq \phi_2^C \Rightarrow \varsigma^{-1}(\phi_1)^C \supseteq \varsigma^{-1}(\phi_2)^C$.

Example 6. Consider a polynomial constraint function $\phi = (x^2 + y^2 + z^2 \leq r^2 \wedge x + y + z = 1)$ representing a set of distributions by a shaded-circular region of radius r within each triangle of Fig. 5 and 6. Let ς_1 and ς_2 be the constraint-approximating functions such that $\varsigma_1(\phi)$ represents the region enclosed among the lines l_1, l_2, l_3 and the sides of the left triangle in Fig. 5; and $\varsigma_2(\phi)$ represents the region among the lines $l_1, \ldots, l_{12}$ in the right triangle. Let $\varsigma_2^{-1}(\phi)$ represent

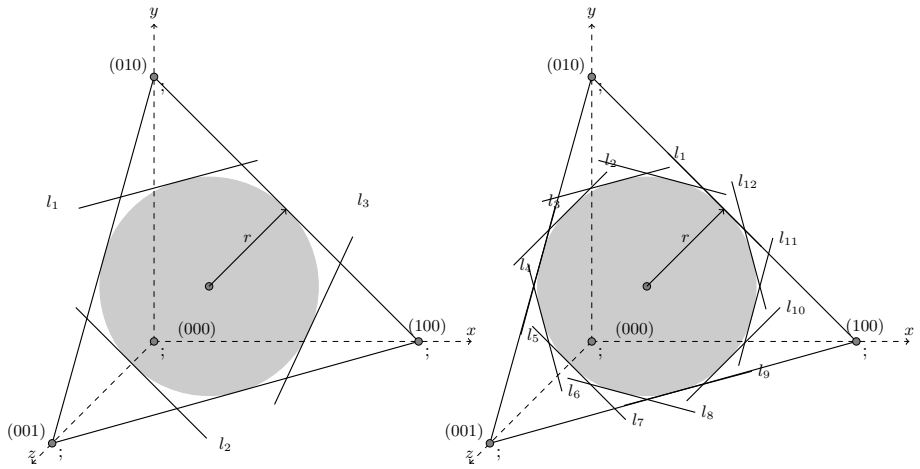

Fig. 5: An example polynomial constraint function (left) and a linear over-approximation (right).

the region among the lines $l_1, \ldots l_6$ in Fig. 6. It is clear that $\varsigma_2(\phi)$ is an over-approximation and $\varsigma_2^{-1}(\phi)$ is an under-approximation of ϕ, i.e., $\varsigma_2^{-1}(\phi) \subseteq \phi \subseteq \varsigma_2(\phi)$. Moreover, $\varsigma_2(\phi)$ gives a better over-approximation of ϕ than $\varsigma_1(\phi)$, i.e., $\phi \subseteq \varsigma_2(\phi) \subseteq \varsigma_1(\phi)$.

It implies that an APA with polynomial constraint functions can be further abstracted by over/under-approximating its constraint functions with linear ones. The following definition lifts ς from constraint functions to APA.

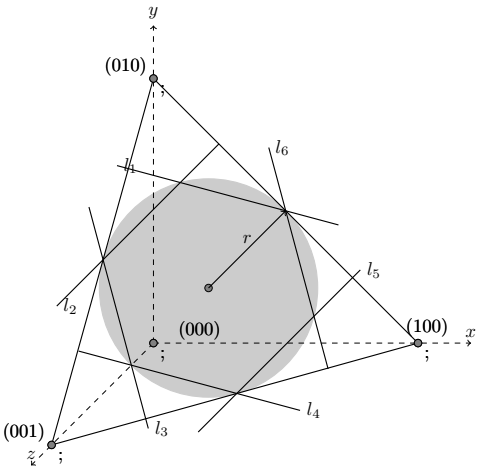

Fig. 6: Linear under-approximation of a polynomial constraint function.

Definition 15. *For APA N with polynomial constraints, the constraint-approximating function $\varsigma : C(S) \to C(S)$ induces the APA $N' = \varsigma(N)$ with the same set of states S, actions A, atomic propositions AP and state-labeling functions, where for all $s \in S$,*

1. $s \xrightarrow{a}_r \varsigma^{-1}(\phi) \in \Delta'_r$ *iff* $s \overset{a}{\rightrightarrows}_r \phi \in \Delta_r$,
2. $s \xrightarrow{a}_p \varsigma(\phi) \in \Delta'_p$ *iff* $s \overset{a}{\rightrightarrows}_p \phi \in \Delta_p$.

