A Proof System for Compositional Verification of Probabilistic Concurrent Processes

Mio, M., & Simpson, A. (2013). A Proof System for Compositional Verification of Probabilistic Concurrent Processes. In Pfenning, F. (Ed.), Foundations of Software Science and Computation Structures. (pp. 161-176). (Lecture Notes in Computer Science). Springer Berlin / Heidelberg. doi: 10.1007/978-3-642-37075-5_11

A Proof System for Compositional Verification of Probabilistic Concurrent Processes

Matteo Mio[1]* and Alex Simpson[2]

[1] INRIA and LIX, Ecole Polytechnique, France
[2] LFCS, School of Informatics, University of Edinburgh, Scotland

Abstract. We present a formal proof system for compositional verification of probabilistic concurrent processes. Processes are specified using an SOS-style process algebra with probabilistic operators. Properties are expressed using a probabilistic modal μ-calculus. And the proof system is formulated as a sequent calculus in which sequents are given a quantitative interpretation. A key feature is that the probabilistic scenario is handled by introducing the notion of *Markov proof*, according to which proof trees contain probabilistic branches and are required to satisfy a condition formulated by interpreting them as Markov Decision Processes. We present simple but illustrative examples demonstrating the applicability of the approach to the compositional verification of infinite state processes. Our main result is the soundness of the proof system, which is proved by applying the coupling method from probability theory to the game semantics of the probabilistic modal μ-calculus.

1 Introduction

In recent years, *model checking* has established itself as a powerful and widely applicable method for verifying properties of systems, with its techniques adaptable to systems embodying, for example, concurrency, real-time behaviour and probabilistic choice, see [1] for a detailed overview. However, model checking has its limitations. In particular, its applicability is typically restricted to finite-state systems, or to carefully crafted classes of infinite-state systems. Moreover, even in the finite-state case, the applicability of model checking is limited by the *state explosion* problem: the state space of a concurrent system grows exponentially in the number of parallel components.

Many phenomena in computer science give rise to infinite-state systems when modelled at a natural level of abstraction. So it is important to have verification methods that can cope with such systems. Since infinite-state systems are

* This research was partially supported by PhD studentships from LFCS and the IGS at the School of Informatics, University of Edinburgh; by EPSRC research grant EP-F042043-1; by project ANR-09-BLAN-0169-01 PANDA; and by project ANR-11-IS02-0002 LOCALI. It was completed during the tenure of an ERCIM "Alain Bensoussan" Fellowship, supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

defined using finite descriptions (given using a programming language, process calculus or similar language for system specification), one seeks verification methods that relate descriptions of systems to their properties. Such methods cannot, in general, be fully automatic since, for most interesting cases, the problem of ascertaining whether a description satisfies a property is undecidable.

One general methodology for obtaining such broader verification methods is to develop *formal proof systems* tailored to the goal of establishing that (descriptions of) systems satisfy properties. The verification task then becomes one amenable to the technology of computer-assisted reasoning. An important desideratum for a proof system for verification is that it should support *compositional* reasoning methods, by which the the task of establishing a property of a complex system is broken down into suitable verification goals for the components of the system. As has been extensively discussed in the literature, see, e.g., [9], such compositional methods support methodologies for the modular development and verification of systems. Compositional methods also provide a route to taming the state explosion problem, since the size of a compound system is usually much larger than that of its components.

The purpose of this paper is to present one interesting instantiation of the above general approach to verification. We develop a formal proof system for compositional reasoning about (possibly infinite state) concurrent probabilistic systems. The systems we deal with are ones described by a simple algebra for concurrent probabilistic processes, with SOS-style operational semantics (Section 2). While, in this paper itself, we consider only a few basic process operators, a key feature of our approach is that it is applicable to a wide class of operators (any that can be described in the *probabilistic* GSOS framework of [2]).

Properties are specified using the *probabilistic* (a.k.a. *quantitative*) *modal μ-calculus* (pLμ) introduced independently in [10,12] (Section 3). Our reason for not using a standard logic for stating properties of probabilistic systems, such as PCTL [1, §10.2], is that fixed-point logics such as pLμ appear to be better adapted to compositional reasoning. One reason for this is that formulas express properties of states rather than properties of objects of higher complexity, such as paths or Markov chains. Also, powerful proof methods for reasoning about fixed points are available. Nevertheless, as the first author has shown [13], the full expressivity of PCTL (and beyond) can be recovered by extending pLμ with a few additional operators. This makes it plausible that the proof system of the present paper might be similarly extended to provide a system capable of reasoning about arbitrary PCTL-properties.

The main contributions of the paper are the proof system itself and its (non-trivial) soundness theorem (Section 4). Adapting a general methodology for compositional verification, expounded in [19], the proof system is a sequent calculus with sequents of assertions of the form $p : F$. In our quantitative setting, the semantics of $p : F$ is a real number in $[0, 1]$, which roughly (see Section 3 for clarification) expresses the probability that property F holds of process p. The right-hand side of a sequent is a multiset of such assertions, itself given a quantitative meaning as the Łukasiewicz disjunction (see, e.g., [8] and [10]) of the

individual assertions. The use of this disjunction from fuzzy logic underpins several features of our proof system, most importantly the soundness of certain crucial proof rules which allow probabilistic choices within different processes to be *coupled* in the reasoning.

To enable the proof system to handle the fixed points in the logic, we allow cyclic derivations and require a combinatorial condition to hold in order for a proof to count as valid. This approach is familiar from proof systems for the ordinary (non-probabilistic) modal μ-calculus [15,21] and other fixed-point logics [3]. However, there is a twist in our probabilistic setting. One of the proof rules in our system introduces probabilistic branching into the proof tree itself. This is addressed by interpreting the proof tree as specifying a Markov Decision Process (MDP), in which the participant, Refuter, is trying to refute the correctness of the proof. Refuter's goal is to try to find an infinite branch through the proof along which all sequences of fixed-point unfoldings illegitimately unfold a least-fixed-point infinitely often. The proof tree is declared valid just in case Refuter almost surely fails in his endeavour; that is when the *value* of the MDP is zero. Due to the critical role played by the probabilistic rule, we call such a valid proof tree a *Markov proof*. An important fact, crucial for the applicability of the approach, is that the property of being a Markov proof is decidable. This follows from known decidability results for one-player stochastic parity games [5].

In Section 5, we present two examples of Markov proofs, illustrating the sort of compositional reasoning possible within our system and establishing nontrivial properties of infinite state systems.

Related work: Several approaches to compositional verification methods for concurrent probabilistic systems have received attention in the recent literature. In Cardelli *et. al.* [4], new 'spatial' operators are added to a probabilistic modal logic to support compositional reasoning about labelled Markov processes [16] enriched with an algebraic structure defining composition of systems, and a completeness result is obtained for a Hilbert-style axiomatization. However, the logic is limited to expressing *local* properties of systems (that is, it cannot state properties of infinite runs). In Kwiatkowska *et. al.* [11,7], assume-guarantee techniques for compositional verification of (parallel composition of) probabilistic automata [18] are developed. Their approach does handle some global properties of systems, namely safety and liveness properties expressed using automata. However, being based on fully automatizable model-checking techniques, it is restricted to finite models. Similarly, Larsen *et. al.* [6] introduce compositional methodologies for design and verification of finite probabilistic concurrent systems. These are based on *Abstract Probabilistic Automata* which are structures capable of modelling both specifications and implementations, and closed under natural logical operations such as composition and refinement.

