Probabilistic Concurrent Kleene Algebra

Annabelle McIver Tahiry Rabehaja
Department of Computing [*]
Macquarie University
Sydney, Australia
{annabelle.mciver,tahiry.rabehaja}@mq.edu.au

Georg Struth
Department of computer Science
University of Sheffield
United Kingdom
g.struth@dcs.shef.ac.uk

We provide an extension of concurrent Kleene algebras to account for probabilistic properties. The algebra yields a unified framework containing nondeterminism, concurrency and probability and is sound with respect to the set of probabilistic automata modulo probabilistic simulation. We use the resulting algebra to generalise the algebraic formulation of a variant of Jones' rely/guarantee calculus.

1 Introduction

Since Rabin's seminal paper introducing probabilistic algorithms [18], the role of probability in the design of systems has increased in popularity and effectiveness. However, probability and concurrency results in greater complexity and so the formal verification of such system poses a particular challenge. Many process algebras and automated tools have been developed to combine probability and concurrency but there are some important features of concurrent systems such as interference between shared variables that are not easily captured. Interference is for example a feature that appears in Jones' rely/guarantee calculus. In this paper, we generalise concurrent Kleene algebra and extend the variant of Jones' rely/guarantee rules found in [5] to apply to systems with probability.

Introduced by Kleene in his study of automata and regular languages [7], Kleene algebras provide an elegant tool that is able to express static properties of standard sequential programs [10–12]. The axiomatisation of Kleene algebras and their extensions have been well studied by Conway [1], Kozen [9], Salomaa [19] and others [4, 17].

Concurrent Kleene algebra has been developed and used to give robust proofs of concurrent systems, and in particular for verification techniques such as a variant of Jones' rely/guarantee rules [5,6]. Algebra is an important mathematical tool for carrying out complex proofs, however its effectiveness relies on realistic models. In this paper, we give the first Kleene like structure that extends concurrent Kleene algebra with probabilistic behaviour.

Our algebra is constructed from concurrent and probabilistic Kleene algebras [5, 16]. It provides an abstraction of concurrent systems and is able to express properties such as interference. The most important rule of concurrent Kleene algebra is the interchange law which establishes the interaction between concurrency and sequential execution. Informally, it says that the sequential execution of two concurrent systems entails an explicit dependency between the two concurrent parts, therefore no interference. But the concurrent execution of two sequential systems allows interference between any parts of the two systems. The relationship between these behaviours is expressed within a partial order determined by traces, and the algebraic rules mandate program operators' behaviours when used in combination.

In contrast, probabilistic Kleene algebra provides an abstraction of probability whose existence can be deduced from a subdistributivity law. The law describes the interaction between sequential execution

[*] This research has been supported by the Australia Research Council Discovery Grant DP1092464 and the iMQRS Grant from Macquarie University.

L. Bortolussi and H. Wiklicky (Eds.): QAPL 2013
EPTCS 117, 2013, pp. 97–115, doi:10.4204/EPTCS.117.7

and nondeterminism in the presence of probability. Informally, it says that a nondeterministic choice preceded by a probabilistic action can be resolved using the outcome of the probabilistic choice, that is,

$$x \cdot y + x \cdot z \leq x \cdot (y + z)$$

where x, y, z are probabilistic programs, $+$ is the nondeterministic choice and $\cdot$ denotes sequential execution. This law also ensure monotonicity of the sequential composition which is an important property for combining sequential system from smaller components.

Our first contribution is to expand the set of axioms for concurrent Kleene algebra to account for the presence of probability and prove their soundness with respect to an automata model of probabilistic concurrency modulo simulation equivalence. In proving the soundness we find that some of the original axioms, both from concurrent and probabilistic Kleene algebras, needed to be weakened to accommodate the presence of both features in a single framework. Our simulation is based on the definition of Deng et al. [2] though we show that it is equivalent to Segala's definition of probabilistic weak forward simulation [21]. Segala's simulation completely characterises the coarsest precongruence included in the trace distribution equivalence of probabilistic automata [14] and coincides with the infinitary probabilistic vector may testing order [22].