(1) As by Definition 14 $\varsigma^{-1}(\phi) \subseteq \phi$, this implies that the set of required a-transitions $s \xrightarrow{a}_r \varsigma^{-1}(\phi)$ from state s in N' is a subset of that of from s in N. (2) However, in the case of possible transitions from s, it is the other way around as $\phi \subseteq \varsigma(\phi)$. This leads to the follow lemma:

Lemma 4. *For APA N, $N \preceq \varsigma(N)$.*

The satisfaction set of a linear constraint function may be infinite, but this can be simplified by just considering its *extreme distributions*.

Definition 16. *The set of* extreme distributions *of a linear constraint function φ, denoted φ_{extr}, is the smallest finite subset of φ such that $\varphi^C_{extr} = \varphi^C$.*

The concept of extreme distributions is explained in [11] for interval constraints that can easily be extended for linear constraints. Thus, an APA with linear constraints can be simplified by just considering the extreme distributions of its constraint functions. Let N_{extr} represents an APA in which every transition $s \xrightarrow{a}_p \varphi$ is replaced with concrete transitions $s \xrightarrow{a}_p \mu$ where $\mu \in sat(\varphi_{extr})$. In the following we assume that every APA N with linear constraints is simplified, i.e., $N = N_{extr}$.

Extreme refinements: Now we consider two refinements of APA N; one in which every possible transition/valuation is converted into a required one and the other in which they are all removed. Let $N^{\uparrow}$ and $N^{\downarrow}$ be the refinements of APA N with $\Delta^{\uparrow}_r = \Delta_p$ and $\mathbf{V}^{\uparrow}_r(s) = \mathbf{V}_p(s)$; and $\Delta^{\downarrow}_p = \Delta_r$ and $\mathbf{V}^{\downarrow}_p(s) = \mathbf{V}_r(s)$ for $s \in S$ respectively. $N^{\uparrow}$ and $N^{\downarrow}$ are called *extreme refinements* of N.

Consistency: Note that as there do not exist possible transitions/valuations in APA $N^{\downarrow}$, we may have some state, say s, in $N^{\downarrow}$ with $\mathbf{V}^{\downarrow}_p(s) = \emptyset$. Such states are called *inconsistent*. Formally, a state s in $N^{\downarrow}$ is *inconsistent* if either $\mathbf{V}^{\downarrow}_p(s) = \emptyset$ or $s \overset{a}{\Rightarrow}_r \varphi$ implies either $sat(\varphi) = \emptyset$ or there exists a distribution in $sat(\varphi)$ that assigns a positive mass to at least one state s' with $\mathbf{V}^{\downarrow}_p(s') = \emptyset$. We call such a distribution *inconsistent*. As discussed in [8], such inconsistent states can be iteratively removed from the system by a *pruning operator* β. This process is repeated until there are no more inconsistent states in the system. If the resulting system contains at least one state, we say that it is consistent; otherwise it is inconsistent. (Further details about pruning can be found in [8].)

The following lemma tells how Segala's strong probabilistic simulation and bisimulation [15, 13] relate implementations of APA N to $\varsigma(N)^{\uparrow}$ and $\varsigma(N)^{\downarrow}$.

Lemma 5. *For APA N and for each PA $M \in \wr N \wr$, $M \sqsubseteq \varsigma(N)^{\uparrow}$ and $\varsigma(N)^{\downarrow} \sqsubseteq M$.*

As the whole behaviour of each implementation M of APA N is derived from N (conditions (2) and (3) of Definition 8), this implies that $M \sqsubseteq \varsigma(N)^{\uparrow}$. As every implementation M of N depicts at least the required behaviour of N (condition (1) of Definition 8), this implies that $\varsigma(N)^{\downarrow} \sqsubseteq \varsigma(N^{\downarrow}) \sqsubseteq M$. Note that $\varsigma(N)^{\downarrow} \neq \varsigma(N^{\downarrow})$. This is because every required transition is also a possible transition and ς over approximates every possible transition in $\varsigma(N^{\downarrow})$.