Distinguishing features of our approach are: we have a clear separation between the process language and a purely behavioural *endogenous* [17] logic; the nested fixed-points of the logic allow the specification of complex global behaviour; and we are able to establish nontrivial properties of infinite state systems.

2 Probabilistic Concurrent Processes

Ordinary *Labeled Transition Systems* (LTS) allow the description of processes exhibiting nondeterministic behavior. In his PhD thesis, R. Segala introduced a new class of models, nowadays known as *Probabilistic Labeled Transition Systems* (PLTS), for modelling processes exhibiting both nondeterministic and probabilistic behaviours. Since their introduction, PLTS's have been successfully adopted as models for formal languages describing concurrent probabilistic systems, such as the class of PGSOS languages of [2] among others.

Definition 1 (PLTS [18]). *Given a countable set L of labels, a* Probabilistic Labeled Transition System *(PLTS) is a pair $\mathcal{L} = \langle P, \{\xrightarrow{a}\}_{a\in L}\rangle$, where P is a set of states and $\xrightarrow{a} \subseteq P \times \mathcal{D}(P)$, for every $a \in L$, where $\mathcal{D}(P)$ denotes the set of discrete probability distributions over P.*

The intended interpretation of a PLTS $\mathcal{L} = \langle P, \{\xrightarrow{a}\}_{a\in L}\rangle$ is the following: the process states $p \in P$ represent the possible configurations of the system. At a process state p, the system can react to an a-action, for $a \in L$, by changing its state to a process q in accordance with some nondeterministically chosen probability distribution $d \in \mathcal{D}(P)$ such that $p \xrightarrow{a} d$.

$$\frac{}{a.x \xrightarrow{a} \delta(x)}\ \text{prefix}$$

$$\frac{x \xrightarrow{a} \alpha}{x|y \xrightarrow{a} \{\alpha \rightsquigarrow z\}\delta(z|y)}\ |\ \text{left} \qquad \frac{y \xrightarrow{a} \alpha}{x|y \xrightarrow{a} \{\alpha \rightsquigarrow z\}\delta(x|z)}\ |\ \text{right}$$

$$\frac{x \xrightarrow{a} \alpha}{!x \xrightarrow{a} \{\alpha \rightsquigarrow z\}\delta(z|!x)}\ ! \qquad \frac{x \xrightarrow{a} \alpha}{!^{\frac{1}{2}}x \xrightarrow{a} \alpha +_{\frac{1}{2}} \{\alpha \rightsquigarrow y\}\delta(y|!^{\frac{1}{2}}x)}\ !^{\frac{1}{2}}$$

Fig. 1. SOS Rules. The letter a ranges over a fixed set L of labels.

In this paper, we consider PLTS's described by a few very simple process operators chosen to present the examples in Section 5. Our approach, however, adapts straightforwardly to handle arbitrary process algebras described by means of well-behaved operational rules (such as, e.g., the PGSOS format of [2]). The term constructors we consider are the constant 0 denoting the inactive process, the *prefix operation* $(a._-)$ of arity 1, the non-communicating asynchronous parallel composition $(\_ \mid \_)$ of arity 2, the *bang* operator $(!\_)$ and a probabilistic variant of it $(!^{\frac{1}{2}}\_)$, both of arity 1. Their semantics, specified by the SOS operational rules of Figure 1, allows the derivation of statements of the form $p \xrightarrow{a} d$ where p is a *process term*, the letter a is a label and d is a *process distribution term*. Process distribution terms, denoting probability distributions over processes, are specified by the syntax: $d, e ::= \alpha \mid \delta(p) \mid d +_\lambda e \mid \{d \rightsquigarrow x\}e$, where α is a *process distribution variable* and $\lambda \in [0, 1]$. The term $\delta(p)$ denotes the (Dirac) probability distribution that proceeds deterministically onward to p, and $d +_\lambda e$ denotes the probabilistic choice that chooses d with probability λ and e with probability

$1-\lambda$. The distribution term $\{d \rightsquigarrow x\}e$, first, randomly chooses a process p in accordance with probability distribution d, and then proceeds as $e[p/x]$.

A *SOS-model*, or just a *model*, is a PLTS equipped with sound interpretations for all process constructors under consideration (see, e.g., [19] and [2] for general definitions). In the case of prefix, for example, the PLTS $\langle P, \{\xrightarrow{a}\}_{a\in L}\rangle$ is required to come with a function $f_{a.\_} : P \to P$ for which $f_{a.\_}(p) \xrightarrow{b} d$ holds if and only if $a=b$ and d is the Dirac distribution with mass at p, for every $p\in P$. In what follows we reserve the letter M to range over models.

Definition 2 (Interpretations). *Given a model $M=\langle P, \{\xrightarrow{a}\}_{a\in L}\rangle$, an* interpretation of the variables *is a function γ mapping process-variables x to states $p\in P$ and process distribution-variables α to probability distributions $d\in\mathcal{D}(P)$. The map γ extends uniquely to a function from process-terms to P and to process distribution terms to $\mathcal{D}(P)$, defined as expected. In particular*

$$\gamma\big(\{d \rightsquigarrow x\}e\big)(p) \stackrel{\text{def}}{=} \sum_{q\in P} \Big(\gamma\big(d\big)(q)\cdot \gamma[q/x]\big(e\big)(p)\Big)$$

where $\gamma[q/x](x)=q$ and $\gamma[q/x](y)=\gamma(y)$ for all variables $x\neq y$.

3 Probabilistic Modal μ-Calculus (pLμ)

The *probabilistic* (or *quantitative*) modal μ-calculus (pLμ) [14,12] is a fixed-point logic designed for expressing properties of PLTS's. The syntax of pLμ formulas is the same of the standard modal μ-calculus (Lμ) [20].

Definition 3. *Given a countable set of propositional variables* Var *ranged over by the letters X, Y, Z and a set of labels L ranged over by the letters a, b, c, the formulas of the logic pLμ (in positive form) are defined by the following grammar:*

$$F, G ::= X \mid \langle a\rangle F \mid [a]\,F \mid F\vee G \mid F\wedge G \mid \mu X.F \mid \nu X.F$$

As usual the operators $\nu X.F$ and $\mu X.F$ bind the variable X in F.

Definition 4 (Denotational Semantics [12]). *Given $\mathcal{L}=\langle P, \{\xrightarrow{a}\}_{a\in L}\}$, the denotational semantics of the pLμ formula F under the interpretation $\rho:\mathrm{Var}\to(P\to[0,1])$, is the map $[\![F]\!]_\rho : P\to[0,1]$ defined by structural induction on F as:*

$$(\!|\, X \,|\!)_\rho(p) = \rho(X)(p)$$

$$(\!|\, G\vee H \,|\!)_\rho(p) = (\!|\, G\,|\!)_\rho(p) \sqcup (\!|\, H\,|\!)_\rho(p) \qquad (\!|\, G\wedge H \,|\!)_\rho(p) = (\!|\, G\,|\!)_\rho(p) \sqcap (\!|\, H\,|\!)_\rho(p)$$

$$(\!|\, \langle a\rangle G \,|\!)_\rho(p) = \bigsqcup\big\{(\!|\, G\,|\!)_\rho(d) \mid p \xrightarrow{a} d\big\} \qquad (\!|\, [a]\,G \,|\!)_\rho(p) = \bigsqcap\big\{(\!|\, G\,|\!)_\rho(d) \mid p \xrightarrow{a} d\big\}$$

$$(\!|\, \mu X.G \,|\!)_\rho(p) = \mathrm{lfp}\big(\lambda f.((\!|\, G\,|\!)_{\rho[f/X]})\big)(p) \qquad (\!|\, \nu X.G \,|\!)_\rho(p) = \mathrm{gfp}\big(\lambda f.((\!|\, G\,|\!)_{\rho[f/X]})\big)(p)$$

where $\sqcup$, $\sqcap$, lfp and gfp denote the join, meet, least and greatest fixed point operations of the complete lattice $[0,1]$ with its standard order, and $(\!|\, F\,|\!)_\rho(d)$ is defined as $(\!|\, F\,|\!)_\rho(d)=\sum_{p\in P} d(p)\cdot(\!|\, F\,|\!)_\rho(p)$.