Our second contribution is the extension of the rely/guarantee calculus of [5] to concurrent systems exhibiting probabilistic behaviours. We show that with the axiomatisation presented in Section 4, the algebra is sufficient to prove some important rely/guarantee rules which hold in the action-based interleaving model as well as any other semantics that satisfy the necessary algebraic properties. Since the inequality in the rules can be interpreted as the existence of a probabilistic simulation, the testing interpretation of simulation [3] allows us to provide bounds for the maximal probability of failure.

In Section 2, we provide a summary of probabilistic automata and the operators which will be used to interpret our algebraic terms. Section 3 contains a survey of probabilistic simulation where the formulation of Deng et al is shown to be equivalent to Segala's definition of probabilistic weak forward simulation. Section 4 provides the soundness of the algebra with respect to automata and probabilistic simulation. In particular, we show how the probability interacts with the sequential as well as concurrent operators. We also provide the derivation of some rely/guarantee rules within and study a simple example to show the application of the algebraic reasoning developed.

2 Probabilistic Automata and Operations

The standard constructions of automata theory have been generalised to capture probabilistic behaviour which we summarise briefly here. We will use these automata as a model to exhibit soundness for the generalised concurrent Kleena Algebra we describe in Section 4.

Probability is encoded in terms of distributions over the state space. A transition in a probabilistic automaton starts from a source state, executes an action from a given alphabet Σ and ends in a target distribution [21]. Such a distribution is then resolved into a probabilistic choice which specifies the new state of the automaton up to some probabilistic factor.

Definition 2.1. *A probabilistic automaton is a tuple $(P, \Sigma, \longrightarrow, \phi_P, F_P)$ where*

- *P is a set of states,*
- *Σ is a set of actions,*
- *$\longrightarrow : P \times \Sigma \times \mathscr{D}(P)$ is a set of probabilistic transitions where $\mathscr{D}(P)$ is the set of finitely (or countably) supported probability distributions over P,*

- *ϕ_P is the start or initial distribution over the states in P,*
- *and F_P is a set of final states.*

We usually identify an automoton with its set of states and explicit distinction will be made only when confusion may arise.

Example 2.2. *Figure 1 depicts two probabilistic automata. The automaton on the left models a faulty vending machine that becomes stuck with probability* 0.2*; on the right the automaton represents the actions of a user interacting with the automaton by kicking it if he fails to get his tea. Two kicks means the machine really is broken.* [1]

The states of the two automata are labelled by s_i, t_i respectively and distributions are not labelled unless they are initial and their components correspond to dotted arrows labelled with the probability. The set of actions is $\Sigma = \{\texttt{coin}, \texttt{tea}, \texttt{kick}, \texttt{fail}, \texttt{stuck}\}$ where `stuck` *is the only internal action. We assume that the two automata have no final state to facilitate the upcoming calculations.*

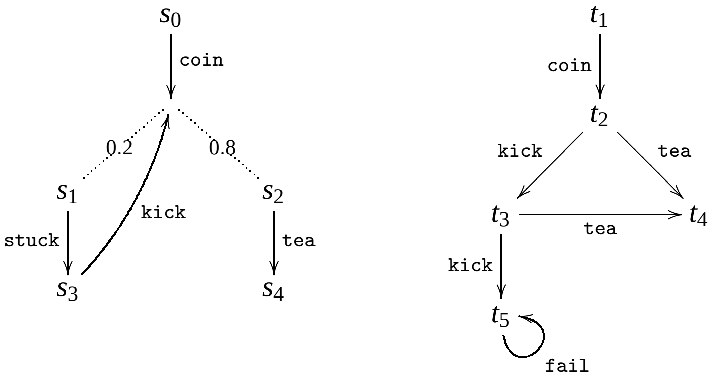

Figure 1: A probabilistic vending machine $V = \texttt{coin} \cdot M$ and a user $U = \texttt{coin} \cdot U'$ who kicks the machine once if it gets stuck.