Moreover, extreme refinements of APA N, and abstractions $\alpha(N)$ and $\alpha_c(N)$ are related by Definition 11 as:

Lemma 6. *For APA N, $\varsigma(\alpha(N))^{\uparrow} \simeq \varsigma(\alpha_c(N))^{\uparrow}$, $\varsigma(N)^{\downarrow} \sqsubseteq \varsigma(\alpha(N))^{\uparrow}$ and $\varsigma(\alpha(N))^{\downarrow} \sqsubseteq \varsigma(\alpha_c(N))^{\downarrow} \sqsubseteq \varsigma(N)^{\downarrow}$.*

The proof of $\varsigma(\alpha(N))^{\uparrow} \simeq \varsigma(\alpha_c(N))^{\uparrow}$ follows from the fact that the set of possible transitions of $\alpha(N)$ and $\alpha_c(N)$ are the same. As $N \preceq \alpha(N)$, this leads to $\varsigma(N)^{\downarrow} \sqsubseteq \varsigma(\alpha(N))^{\uparrow}$. Moreover, the proof of $\varsigma(\alpha(N))^{\downarrow} \sqsubseteq \varsigma(\alpha_c(N))^{\downarrow} \sqsubseteq \varsigma(N)^{\downarrow}$ follows from the fact that $N \preceq \alpha_c(N) \preceq \alpha(N)$, i.e., the whole required behaviour of $\alpha(N)$ is present in $\alpha_c(N)$ and subsequently in N.

Based on the above lemma we, as a main theorem of this paper, give *lower* and *upper bounds* for maximum/minimum reachability/expected reachability values for PA. In our case as we assume that every PA has only one initial state s_0, therefore, for simplicity we can write $s_0 = \alpha(s_0) = \alpha_c(s_0)$.

Theorem 2. *For PA M and $x \in \{e, p\}$, let $T \subseteq 2^{AP}$, and $M_1 = \varsigma(\alpha(M))^{\uparrow}$, $M_2 = \varsigma(\alpha(M))^{\downarrow}$ and $M_3 = \varsigma(\alpha_c(M))^{\downarrow}$ be PA. Then,*

- $x^{max}(M_2, s_0, T) \leq x^{max}(M_3, s_0, T) \leq x^{max}(M, s_0, T) \leq x^{max}(M_1, s_0, T)$,
- $x^{min}(M_1, s_0, T) \leq x^{min}(M, s_0, T) \leq x^{min}(M_3, s_0, T) \leq x^{min}(M_2, s_0, T)$.

The proof of the above theorem is based on the fact that $M_2 \sqsubseteq M_3 \sqsubseteq M \sqsubseteq M_1$. The bounds given in the above theorem are dependent on the constraint-approximating function ς. The better the ς is, the tighter the bounds will be.

7 Parallel Composition

We define a composition operation that allows to combine two APA. It 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 APA; other actions can occur autonomously. The following definition is just an extension of the parallel composition definition of APA in [8] with multi transitions.

Definition 17. (Parallel composition) *For APA N and N', the parallel composition w.r.t. synchronization set $\bar{A} \subseteq (A \cap A')$ is given as: $N \|_{\bar{A}} N' = (S \times S', A \cup A', \tilde{\Delta}_r, \tilde{\Delta}_p, AP \cup AP', \tilde{\mathbf{V}}_r, \tilde{\mathbf{V}}_p, (s_0, s_0'))$, where for all $a \in A \cup A'$, $(s, s') \in S \times S'$ and $\tilde{\varphi} \in C(S \times S')$:*

1. $(s, s') \overset{a}{\rightrightarrows}_r \tilde{\varphi}$ *iff one of the following holds:*
   (a) $a \in \bar{A}$, $s \overset{a}{\rightrightarrows}_r \varphi$, $s' \overset{a}{\rightrightarrows}_r \varphi'$ *and* $\tilde{\varphi} = \varphi \cdot \varphi'$, *or*
   (b) $a \in A$, $s \overset{a}{\rightrightarrows}_r \varphi$ *and* $\tilde{\varphi} = \varphi \cdot \iota_{s'}$, *or*
   (c) $a \in A'$, $s' \overset{a}{\rightrightarrows}_r \varphi'$ *and* $\tilde{\varphi} = \iota_s \cdot \varphi'$.
2. $(s, s') \overset{a}{\rightrightarrows}_p \tilde{\varphi}$ *iff one of the following holds:*
   (a) $a \in \bar{A}$, $s \overset{a}{\rightrightarrows}_p \varphi$, $s' \overset{a}{\rightrightarrows}_p \varphi'$ *and* $\tilde{\varphi} = \varphi \cdot \varphi'$, *or*
   (b) $a \in A$, $s \overset{a}{\rightrightarrows}_p \varphi$ *and* $\tilde{\varphi} = \varphi \cdot \iota_{s'}$, *or*
   (c) $a \in A'$, $s' \overset{a}{\rightrightarrows}_p \varphi'$ *and* $\tilde{\varphi} = \iota_s \cdot \varphi'$.
3. $\tilde{\mathbf{V}}_r((s, s')) = \mathbf{V}_r(s) \cup \mathbf{V}'_r(s')$ *and* $\tilde{\mathbf{V}}_p((s, s')) = \mathbf{V}_p(s) \cup \mathbf{V}'_p(s')$.