It is easy to verify that the interpretation assigned to every pLμ operator is monotone. Thus, the existence of the least and greatest fixed points is guaranteed by the Knaster-Tarski theorem. Although it is convenient to consider open formulas when defining the semantics of pLμ, logical specifications are generally formulated as closed formulas. For this reason, and for simplifying the presentation of the proof system in Section 4, we shall only consider closed formulas in the rest of this paper. Thus we omit the interpretation ρ from $(\!| F |\!)_\rho$ and just write $(\!| F |\!)$. Given a formula F we denote with $\neg F$ its De Morgan dual obtained by replacing every connective appearing in F with its dual. Note that $F = \neg\neg F$. As customary, we denote with $\top$ the pLμ formula $\nu X.X$ and define $\bot = \neg\top$.

Proposition 5. *For every PLTS $\mathcal{L} = \langle P, \{\stackrel{a}{\longrightarrow}\}_{a\in L}\rangle$, the equalities $(\!| \top |\!)(p) = 1$, $(\!| \bot |\!)(p) = 0$ and $(\!| \neg F |\!)(p) = 1 - (\!| F |\!)(p)$ hold, for all $p \in P$.*

It is often suggestive to think of the value of $(\!| F |\!)(p)$ as representing the probability that a property asserted by F holds for p. Technically, this is justified by an alternative semantics for pLμ, which interprets a formula as the *value* of a two-player stochastic parity game [12]. The game in question is obtained by running the usual two-player game for the modal μ-calculus formula over the PLTS. As with ordinary modal μ-calculus games, game configurations $p : \langle a\rangle F$ and $p : [a]\, F$ are under the control of different players (here called Maximizer and Minimizer respectively) whose move is to choose an a transition out of p. In the case of pLμ, the destination of this transition is a probability distribution, and Nature intercepts in the game to make the probabilistic choice. The winning condition for Maximizer is the usual one that a greatest fixed-point gets unfolded infinitely often. One can then think of the value of the game as the (upper limit) probability with which Maximizer is able to verify the property expressed by the ordinary μ-calculus formula F.

It is a nontrivial fact that the game interpretation of a pLμ formula coincides with the denotational one of Definition 4. This was originally shown just for finite PLTS's in [12], and only recently for general PLTS's in [14].

4 Proof System

We introduce in this section our proof system designed to reason about pLμ-calculus properties of processes given in our process algebra. The system is a *sequent calculus*, in which sequents have the form $\Sigma \vdash \Delta$, where Σ and Δ are multisets of (different kinds of) assertions. We use the letter J to range over *operational assertions* in Σ which are either of the form $d \simeq e$, where d and e are process distribution terms, or of the form $p \stackrel{a}{\longrightarrow} d$, where p is a process term, a is an action-label and d is a process distribution term. We use the letters ϕ and ψ to range over *logical assertions* in Δ, which are of the form $p : F$ or $d : F$, where F is a closed pLμ formula, p a process term and d a distribution term. Given an assertion ϕ of the form $t : F$, with $t \in \{p, d\}$, we write $\neg\phi$ for the assertion $t : \neg F$.

Definition 6 (Semantics of assertions). *Given a model M and an interpretation of the variables γ, the meaning $[\![\_]\!]^M_\gamma$ of the assertions is defined as:*

1. $[\![d \simeq e]\!]^M_\gamma = 1$ *if* $\gamma(d) = \gamma(e)$ *and* $[\![d \simeq e]\!]^M_\gamma = 0$ *otherwise.*
2. $[\![p \xrightarrow{a} d]\!]^M_\gamma = 1$ *if* $\gamma(p) \xrightarrow{a}_M \gamma(d)$ *and* $[\![p \xrightarrow{a} d]\!]^M_\gamma = 0$ *otherwise.*
3. $[\![t : F]\!]^M_\gamma \stackrel{\text{def}}{=} (\!|\, F \,|\!)\big(\gamma(t)\big)$*, for any process or distribution term* t*.*

We write $(M, \gamma) \models J$ *if the equality* $[\![J]\!]^M_\gamma = 1$ *holds. We write* $(M, \gamma) \models \Sigma$*, for* $\Sigma = J_1, \ldots, J_m$*, if for all* $i \in \{1, \ldots, m\}$ *it holds that* $(M, \gamma) \models J_i$*.*

Note that the value $[\![J]\!]^M_\gamma$ of a logical assertion is either 1 or 0, whereas $[\![t : F]\!]^M_\gamma$ lies anywhere in $[0, 1]$, representing, using the informal reading of pLμ discussed in Section 3, the probability of the property expressed by F holding at $\gamma(t)$. In order to extend the semantics from assertions to sequents, we first recall basic notions from *Łukasiewicz* logic [8].

Definition 7 ([8]). *The operations* $\oplus : [0, 1]^2 \to [0, 1]$ *and* $\neg : [0, 1] \to [0, 1]$ *defined as* $x \oplus y = \min\{x + y, 1\}$ *and* $\neg x = 1 - x$*, are known as* Łukasiewicz *disjunction and negation. The induced conjuction* ($\ominus$) *and implication* ($\Rightarrow$) *operations are defined as* $x \ominus y = \neg(\neg x \oplus \neg y)$ *and* $x \Rightarrow y = \neg x \oplus y$*. Note that* $((x \ominus y) \Rightarrow z) = (x \Rightarrow (\neg y \oplus z))$ *and* $(x \Rightarrow y) = 1$ *if and only if* $x \leq y$*.*

Definition 8 (Semantics of Sequents). *Let* $\Sigma \vdash \Delta$ *be a sequent with* $\Sigma = J_1, \ldots, J_n$ *and* $\Delta = \phi_1, ..., \phi_m$*. Given a model* M *and an interpretation* γ *of the variables, we define the semantics* $[\![\Sigma \vdash \Delta]\!]^M_\gamma \in [0, 1]$ *of the sequent as:*

$$[\![\Sigma \vdash \Delta]\!]^M_\gamma \stackrel{\text{def}}{=} \big([\![J_1]\!]^M_\gamma \ominus \ldots \ominus [\![J_n]\!]^M_\gamma\big) \Rightarrow \big([\![\phi_1]\!]^M_\gamma \oplus \ldots \oplus [\![\phi_m]\!]^M_\gamma\big)$$

Note that, since each $[\![J_i]\!]^M_\gamma$ *is in* $\{0, 1\}$*, so is the value of the antecedent. We write* $(M, \gamma) \models \Sigma \vdash \Delta$ *if* $[\![\Sigma \vdash \Delta]\!]^M_\gamma = 1$*. A sequent is* valid*, written* $\models \Sigma \vdash \Delta$*, if* $(M, \gamma) \models \Sigma \vdash \Delta$ *for every pair* (M, γ)*.*