Definition 2.1 provides a specialised version of probabilistic automata. Generally, a transition is composed of a state and a distribution over $\Sigma \times P$ but restriction to automata with simple transitions suffices for our results in this paper. We denote by **PAut** the set of such probabilistic automata.

The set of actions Σ is divided into two parts, namely, *internal* and *external* actions. Internal actions are either local or invisible and are usually intrinsic to the automaton where they are defined. They are not shared with other automata in the sense that they can be executed independently from the environment. A special case is the silent action τ which does not belong to the set of internal actions I and we write $I_\tau = I \cup \{\tau\}$. In contrast, external actions are visible to the environment and may be synchronised. We denote the set of external actions by E, and define $\Sigma = I \cup E$ and $\Sigma_\tau = I_\tau \cup E$. The set Σ is assumed implicitly and is fixed for every automaton.

The linear run of a probabilistic automaton yields a *path*, as in the standard case, which is quantified with respect to a family of probability measures (indexed with a set of probabilistic scheduler). Formally, a path is a sequence $x_0 a_1 x_1 a_2 x_2 \cdots$ of alternating states and actions such that there is a sequence of transitions $x_i \xrightarrow{a_{i+1}} \mu_{i+1}$, $i \geq 0$, where $x_i \in \text{supp}(\mu_i)$ (the support of μ) for every $i > 0$. A path α always

[1] This example was suggested by Steve Schneider [20].

starts with a state and ends with another state, denoted last(α), if it is finite. Usually, we want a path to start from a state in the support of the initial distribution. We denote Path(P) the set of all finite paths of an automaton P.

Next we provide some operations over probabilistic automata. The regular operators include nondeterministic choice $(+)$, sequential composition $(\cdot)$ and Kleene star $(*)$ which abstracts tail iteration together with the constants skip (**1**), deadlock (**0**) and the automaton that enables a single successfully terminating action. Such an automaton is denoted **a** where the action is a. Formally, we have

- Deadlock **0** corresponds to $(\{x\},\emptyset,\delta_x,\emptyset)$ where δ_x is the point probability distribution concentrated on the state x.
- Skip **1** corresponds to $(\{x\},\emptyset,\delta_x,\{x\})$.
- **a** corresponds to $(\{x,x'\},\{x \xrightarrow{a} \delta_{x'}\},\delta_x,\{x'\})$.

In the reminder of this section, we fix two probabilistic automata P,Q with respective initial distribution μ_0,ν_0, sets of final states F_P,F_Q and sets of transitions $\longrightarrow_P,\longrightarrow_Q$. We also assume that the state spaces of P and Q are disjoint. We now give an automata semantics for each of the named operators.

Nondeterminism is defined as in the standard case by constructing a new initial point distribution δ_z such that z leads to the respective initial distributions of the operands via τ transitions.

Definition 2.3. *Nondeterministic choice between P and Q is defined by:*

$$P+Q = (P\cup Q\cup\{z\},\longrightarrow_P\cup\longrightarrow_Q\cup\{z\xrightarrow{\tau}\mu_0, z\xrightarrow{\tau}\nu_0\},\delta_z,F_P\cup F_Q)$$

where $z\notin P\cup Q$.

Definition 2.4. *The sequential composition of P followed by Q is defined by:*

$$P\cdot Q = (P\cup Q,\longrightarrow_{P\cdot Q},\mu_0,F_Q)$$

where

$$\longrightarrow_{P\cdot Q}=\longrightarrow_P\cup\longrightarrow_Q\cup\{x\xrightarrow{\tau}\nu_0 \mid x\in F_P\}.$$

This definition is a straightforward generalisation of the standard definition from automata theory.

Now we define the Kleene star in the standard way.

Definition 2.5. *The tail iteration or Kleene star of P is* $P^* = (P\cup\{z\},\longrightarrow_{P^*},\delta_z,\{z\})$ *where*

$$\longrightarrow_{P^*}=\longrightarrow_P\cup\{z\xrightarrow{\tau}\mu_0, x\xrightarrow{\tau}\delta_z \mid x\in F_P\}$$

and $z\notin P$.