In (1a) both s and s' synchronize and perform required multi a-transitions, whereas in (1b) and (1c) they behave independently. Condition (2) considers possible multi transitions from s and s'.

Theorem 3. *For any set $\bar{A}$, $\preceq$ is a pre-congruence w.r.t. $||_{\bar{A}}$.*

The composite APA is exponentially larger in size as compared to the composing ones. This problem could be avoided by applying abstraction prior to composition. The following result shows that the resulting APA is the same as we get by first applying the composition operator to individual APA and then abstracting the monolithic one.

Theorem 4. *For APA N_1 and N_2, synchronization set $\bar{A}$ and abstraction functions α_1, α_2 of the same type: $\alpha_1(N_1)\,||_{\bar{A}}\,\alpha_2(N_2) = (\alpha_1 \times \alpha_2)(N_1||_{\bar{A}}N_2)$ up to isomorphism, where $\alpha_1 \times \alpha_2$ is defined as $(\alpha_1 \times \alpha_2)((s, s')) = (\alpha_1(s), \alpha_2(s'))$.*

8 Conclusion

This paper presented novel compositional abstraction techniques for probabilistic automata (PA) as well as a new refinement relation which is pre-congruence w.r.t. parallel composition. The key idea is to find out common combined-transitions from a set of concrete states and put them as required transitions in the abstract state. Moreover, for the analysis and verification of PA, reachability and expected reachability properties are also discussed. We expect the layered composition operator, defined in [17] for PA, can be extended for APA. Future work includes the application of this technique to practical case studies and the development of a counterexample-guided abstraction-refinement framework.

References

1. Alfaro, L. d.: Computing minimum and maximum reachability times in probabilistic systems. In: *Proceedings of the 10th International Conference on Concurrency Theory. LNCS*, Vol. 1664. Springer (1999) 66–81
2. Ash, R. B., Doléans-Dade, C. A.: *Probabilty & Measure Theory, 2nd Edition.* Academic Press (2000)
3. Baier, C., Kwiatkowska, M.: Model checking for a probabilistic branching time logic with fairness. *Distributed Computing* **11** (1998) 125–155
4. Bertsekas, D. P., Tsitsiklis, J. N.: An analysis of stochastic shortest path problems. *Mathematics of Operations Research* **16** (1991) 580–595
5. Caillaud, B., Delahaye, B., Larsen, K. G., Legay, A., Pedersen, M. L., Wasowski, A.: Constraint Markov chains. *Theor. Comput. Sci.* **412** (2011) 4373–4404
6. de Alfaro, L.: *Formal Verification of Probabilistic Systems.* PhD thesis, Stanford University, Stanford, CA, USA (1998)
7. Delahaye, B., Katoen, J.-P., Larsen, K., Legay, A., Pedersen, M., Sher, F., Wasowski, A.: New results on abstract probabilistic automata. In: *Application of Concurrency to System Design (ACSD), 2011 11th International Conference on.* IEEE CS Press (2011) 118 –127

8. Delahaye, B., Katoen, J.-P., Larsen, K. G., Legay, A., Pedersen, M. L., Sher, F., Wasowski, A.: Abstract probabilistic automata. In: *VMCAI. LNCS*, Vol. 6538. Springer (2011) 324–339
9. Huth, M., Jagadeesan, R., Schmidt, D.: Modal transition systems: A foundation for three-valued program analysis. In: *ESOP. LNCS*, Vol. 2028. Springer (2001) 155–169
10. Jonsson, B., Larsen, K. G.: Specification and refinement of probabilistic processes. In: *Logic in Computer Science*. IEEE CS Press (1991) 266–277
11. Katoen, J.-P., Klink, D., Neuhäußer, M. R.: Compositional abstraction for stochastic systems. In: *FORMATS. LNCS*, Vol. 5813. Springer (2009) 195–211
12. 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** (2010) 246–280
13. Lynch, N. A., Segala, R., Vaandrager, F. W.: Observing branching structure through probabilistic contexts. *SIAM J. Comput.* **37** (2007) 977–1013
14. Milner, R.: *Communication and Concurrency*. Prentice-Hall, Inc. (1989)
15. Segala, R.: *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology (1995)
16. Segala, R., Lynch, N. A.: Probabilistic simulations for probabilistic processes. *Nordic J. of Computing* **2** (1995) 250–273
17. Swaminathan, M., Katoen, J.-P., Olderog, E.-R.: Layered reasoning for randomized distributed algorithms. *Formal Aspects of Computing* (2012 (to appear))