The choice of considering a quantitative semantics of sequents is naturally motivated by the $[0, 1]$-valued semantics of the logic pLμ. Among the many possible choices (several are studied in *fuzzy logic* [8]) for interpreting commas in sequents, Łukasiewicz logic enjoys pleasant properties. Its operators coincide with the ordinary boolean ones when arguments have values in $\{0, 1\}$. Furthermore Ł-negation coincides with pLμ negation (see Proposition 5) and interacts well with Ł-disjunction (as in Definition 7). This allows our one-sided (in the set of logical assertions) formulation of sequents. Lastly, and most importantly, Ł-disjuction validates a sound probabilistic interpretation of some key rules of the proof system (see Proposition 15 below). The validity of a sequent $\Sigma \vdash \Delta$ expresses a form of implication: for every model (M, γ) satisfying all the operational assertions in Σ, the sum of the values of the (interpreted) assertions in Δ is at least 1. Valid sequents express nontrivial relations between the probabilities associated to the logical assertions in Δ. For example, the validity of a sequent of the form $\Sigma \vdash \neg\phi_1, \neg\phi_2, \phi$ can be understood as follows: in every model (M, γ) satisfying the operational assertions in Σ, the value of the assertion ϕ is bounded below by a function ($\ominus$) of the values of ϕ_1 and ϕ_2. We now briefly discuss a few illustrative examples of valid sequents showing how the chosen quantitative interpretation allows the expression of interesting properties.

Example 9. Define $F \stackrel{\text{def}}{=} \mu X.\,[a]\,X$. The following sequents are valid:

$$\text{Seq}_1 \stackrel{\text{def}}{=} x \xrightarrow{a} \alpha \vdash x : \langle a \rangle \top \qquad \text{Seq}_3 \stackrel{\text{def}}{=} \emptyset \vdash x : \neg F, y : \neg F, x|y : F$$
$$\text{Seq}_2 \stackrel{\text{def}}{=} \emptyset \vdash x|y : \neg F, x : F \qquad \text{Seq}_4 \stackrel{\text{def}}{=} \emptyset \vdash !^{\frac{1}{2}}(a.0) : F$$

The first example expresses a trivial property: if a process x can perform an a-labeled transition then it satisfies the pLμ formula $\langle a \rangle \top$ with probability 1. The meaning of the pLμ formula F above can be understood as expressing a termination goal (i.e., the impossibility of producing an infinite sequence of a's) under an adversary environment. Thus the sequent Seq_2 expresses a simple property: the parallel (non-communicating) system $x|y$ with two components has termination probability less than or equal to that of its components. The third sequent Seq_3 expresses a slightly less obvious property, providing a lower bound on the termination probability for $x|y$. Lastly, the fourth sequent Seq_4 expresses the fact that the process $!^{\frac{1}{2}}(a.0)$ terminates with probability 1, i.e., almost surely. All the sequents above can be proven valid by our proof system. In Section 5 we present proofs for Seq_3 and Seq_4.

Before introducing the derivation rules of our system, we introduce an auxiliary kind of judgement useful for expressing entailments between operational assertions. We shall consider *operational judgments* of the form $\Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$.

Definition 10. *Given an operational judgment $\Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$, we write $(M, \gamma) \models \Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$ when the following implication holds: if $(M, \gamma) \models \Sigma$ then there is some $i \in \{0, \ldots, n\}$ such that $(M, \gamma^i) \models \Sigma_i$, for some interpretation γ^i that agrees with γ on all (process and distribution) variables appearing in Σ. We say that $\Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$ is* valid*, written $\models \Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$, if for every pair (M, γ) it holds that $(M, \gamma) \models \Sigma \rhd \{\Sigma_i\}_{0 \le i \le n}$. Note that $\models \Sigma \rhd \{\emptyset\}$ holds, $\models \Sigma \vdash \emptyset$ holds iff Σ is not satisfiable and $\emptyset \rhd \{\Sigma\}$ holds iff $(M, \gamma) \models \Sigma$ for all (M, γ). In order to improve readability, we just write $\Sigma \vdash J$ instead of $\Sigma \vdash \{\{J\}\}$.*

Example 11. The following are examples of valid operational judgments:

1. $0 \xrightarrow{a} \alpha \rhd \emptyset$ 2. $\emptyset \rhd \{\{a.x \xrightarrow{a} \alpha, \alpha \simeq \delta(x)\}\}$ 3. $a.x \xrightarrow{b} \alpha \rhd \emptyset$, if $b \neq a$
4. $x|y \xrightarrow{a} \alpha \rhd \{\Sigma_{\mathcal{L}}, \Sigma_{\mathcal{R}}\}$ 5. $!x \xrightarrow{a} \alpha \rhd \left\{\left\{x \xrightarrow{a} \beta, \alpha \simeq \{\beta \rightsquigarrow y\}\delta(y|!x)\right\}\right\}$
5. $\emptyset \rhd \alpha \simeq \{\alpha \rightsquigarrow x\}\delta(x)$ 6. $\emptyset \rhd \alpha +_{\frac{1}{4}} \beta \simeq (\alpha +_{\frac{1}{2}} \beta) +_{\frac{1}{2}} \beta$

where $\Sigma_{\mathcal{L}} = x \xrightarrow{a} \beta, \alpha \simeq \{\beta \rightsquigarrow x'\}\delta(x'|y)$ and $\Sigma_{\mathcal{R}} = y \xrightarrow{a} \beta, \alpha \simeq \{\beta \rightsquigarrow y'\}\delta(x|y')$.

The derivation rules for our main proof system for quantitative sequents are presented in Figure 2. The rule Σ-Rule supports reasoning about the operational semantics by means of case analysis, using a side-condition exploiting the semantic validity of operational judgements (Definition 10). In practice, this semantic side-condition can be replaced with a formal proof system for proving validity of operational judgements, which can be constructed following established approaches (see, e.g., "action assertions rules" in [19, p. 18]). For lack of space, we do not go into further details about this, focussing instead on our main proof system for quantitative sequents, whose design is significantly more intricate.

$$\frac{\{\Sigma_i \vdash \Delta\}_{i\in I}}{\Sigma \vdash \Delta}\ \Sigma\text{-Rule}\ \Big(\text{proviso: } \Sigma \rhd \{\Sigma_i\}_{i\in I}\Big)$$

$$\frac{\Sigma \vdash \Gamma, \psi \qquad \Sigma \vdash \Delta, \neg\psi}{\Sigma \vdash \Gamma, \Delta}\ \text{Cut}$$

$$\frac{\Sigma \vdash \Delta}{\Sigma[q/x] \vdash \Delta[q/x]}\ \text{P-Sub} \qquad \frac{\Sigma \vdash \Delta}{\Sigma[d/\alpha] \vdash \Delta[d/\alpha]}\ \text{D-Sub}$$

$$\frac{\Sigma[e/\alpha] \vdash \Delta[e/\alpha]}{\Sigma[d/\alpha] \vdash \Delta[d/\alpha]}\ \Sigma\text{-Sub}\ \Big(\text{proviso: } \Sigma \rhd d \simeq e\Big)$$

$$\frac{\Sigma \vdash p\!:\!F_i, \Delta}{\Sigma \vdash p\!:\!F_1 \vee F_2, \Delta}\ \vee_i \quad i \in \{1,2\} \qquad \frac{\Sigma \vdash p\!:\!F, \Delta \qquad \Sigma \vdash p\!:\!G, \Delta}{\Sigma \vdash p\!:\!F \wedge G, \Delta}\ \wedge$$