The implementation of a probabilistic choice between two automata is defined below.

Definition 2.6. *We define the probabilistic choice between P, with probability p, and Q, with probability* $1-p$*, as*

$$P\ {}_p\!\oplus\ Q = (P\cup Q,\longrightarrow_P\cup\longrightarrow_Q,p\mu_0+(1-p)\nu_0,F_P\cup F_Q).$$

Finally, the parallel composition is defined using a probabilistic version of CSP parallel composition operation that synchronises on the actions in $A\subseteq E$. The frame set A is assumed to be fixed throughout this paper.

Firstly, given $\mu\in\mathscr{D}(P)$ and $\nu\in\mathscr{D}(Q)$, the product $\mu\times\nu$ is a distribution over $P\times Q$ such that $(\mu\times\nu)(x,y)=\mu(x)\nu(y)$ (component-wise multiplication).

Definition 2.7. *We define the parallel composition of P and Q as*

$$P_A \| Q = (P \times Q, \longrightarrow_{P_A \| Q}, \mu_0 \times \nu_0, F_P \times F_Q)$$

where, for each $a \in \Sigma_\tau$, a transition $(x,y) \xrightarrow{a} \mu \times \nu$ belongs to $\longrightarrow_{P\|Q}$ if one of the following conditions holds:

- *$a \in A$ and $x \xrightarrow{a} \mu$ and $y \xrightarrow{a} \nu$,*
- *$a \notin A$ and $x \xrightarrow{a} \mu$ and $\nu = \delta_y$,*
- *$a \notin A$ and $y \xrightarrow{a} \nu$ and $\mu = \delta_x$.*

In the construction of the transition relations of $P_A \| Q$, given a transition of P, if it is labelled by an action in A then it is blocked until it gets synchronised with a transition of Q labelled with the same action. Otherwise, that transition is interleaved with the transitions of Q.

Example 2.8. *Using this language we can describe the automata from Figure 1. The right hand side automaton of Figure 1 corresponds to the algebraic expression*

$$\texttt{coin} \cdot (\texttt{kick} \cdot (\texttt{kick} \cdot \texttt{fail}^* + \texttt{tea}) + \texttt{tea})$$

where we have abused notation by denoting the automaton that does a single action, say `coin`*, and then terminates successfully with the same notation* `coin`*.*

The left hand side is obtained as a sequential composition $\texttt{coin} \cdot M$ where M corresponds to the least fixed point of

$$f(X) = \texttt{stuck} \cdot \texttt{kick} \cdot X \cdot \mathbf{0} \ {}_{0.2}\oplus \ \texttt{tea} \cdot \mathbf{0}.$$

We will express the least fixed point of f as an algebraic expression in Section 4.

3 Probabilistic Simulation

In this section, we define an inequality on the set **PAut** as per the constructions of [2, 3, 14, 22]. The equivalence relation is based on weak simulation and we are mainly interested in the equational theory of this model and relate it to the axiomatisation of probabilistic and concurrent Kleene algebras.

We provide two equivalent definitions of simulation. The first definition is the probabilistic simulation of [2] and the second is probabilistic weak forward simulation of [22]. Both definitions of simulation rely on the lifting of relations from states to distributions. Given a relation $S \subseteq X \times \mathscr{D}(Y)$, the lifting [2] of S is a relation $\overline{S} \subseteq \mathscr{D}(X) \times \mathscr{D}(Y)$ such that $\mu \overline{S} \nu$ if and only if there exists a family of real number $\{p_n \mid n \in N\} \subseteq [0,1]$ such that $\sum_n p_n = 1$ and

1. $\mu = \sum_{n \in N} p_n \delta_{x_n}$,
2. for each $n \in N$, there exists $\nu_n \in \mathscr{D}(X)$ such that $x_n S \nu_n$,
3. $\nu = \sum_{n \in N} p_n \nu_n$.

The lifting is a probability preserving function that associates to each probabilistic relation R a standard relation $\overline{R}$ over the set of distributions. It is important to notice that the decomposition of μ is not necessarily canonical that is, there may be some repetition in the x_is. Moreover, the lifting also applies to labelled transition because $\cdot \xrightarrow{a} \cdot \subseteq P \times \mathscr{D}(P)$ for any probabilistic automaton P and any action $a \in \Sigma_\tau$. Hence, we denote $\xrightarrow{\overline{a}}$ the lifting of this transition which corresponds to the notion of combined transition of [14, 22].

Lastly, we extend internal transitions with reflexivity, that is, we write $x \xrightarrow{\tau} \mu$ if such a transition exists in the automaton or $\mu = \delta_x$. The lifted version is again denoted $\xrightarrow{\overline{\tau}} \subseteq \mathscr{D}(P) \times \mathscr{D}(P)$. Finally, weak transitions are obtained from the reflexive transitive closure of $\xrightarrow{\overline{\tau}}$, denoted $\Longrightarrow$, and we write $\mu \overset{a}{\Longrightarrow} \mu'$ if there exist μ_1, μ_2 such that $\mu \Longrightarrow \mu_1 \xrightarrow{\overline{a}} \mu_2 \Longrightarrow \mu'$.

We now give the formal definition of simulation by straightforwardly generalising [2] to automata with final states.

Definition 3.1. *A probabilistic simulation S from P to Q is a relation $S \subseteq P \times \mathscr{D}(Q)$ satisfying the following properties:*

1. *there exists ν_0' such that $\mu_0 \overline{S} \nu_0'$ and $\nu_0 \Longrightarrow \nu_0'$,*
2. *if $x \xrightarrow{a} \mu'$ is a valid transition of P and $xS\nu$, there exists $\nu' \in \mathscr{D}(Q)$ such that $\nu \overset{a}{\Longrightarrow} \nu'$ and $\mu' \overline{S} \nu'$,*
3. *if $x \in F_P$ and $xS\nu$ then there exists $\nu' \in \mathscr{D}(F_Q)$ such that $\nu \Longrightarrow \nu'$.*

Property (a) ensures that preceding τ actions do not interfere with probabilistic choice (i.e. $P\ {}_p\!\oplus\ Q$ and $\tau \cdot (P\ {}_p\!\oplus\ Q)$ are equal). Property (b) is the usual co-inductive definition of simulation and property (c) ensures that if a state $x \in P$ is simulated by a distribution $\nu \in \mathscr{D}(Q)$ and P can terminate successfully at x then Q can also terminate successfully from ν after a *finite number* of internal transitions.

A simulation is always total on reachable states, that is, if $S \subseteq P \times \mathscr{D}(Q)$ is a simulation and $x \in P$ such that $x_0 a_1 x_1 \cdots x$ is a path that occurs with positive maximal probability, then there exists $\nu \in \mathscr{D}(Q)$ such that $xS\nu$.

We write $P \leq Q$ if there is a simulation from P to Q and $P \equiv Q$ iff $P \leq Q$ and $Q \leq P$.

Example 3.2. *Figure 2 depicts two automata related by a simulation relation i.e. $M \leq H$ where M (resp. H) is the left (resp. right) automaton. The simulation is obtained from the relation $S = S' \cup \{(s_3, \mu) \mid (s_1, \mu) \in S'\}$ where*

$$S' = \{(s_1, 0.2\delta_{u_0} + 0.8\delta_{u_1}), (s_1, \delta_{u_2}), (s_1, \delta_{u_4}), (s_2, \delta_{u_1}), (s_2, \delta_{u_3}), (s_4, \delta_{u_5})\}.$$