$$\frac{\Sigma \vdash d\!:\!F, \Delta}{\Sigma \vdash p\!:\!\langle a\rangle F, \Delta}\ \langle a\rangle \Big(\text{proviso: } \Sigma \rhd p \xrightarrow{a} d\Big) \qquad \frac{\Sigma, p \xrightarrow{a} \alpha \vdash \alpha\!:\!F, \Delta}{\Sigma \vdash p\!:\![a]\,F, \Delta}\ [a] \quad \alpha \text{ fresh}$$

$$\frac{\Sigma \vdash p\!:\!F[\mu X.F/X], \Delta}{\Sigma \vdash p\!:\!\mu X.F, \Delta}\ \mu \qquad \frac{\Sigma \vdash p\!:\!F[\nu X.F/X], \Delta}{\Sigma \vdash p\!:\!\nu X.F, \Delta}\ \nu$$

$$\frac{\Sigma \vdash \Delta, p\!:\!F}{\Sigma \vdash \Delta, \delta(p)\!:\!F}\ \delta$$

$$\frac{\Sigma \vdash \Delta, d_1\!:\!F_1, \ldots, d_n\!:\!F_n \qquad \Sigma \vdash \Delta, e_1\!:\!F_1, \ldots, e_n\!:\!F_n}{\Sigma \vdash \Delta, d_1 +_\lambda e_1\!:\!F_1, \ldots, d_n +_\lambda e_n\!:\!F_n}\ +_\lambda \quad \lambda \in (0,1)$$

$$\frac{\Sigma \vdash \Delta, e_1[y/x_1]\!:\!F_1, \ldots, e_n[y/x_n]\!:\!F_n}{\Sigma \vdash \Delta, \{d \rightsquigarrow x_1\}e_1\!:\!F_1, \ldots, \{d \rightsquigarrow x_n\}e_n\!:\!F_n}\ \{\rightsquigarrow\} \quad y \text{ fresh.}$$

Fig. 2. Derivation rules.

A typical usage of the Σ-Rule is better explained by means of a simple example. Consider the valid sequent $x|y \xrightarrow{a} \alpha \vdash x : \langle a\rangle\top, y : \langle a\rangle\top$ asserting that in every model such that $x|y$ can make an a-transition then either x or y or both can make an a-transition (this qualitative interpretation holds since $(\!|\langle a\rangle\top|\!)(p) \in \{0,1\}$ in every model). The crucial step in proving its validity is:

$$\frac{\Sigma_{\mathcal{L}} \vdash x\!:\!\langle a\rangle\top, y\!:\!\langle a\rangle\top \qquad \Sigma_{\mathcal{R}} \vdash x\!:\!\langle a\rangle\top, y\!:\!\langle a\rangle\top}{x|y \xrightarrow{a} \alpha \vdash x\!:\!\langle a\rangle\top, y\!:\!\langle a\rangle\top}\ \Sigma\text{-Rule: } x|y \xrightarrow{a} \alpha \rhd \{\Sigma_{\mathcal{L}}, \Sigma_{\mathcal{R}}\}$$

where $\Sigma_{\mathcal{L}}$ and $\Sigma_{\mathcal{R}}$ are as in Example 11. This step performs the required case analysis, based on the operational semantics of the (non-communicating) parallel operator, required to distinguish the two relevant cases. Both premises above are easily seen to be valid (see also Seq_1 in Example 9). Note that the only axiom rule (i.e., rule without premises) in the proof system is the instance of the Σ-Rule when the proviso is of the form $\Sigma \rhd \emptyset$, i.e., when Σ is unsatisfiable. We refer to this particular use of this rule as Σ-Axiom. For example, the sequent $\emptyset \vdash 0\!:\![a]\,\bot$ can be proved as follows,

$$\cfrac{\cfrac{}{0 \xrightarrow{a} \alpha \vdash \alpha\!:\!\bot}\ \Sigma\text{-Rule}\ \Big(0 \xrightarrow{a} \alpha \rhd \emptyset\Big)}{\emptyset \vdash 0\!:\![a]\,\bot}\ [a]$$

where the axiom is used to reveal the inconsistency in the assumption that the null process 0 could make an a-transition.

Definition 12. *Let $\mathcal{R}$ be a derivation rule. We say that $\mathcal{R}$ is* sound *if, whenever the sequent $\Sigma \vdash \Delta$ is derived using $\mathcal{R}$ from the premises $\{\Sigma_i \vdash \Delta_i\}_{i\in I}$ (for some finite index set I) which are all valid, then also $\Sigma \vdash \Delta$ is valid. We also say that $\mathcal{R}$ is* strongly sound *if, for every (M, γ), the following inequality holds*

$$[\![\Sigma \vdash \Delta]\!]^M_\gamma \geq \min\left\{[\![\Sigma_i \vdash \Delta_i]\!]^M_{\gamma'}\right\}_{i\in I}$$

for all interpretations γ' that agree with γ on all variables appearing in $\Sigma \vdash \Delta$.

The notion of strong soundness clearly implies the ordinary one. The proposition bellow explains the reason for omitting the contraction rule.

Proposition 13. *The contraction rule* $\dfrac{\Sigma \vdash \Delta, \phi, \phi}{\Sigma \vdash \Delta, \phi}$ *is not sound.*

Proposition 14. *The CUT rule is sound but not strongly sound. All other derivation rules of Figure 2 are strongly sound.*

Proof. Most cases are trivial to verify. The strong soundness of the rules $+_\lambda$ and $\{\rightsquigarrow\}$ follows from Proposition 15 below. □

Remark 1. Strong soundness of derivation rules is a technical requirement needed in the proof of our main theorem (see remarks after Theorem 17 below). As stated in Proposition 14, the CUT rule is not strongly sound. This fact requires restrictions to be placed on applications of CUT in proofs (see Definition 16 below). These restrictions are needed for our soundness proof to go through.

The rules $\{\text{P-Sub}, \text{D-Sub}, \Sigma\text{-Sub}\}$ are called *substitution rules* and support parametric reasoning [19]. In particular note how using the rule Σ-Sub one can substitute some of the occurrences of a compound distribution term with another equivalent (in all models satisfying Σ) compound distribution term. For example, the term $d +_{\frac{1}{3}} e$ can be rewritten to $(d +_{\frac{2}{3}} e) +_{\frac{1}{2}} e$. Such equational reasoning on distribution terms can be very useful (see, e.g., Remark 2 below). The rules $\{\vee_1, \vee_2, \wedge, \langle a\rangle, [a], \mu, \nu\}$ are called *logical rules*. The rules $\vee_1$, $\vee_2$, $\wedge$, μ, ν are standard and also the rules $\langle a\rangle$, $[a]$ for reasoning about modalities are natural counterparts to the analogous rules adopted in proof systems for modal (fixed point) logics appeared in the literature (see, e.g., [19], [21] and [15]). The rules $\{\delta, +_\lambda, \{\rightsquigarrow\}\}$ are called *distribution rules* and constitute a crucial aspect of the system. Together with the rule Σ-Sub, these are the only rules that operate on logical assertions containing distribution terms. All distribution rules can be understood probabilistically as (partially) evaluating the probability distribution terms of the active logical assertions. The simplest rule δ just evaluates a Dirac distribution to the corresponding process. In the rule $+_\lambda$, each active logical assertions (of the form $d_i +_\lambda e_i : F_i$ for a fixed $\lambda \in [0, 1]$) is evaluated to the left (resp. right) sub-term in the left (resp. right) premise of the rule with probability λ (resp. $1 - \lambda$). Note that the evaluation steps of each active probability distribution are not independent of each other. To the contrary, useful dependencies can be established by applications of the rule $+_\lambda$. The rule $\{\rightsquigarrow\}$ can be understood, by similar arguments, as a symbolic variant of the rule $+_\lambda$.