In fact, we can write $\nu_0 = 0.2(0.2\delta_{u_0} + 0.8\delta_{u_1}) + 0.8\delta_{u_1}$ where $s_1 S(0.2\delta_{u_0} + 0.8\delta_{u_1})$ and $s_2 S\delta_{u_1}$. Hence, $\mu_0 \overline{S} \nu_0$. Since `stuck` *is an internal action, it follows $s_3 S\mu$ and $\mu \overset{\overline{\tau}}{\Longrightarrow} \mu$ for every distribution μ such that $s_1 S\mu$. Next, we have $s_3 S(0.2\delta_{u_0} + 0.8\delta_{u_1})$ and $s_3 \xrightarrow{\texttt{kick}} \mu_0$. Since $\mu_0 = 0.2\delta_{s_1} + 0.8\delta_{s_2}$ and $s_1 S\delta_{u_2}$ and $s_2 S\delta_{u_3}$, it follows that $\mu_0 \overline{S}(0.2\delta_{u_2} + 0.8\delta_{u_3})$ and $(0.2\delta_{u_0} + 0.8\delta_{u_1}) \overset{\overline{\texttt{kick}}}{\Longrightarrow} (0.2\delta_{u_2} + 0.8\delta_{u_3})$. The other inductive cases are proved in similar fashion. Moreover, an algebraic proof is given in the next section.*

Proposition 3.3. *The simulation relation is a preorder.*

Any probabilistic simulation satisfying Definition 3.1 will be referred simply as a simulation. In contrast, the definition of forward simulation [22] relies on a *double lifting*. Given a relation $S \subseteq X \times Y$, the double lifting of S, denoted $\overline{\overline{S}}$, is a subset of $\mathscr{D}(X) \times \mathscr{D}(Y)$ where $\mu \overline{\overline{S}} \nu$ iff there exists a function $w : X \times Y \to [0,1]$ such that

1. if $w(x,y) > 0$ then xSy,
2. for every $x \in X$, $\sum_{y \in Y} w(x,y) = \mu(x)$,
3. for every $y \in Y$, $\sum_{x \in X} w(x,y) = \nu(y)$.

The function w is again a probability preserving function that provides corresponding decompositions for μ and ν. Double lifting generates a distribution over the set of distributions which complicates the

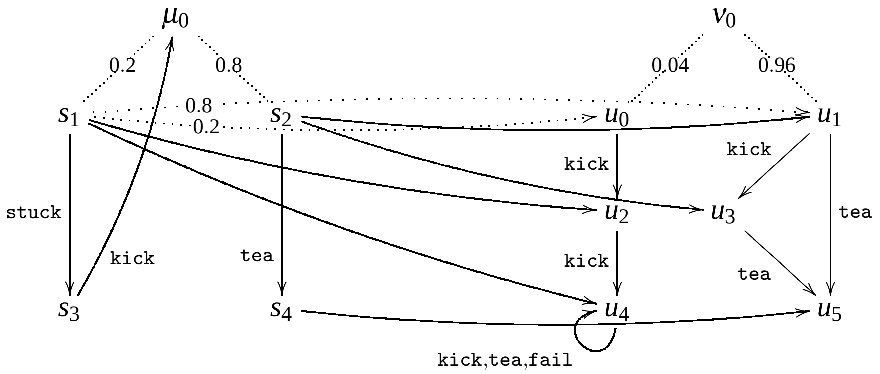

Figure 2: Two automata related by simulation: the left is M and the right is H. Remind that `stuck` is an internal action so we have removed the arrows from s_3 because they are exactly the same as for s_1. The dotted arrow represents non-trivial distribution again.

lifting of transitions. To obtain a standard relation over the set of distributions, Segala [14, 22] provided a flat version of a distribution in $\mathscr{D}(\mathscr{D}(Q))$ through the use of $\pi : \mathscr{D}(\mathscr{D}(X)) \to \mathscr{D}(X)$ such that

$$\pi(\phi) = \sum_{\mu \in \mathrm{supp}(\phi)} \phi(\mu)\mu.$$

We now give the modified version of Segala's probabilistic weak forward simulation.