Remark 2. Note that the distribution rules $[+_\lambda]$ and $[\{\rightsquigarrow\}]$ may only be applicable once the distribution terms have been rewritten. Consider for example the sequent $\Sigma \vdash (d +_{\frac{1}{3}} e) : F, (d' +_{\frac{1}{2}} e) : G$. The distribution rule $\{+_\lambda\}$ is not directly applicable since the two distribution terms have a different outermost connective. However, by application of the rule Σ-Sub, the distribution term $d +_{\frac{1}{3}} e$ can be rewritten as $(d +_{\frac{2}{3}} e) +_{\frac{1}{2}} e$. This could be used as follows

$$\dfrac{\dfrac{\Sigma \vdash (d +_{\frac{2}{3}} e) : F, d' : G \qquad \Sigma \vdash e : F, e : G}{\Sigma \vdash \big((d +_{\frac{2}{3}} e) +_{\frac{1}{2}} e\big) : F, (d' +_{\frac{1}{2}} e) : G}\, +_{\frac{1}{2}}}{\Sigma \vdash (d +_{\frac{1}{3}} e) : F, (d' +_{\frac{1}{2}} e) : G}\, \Sigma\text{-Sub}$$

to reduce the original problem to the verification of the two new subgoals.

Proposition 15. *Let $\Sigma \vdash \Delta$ be derived by application of the rule $+_\lambda$ from the two premises $\Sigma \vdash \Delta_1$ and $\Sigma \vdash \Delta_2$. Then, for every (M, γ) it holds that*

$$[\![\Sigma \vdash \Delta]\!]^M_\gamma \geq \lambda \cdot [\![\Sigma \vdash \Delta_1]\!]^M_\gamma \;+\; (1-\lambda) \cdot [\![\Sigma \vdash \Delta_2]\!]^M_\gamma .$$

Similarly, let $\Sigma \vdash \Delta$ be derived by application of the rule $\{\rightsquigarrow\}$ and let $\Sigma \vdash \Sigma_1$ be its only premise, as depicted in Figure 2. Then, for every (M, γ), it holds that

$$[\![\Sigma \vdash \Delta]\!]^M_\gamma \geq \sum_{m \in M} \gamma(d)(m) \cdot [\![\Sigma \vdash \Delta_1]\!]^M_{\gamma[m/y]}$$

where $\gamma[m/y]$ updates γ by assigning to the fresh variable y the process $m \in M$.

Proof. Both points follow easily from the following arithmetical inequality (and variants thereof): $\oplus_{i \in I}\{x_i +_\lambda y_i\} \geq (\oplus\{x_i\}_{i \in I}) +_\lambda (\oplus\{y_i\}_{i \in I})$ which is valid for every index set I, reals $x_i, y_i \in [0,1]$, where $x +_\lambda y = \lambda \cdot x + (1-\lambda) \cdot y$. □

4.1 Markov Proofs

As anticipated in the introduction, to enable the proof system to handle the fixed points in the logic pLμ, we allow *cyclic* proof trees (cf. [15,21,3]) in which some leaves of the tree are identified with sequents internal to the tree, with the proof looping back to that point. Technically, it is convenient to view such cyclic trees as the infinite trees they unfold to, and to work with general infinite trees, with the finite cyclic ones corresponding exactly to the *regular* trees (those with only finitely many subtrees). We call a (possibly infinite) tree of rule applications, in which all leaves are instances of the axiom rule Σ-Axiom, a *preproof*. A preproof is *cut-free* if it does not contain occurrences of the CUT rule. Since they may have infinite branches, preproofs are not guaranteed to have valid endsequents even though every rule is (strongly) sound.

In the literature on infinitary proof systems for fixed-point logics (see, e.g., [15,21,3]), valid proofs are defined as those preproofs whose infinite branches all contain at least one legitimate sequence (called a *valid trace*) of fixed-point unfoldings, along which a greatest fixed-point is unfolded infinitely often. This can equivalently be reformulated in terms of a single player game. The aim of

the single player, Refuter, is to find an infinite branch along which all traces are invalid. The preproof is then considered a valid proof just in case Refuter cannot win his game.

For the proof system in this paper, we adopt a similar approach, except that we now interpret Refuter's game as a single-player stochastic game $\mathcal{G}(T)$ (i.e., a Markov Decision Process) over the preproof T, and we also need to constrain applications of the CUT rule (see Remark 1 and those following Theorem 17). Once again, in the game $\mathcal{G}(T)$, Refuter is trying to find an infinite branch in T along which all traces are invalid. This time, however, instances of the rule $+_\lambda$ in T are interpreted as probabilistic nodes under the choice of Nature, who extends the branch thus far with the left (resp. right) premise with probability λ (resp. $1-\lambda$). At all other rules, Refuter has the choice of premise. A preproof satisfies the *game condition* just in case Refuter almost surely fails in his goal; that is, no matter what strategy Refuter adopts, the probability of him finding an infinite branch with all traces invalid is 0. We now specify the collection of valid derivations, which we call *Markov proofs*, by the following *inductive* definition.

Definition 16. *A* Markov proof *is a preproof T satisfying the game condition and such that, for that every occurrence of the* CUT *rule in T,*

$$\frac{\begin{array}{cc} T_1 & T_2 \\ \Sigma \vdash \Delta, \phi & \Sigma \vdash \Gamma, \neg\phi \end{array}}{\Sigma \vdash \Delta, \Gamma}\, CUT$$

either the sub-preproof T_1 or T_2 (or both) is a Markov proof.

Note that a Markov proof can contain infinite branches on which Refuter wins as long as the set of such branches has probability 0 for every Refuter strategy in $\mathcal{G}(T)$. We shall see an example of this kind of Markov proof in Section 5. We remark also that the inductive definition could be replaced with a combinatorial condition. A Markof proof could equivalently be defined as a preproof T satisfying the game condition, for which there exists an assignment of a privileged premise (a 'switching') to every CUT rule such that no infinite path in the proof runs through infinitely many privileged ('switched') CUT premises.

The set of rules of Figure 2 has been kept as small as possible to simplify the proof of Theorem 17 below. Other expected rules are admissible, such as:

$$\frac{}{\Sigma \vdash \Delta, x:\top}\,\mathrm{Ax}(\top) \qquad \frac{}{\Sigma \vdash \Delta, x:F, x:\neg F}\,\mathrm{Ax}(\neg) \qquad \frac{\Sigma \vdash \Delta}{\Sigma, \Sigma' \vdash \Delta, \Gamma}\,\mathrm{Weak}$$

The following result is the main technical contribution of this paper.