Definition 3.4. *A relation $S \subseteq P \times \mathscr{D}(Q)$ is a probabilistic weak forward simulation if*

a) there exist $\psi_0 \in \mathscr{D}(\mathscr{D}(Q))$, such that $\mu_0 \overline{\overline{S}} \psi_0$ and $\nu_0 \Longrightarrow \pi(\psi_0)$,

b) if $x \xrightarrow{a} \mu'$ is a valid transition of P and $xS\nu$, there exists $\psi \in \mathscr{D}(\mathscr{D}(Q))$ such that $\nu \overset{a}{\Longrightarrow} \pi(\psi)$ and $\mu' \overline{\overline{S}} \psi$.

c) if $x \in F_P$ and $xS\nu$ then there exists $\psi \in \mathscr{D}(\mathscr{D}(F_Q))$ such that $\nu \Longrightarrow \pi(\psi)$.

Proposition 3.5. *Let X,Y be two sets, $S \subseteq X \times \mathscr{D}(Y)$, $\mu \in \mathscr{D}(X)$ and $\psi \in \mathscr{D}(\mathscr{D}(Y))$. If $\mu \overline{\overline{S}} \psi$ then $\mu \overline{S} \pi(\psi)$.*

Proof. If $\mu \overline{\overline{S}} \psi$, then there exists $w : X \times \mathscr{D}(Y) \to [0,1]$ satisfying the condition above. Then by considering $I = \mathrm{supp}(w)$, it directly follows that $\mu = \sum_{i \in I} w(i)\delta_{x_i}$, each x_i is related to some ν_i and $\pi(\psi) = \sum_{i \in I} w(i)\nu_i$. □

Corollary 3.6. *A relation is a probabilistic simulation iff it is a probabilistic weak forward simulation on* **PAut**.

Proof. We provide a sketch for the proof though the complete proof should not be hard to obtain from it.

That Definition 3.4 implies Definition 3.1 follows directly from the previous proposition.

Assume that $S \subseteq P \times \mathscr{D}(Q)$ satisfies Definition 3.1. If $\mu \overline{S} \nu$, then there exsits a decomposition $\mu = \sum_{i \in I} p_i \delta_{x_i}$ such that for each i, there exists $\nu_i \in \mathscr{D}(Q)$ such that $x_i \overline{S} \nu_i$ for each i, and $\nu = \sum_{i \in I} p_i \nu_i$. Hence $\mu \overline{\overline{S}} \sum_{i \in I} p_i \delta_{\nu_i}$. We just apply this simple construction for each of the three cases. □

Proposition 3.7. *Simulation is a precongruence i.e. if $P \leq Q$ then $P+R \leq Q+R$, $P \cdot R \leq Q \cdot R$, $P^* \leq Q^*$, $P \;{}_p\!\oplus R \leq Q \;{}_p\!\oplus R$, $P_A || R \leq Q_A || R$ and the same holds for binary operators when the order of the arguments is reversed.*

The proof of this proposition can be found in the Appendix Proposition A.1.

We conclude this section with a remark about the simulation of Definitions 3.1 and 3.4. In Proposition 3.6, we have shown that the corresponding definitions of [21] and [2] coincide (Notice that we can replace final states with some special external action and obtain a formulation closer to [2,22]). On the one hand, Segala has shown that the largest precongruence included in the trace distribution equivalence coincides with *vector may testing* where there are uncountably many success actions [22]. On the other hand, Deng et al. have shown that vector and scalar testings coincide on the recursion-free fragment of probabilistic automata and that with the same restriction, Definition 3.1 is complete for testing equivalence [3]. With the help of Proposition 3.6 and the equivalence between probabilistic weak forward simulation (Definition 3.4) and the coarsest precongruence included in the trace distribution, we conclude that Deng's completeness for may testing extends to automata with countable state spaces. However, it is still unknown whether the equivalence between scalar and vector testing in the infinite case is valid.

These equivalences are the main motivation for our use of simulation in order to create an interleaving model for our algebra. It should be noted that simulation equivalence is decidable for finite automata but it is unknown whether an efficient decision procedure exists. This is in contrast to other related results in the literature showing that strong simulation is decidable in polynomial time [8].