Theorem 17 (Soundness). *The endsequent of every Markov proof is valid.*

Proof Sketch. Our proof technique is based on the game semantics of pLμ and, therefore, crucially exploits the equivalence result of [14]. The result is first proved for cut-free Markov proofs and then extended to general Markov proofs. The structure of a Markov proof Π with endsequent $\Sigma \vdash \{\phi_i\}_{i\in I}$ is seen as providing strategies σ_1^i for player Maximizer in the two-player stochastic games

associated with the assertions ϕ_i. On the other hand, a Markov play (i.e., a Markov chain) P_Π in $\mathcal{G}(\Pi)$, resolving the choices corresponding to the occurrences of rules $\{\wedge, \Sigma\text{-Rule}\}$ in Π, is seen as providing strategies σ_2^i for Minimizer as well as information about a counter-model M. This allows us to consider P_Π as a *coupling* of Markov chains, i.e., as a non-independent product of the probabilistic pLμ plays $P^i_{\sigma_1^i,\sigma_2^i}$ associated with ϕ_i, whose probabilistic dependencies have been introduced by the rules $[+_\lambda, \{\rightsquigarrow\}]$ in Π. Our proof is by *reductio ad absurdum*. One assumes that M and σ_2^i's constitute a counterexample to the validity of the endsequent of Π, i.e., that the expected probability of victory for Maximizer in the Markov plays $P^i_{\sigma_1^i,\sigma_2^i}$ sum up to a value $\lambda < 1$. We show that this implies that P_Π must assign at least probability $1 - \lambda$ (i.e., positive measure) to the set of branches in Π corresponding to plays losing for Maximizer in the pLμ game associated with ϕ_i, for all $i \in I$. These are precisely branches without valid traces. From these assumption it follows that Refuter can win in the game $\mathcal{G}(\Pi)$ with positive probability. Thus Π cannot be a Markov proof, a contradiction. □

The following theorem shows that regular (cyclic) Markov proofs do indeed form an effective proof system. This is essential for the potential applicability of the approach. It is proved along the lines of similar results for non-probabilistic cyclic proofs (see, e.g., [15], [21] and [3]), using decidability results for one-player stochastic parity games established in [5].

Theorem 18. *It is decidable if a regular preproof T is a Markov proof.*

5 Examples of Markov proofs

In this section we provide Markov proofs of the sequents Seq_3 and Seq_4 discussed in Example 9. Despite the simplicity of the process algebra considered in this paper, these small examples illustrate nontrivial instances of compositional reasoning and verification of infinite state systems. However, due to the space limits, some important features of our proof system, such as the possibility of recombining distribution terms (see Remark 2) and the capability of handling more realistic process algebras (such as those including, e.g, a *communicating* parallel operator), are not illustrated in this paper.

The validity of Seq_3 is proved by the (cut-free) Markov proof Π_3 depicted in Figure 3 (top), where the proviso of Σ-Rule, expressing a case analysis, is as in Example 11. The left sub-Markov proof Π_L, itself containing Π_3 as sub-Markov proof, is depicted as in Figure 3, and Π_R is the similar Markov proof of the sequent $y \xrightarrow{a} \beta, \alpha \simeq \{\beta \rightsquigarrow y'\}\delta(x|y') \vdash \alpha : F, x : \neg F, y : \neg F$. Each infinite play (i.e., branch in Π_3 since there are no probabilistic vertices in $\mathcal{G}(\Pi_3)$, see Section 4), has a valid trace because the greatest fixed point operators are unfolded infinitely many times. Thus Π_3 is a Markov proof as desired.

Compositional reasoning is supported in our system by the CUT rule (cf. [19]). For instance, as we have established the validity of Seq_3, we can reduce the problem of verifying the validity of the sequent $\emptyset \vdash p|q : F$ (i.e., prove that

$$\dfrac{\dfrac{\dfrac{\Pi_L \qquad \Pi_R}{x|y \xrightarrow{a} \alpha \vdash \alpha : F, x : \neg F, y : \neg F}\,\Sigma\text{-Rule: } x|y \xrightarrow{a} \alpha \rhd \{\Sigma_{\mathcal{L}}, \Sigma_{\mathcal{R}}\}}{\emptyset \vdash x|y : [a]\, F, x : \neg F, y : \neg F}\,[a]}{\emptyset \vdash x|y : \mu Z.\, [a]\, Z, x : \nu X.\langle a\rangle X, y : \nu Y.\langle a\rangle Y}\,\mu$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_3}{\emptyset \vdash x|y : F, x : \neg F, y : \neg F}}{\emptyset \vdash x'|y : F, x' : \neg F, y : \neg F}\,\text{P-SUB: } [x/x']}{\emptyset \vdash \delta(x'|y) : F, \delta(x') : \neg F, y : \neg F}\,\delta, \delta}{\emptyset \vdash \{\beta \rightsquigarrow x'\}\delta(x'|y) : F, \{\beta \rightsquigarrow x'\}\delta(x') : \neg F, y : \neg F}\,\{\rightsquigarrow\}}{\emptyset \vdash \{\beta \rightsquigarrow x'\}\delta(x'|y) : F, \beta : \neg F, y : \neg F}\,\Sigma\text{-Sub}}{x \xrightarrow{a} \beta \vdash \{\beta \rightsquigarrow x'\}\delta(x'|y) : F, x : \langle a\rangle \neg F, y : \neg F}\,\langle a\rangle\text{: } x \xrightarrow{a} \beta \rhd x \xrightarrow{a} \beta}{x \xrightarrow{a} \beta \vdash \{\beta \rightsquigarrow x'\}\delta(x'|y) : F, x : \neg F, y : \neg F}\,\nu}{x \xrightarrow{a} \beta, \alpha \simeq \{\beta \rightsquigarrow x'\}\delta(x'|y) \vdash \alpha : F, x : \neg F, y : \neg F}\,\Sigma\text{-Sub}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi_3}{\emptyset \vdash x : \neg F, y : \neg F, x|y : F}}{\emptyset \vdash p : \neg F, q : \neg F, p|q : F}\,\text{P-SUB} \qquad \emptyset \vdash p : F}{\emptyset \vdash p : \neg F, p|q : F}\,\text{CUT} \qquad \emptyset \vdash q : F}{\emptyset \vdash p|q : F}\,\text{CUT}$$

$$\dfrac{\dfrac{\dfrac{\Pi_3}{\emptyset \vdash x' | z : \neg F, x' : \neg F, z : \neg F}}{\emptyset \vdash (x|y)|z : F, x|y : \neg F, z : \neg F}\,\text{P-SUB } [(x|y)/x'] \qquad \dfrac{\Pi_3}{\emptyset \vdash x|y : F, x : \neg F, y : \neg F}}{\emptyset \vdash (x|y)|z : F, x : \neg F, y : \neg F, z : \neg F}\,\text{CUT}$$

$$\dfrac{\dfrac{\dfrac{\Pi_A}{p \xrightarrow{a} \beta \vdash \beta : F} \qquad \dfrac{\dfrac{\Pi_4}{\emptyset \vdash !^{\frac{1}{2}} p : F} \qquad \dfrac{\Pi_B}{p \xrightarrow{a} \beta \vdash \{\beta \rightsquigarrow y\}\delta(y|!^{\frac{1}{2}} p) : F, !^{\frac{1}{2}} p : \neg F}\,\text{CUT}}{p \xrightarrow{a} \beta \vdash \{\beta \rightsquigarrow y\}\delta(y|^{\frac{1}{2}} p\} : F}\,+_{\frac{1}{2}}}{p \xrightarrow{a} \beta \vdash \beta +_{\frac{1}{2}} \{\beta \rightsquigarrow y\}\delta(y|!^{\frac{1}{2}} p\} : F}\,\Sigma\text{-Sub}}{p \xrightarrow{a} \beta, \alpha \simeq \beta +_{\frac{1}{2}} \{\beta \rightsquigarrow y\}\delta(y|!^{\frac{1}{2}} p\} \vdash \alpha : F}\,\Sigma\text{-Rule: } P}{!^{\frac{1}{2}} p \xrightarrow{a} \alpha \vdash \alpha : F}\,[a]}{\emptyset \vdash !^{\frac{1}{2}} p : [a]\, F}\,\mu}{\emptyset \vdash !^{\frac{1}{2}} p : \mu X.\, [a]\, X}$$

Fig. 3. Examples of Markov proofs.

the compound system $p|q$ almost surely terminates) to the verification of the two smaller goals $\emptyset \vdash p : F$ and $\emptyset \vdash q : F$ by means of the Markov proof depicted in Figure 3. Furthermore, other useful results can be proved, without searching for direct proofs, by using already proved lemmas. For example, the validity of the sequent $\emptyset \vdash (x|y)|z : F, x : \neg F, y : \neg F, z : \neg F$ can be proved as in Figure 3.

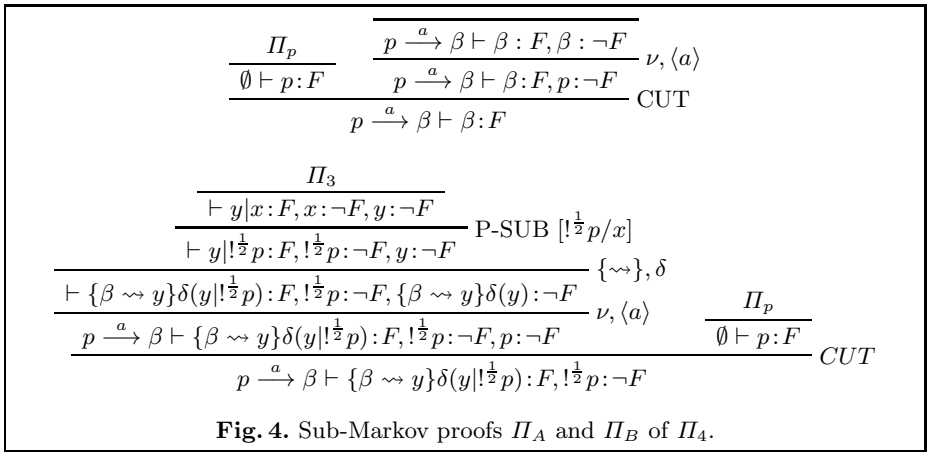

Fig. 4. Sub-Markov proofs Π_A and Π_B of Π_4.

Compositional reasoning is the key to the verification of infinite state systems. Consider, for example, a process p that almost surely terminates, i.e., such that the validity of $S_p = \emptyset \vdash p : F$ has been proven by a Markov proof Π_p. It is simple to verify that $!^{\frac{1}{2}}p$ is an infinite state system, even when p is a finite state process such as $a.0$. Nevertheless, it is possible to prove that $!^{\frac{1}{2}}p$ almost surely terminates. This is expressed (for $p{=}a.0$) by the sequent Seq_4 whose validity is witnessed by the regular Markov proof Π_4 depicted in Figure 3 (bottom) where the operational judgment $P \stackrel{\mathrm{def}}{=} !^{\frac{1}{2}}p \xrightarrow{a} \alpha \rhd \left\{ \{p \xrightarrow{a} \beta, \alpha \simeq \beta +_{\frac{1}{2}} \{\beta \rightsquigarrow y\}\delta(y|!^{\frac{1}{2}}p)\} \right\}$ used in the rule Σ-Rule is defined is valid, and the Markov proofs Π_A and Π_B can be depicted as in Figure 4. Note how the use of the CUT rule in Π_B allows us to make use of the result already proved by the Markov proof Π_3. This time Π_4 contains probabilistic vertices: at occurrences of the rule $+_{\frac{1}{2}}$ the game probabilistically branches. The only infinite branch in Π_4 without valid traces is the one never joining one of the the two sub-Markov proofs Π_A or Π_B. However, the probability that this branch is the outcome of the game $\mathcal{G}(\Pi_4)$ is easily seen to be an event of probability 0. Thus Π_4 satisfies the proof condition and is a Markov proof, as desired.

6 Further directions

There are numerous directions for improvement to the approach of this paper. One is relax the restrictions on applications of CUT. Is it possible to reconfigure the proof system and soundness proof so that an unrestricted CUT rule is available? Another is to address completeness issues, which we have ignored entirely. Are completeness results available for restricted classes of processes (e.g., finite state)? Yet another is to attempt to extend the proof system to deal with extensions of the probabilistic μ-calculus with other operators, for example those considered in [13], allowing the full expressivity of PCTL to be captured.

It is unclear to us whether or not the approach of this paper is able to scale up to establish useful properties of real-world systems. Nevertheless, we see the main value of our paper as contributing novel techniques towards the challenging problem of compositional verification for concurrent probabilistic systems. In particular, we believe that the use of proofs containing probabilistic branching will generalise to other proof systems for other probabilistic logics.

Acknowledgements We thank the anonymous referees for helpful suggestions.

References

1. C. Baier and J. P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.
2. F. Bartels. GSOS for probabilistic transition systems. In *Electronic Notes in Theoretical Computer Science, Volume 65, Issue 1*, 2002.
3. J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.
4. L. Cardelli, K. Larsen, and M. Radu. *Modular Markovian Logic*, volume 6756 of *Lecture Notes in Computer Science*, pages 380–391. Springer Berlin, 2011.
5. K. Chatterjee. *Stochastic ω-Regular Games*. PhD thesis, University of California, Berkeley, 2007.
6. B. Delahaye, J. P. Katoen, K. Larsen, A. Legay, M. Pedersen, F. Sher, and A. Wasowski. Abstract probabilistic automata. In *Proc. of 12th VMCAI*, 2011.
7. V. Forejt, M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Quantitative multi-objective verification for probabilistic systems. In *Proc. of 14th TACAS*, 2011.
8. P. Hájek. *Metamathematics of Fuzzy Logic*. Trends in Logic. Springer, 2001.
9. T. A. Henzinger and J. Sifakis. The embedded systems design challenge. In *Proc. 14th International Symposium on Formal Methods (FM)*, 2006.
10. M. Huth and M. Kwiatkowska. Quantitative analysis and model checking. In *Proc. of 12th LICS*, 1997.
11. M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Assume-guarantee verification for probabilistic systems. In *Proc. of 16th TACAS*, 2010.
12. A. McIver and C. Morgan. Results on the quantitative μ-calculus qMμ. *ACM Transactions on Computational Logic*, 8(1), 2007.
13. M. Mio. *Game Semantics for Probabilistic μ-Calculi*. PhD thesis, School of Informatics, University of Edinburgh, 2012.
14. M. Mio. On the equivalence of denotational and game semantics for the probabilistic μ-calculus. *Logical Methods in Computer Science*, 8(2), 2012.
15. D. Niwinski and I. Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science*, 163:99–116, 1997.
16. P. Panangaden. *Labelled Markov processes*. Imperial College Press, 2009.
17. A. Pnueli. The temporal logic of programs. In *Proc. of 19th FOCS*, 1977.
18. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Laboratory for Computer Science, M.I.T., 1995.
19. A. Simpson. Sequent calculi for process verification: Hennessy-Milner logic for an arbitrary GSOS. *Journal of Logic and Algebraic Programming*, 60-61:287, 2004.
20. C. Stirling. *Modal and temporal logics for processes*. Springer, 2001.
21. T. Studer. On the proof theory of the modal mu-calculus. In *Studia Logica, Volume 89, Number 3*. Springer Netherlands, 2007.