4 Probabilistic Concurrent Kleene Algebra

In this section, we introduce probabilistic concurrent Kleene algebra. We show that the set of probabilistic automata modulo probabilistic simulation as defined by Definitions 3.1 or 3.4 satisfies an extension of Kleene algebras that includes probability and concurrency.

Concurrent Kleene Algebra has four unary and binary operators, namely, $+, \cdot, ||$ and $*$. These operators have pomset operation semantics but the axiomatisation is too weak to allow the presence of probability. In contrast, Probabilistic Kleene Algebra has three operators, namely, $+, \cdot$ and $*$. We ensure that these operators coincide with the respective operators of Concurrent Kleene Algebra and provide a new probabilistic concurrent Kleene algebra that extends both structures. Without explicit probabilistic choice, such combination generates a weak Concurrent Kleene Algebra.

Definition 4.1. *A weak concurrent Kleene algebra is an algebraic structure with signature $(K, +, \cdot, ||, *, \mathbf{0}, \mathbf{1})$ where K is a set closed under the operations and satisfies Equations (1-4), (8-13) and (16-21)*

To gain complete control of probabilities, we append explicit probabilistic choices to weak concurrent Kleene algebra. Many of the following equations have been proven elsewhere so we will prove only those that are specific to our algebra. We concentrate on equations describing the interactions between probabilistic choices, sequential composition, Kleene star and the exchange law (20).

We assume the following precedence between the operators. The Kleene star $*$ binds more tightly than $\cdot$ which binds more tightly than $||$. The operator $||$ binds more tightly than $+$ and ${}_p\!\oplus$ and we use parenthesis to parse expressions having $+$ and ${}_p\!\oplus$ at the same level.

The following equations are standard and the proofs are omitted (they can be found in [2]).

$$P \quad \equiv \quad P+P \tag{1}$$

$$\begin{array}{rcll}
P & \equiv & P+\mathbf{0} & (2)\\
P+Q & \equiv & Q+P & (3)\\
P+(Q+R) & \equiv & (P+Q)+R & (4)\\
P & \equiv & P\ {}_{p}{\oplus}\ P & (5)\\
P\ {}_{p}{\oplus}\ Q & \equiv & Q\ {}_{1-p}{\oplus}\ P & (6)\\
P\ {}_{p}{\oplus}\ (Q\ {}_{q}{\oplus}\ R) & \equiv & (P\ {}_{p'}{\oplus}\ Q)\ {}_{q'}{\oplus}\ R & (7)
\end{array}$$

where $p'q' = p$, $(1-p')q' = (1-p)q$ and $1-q' = (1-p)(1-q)$. Moreover, the equivalence $P \leq Q$ iff $P+Q \equiv Q$ follows from these equations, that is, simulation coincides with the natural order of the algebra. Remind that in our interpretation Q has more behaviours than P. A complete characterisation of the consequences of Equation (5-7) with respect to probabilistic bisimulation can be found in [24].

The proof of the following propositions can be found in the Appendix under Proposition A.2, A.3 and A.4 respectively.

Proposition 4.2. *The sequential composition satisfies*

$$\begin{array}{rcll}
P & \equiv & P\cdot\mathbf{1} & (8)\\
P & \equiv & \mathbf{1}\cdot P & (9)\\
\mathbf{0} & \equiv & \mathbf{0}\cdot P & (10)\\
P\cdot(Q\cdot R) & \equiv & (P\cdot Q)\cdot R & (11)\\
P\cdot R+Q\cdot R & \equiv & (P+Q)\cdot R & (12)\\
P\cdot Q+P\cdot R & \leq & P\cdot(Q+R) & (13)\\
(P\ {}_{p}{\oplus}\ Q)\cdot R & \equiv & P\cdot R\ {}_{p}{\oplus}\ Q\cdot R & (14)\\
P\cdot(Q\ {}_{p}{\oplus}\ R) & \leq & P\cdot Q\ {}_{p}{\oplus}\ P\cdot R & (15)
\end{array}$$

Proposition 4.3. *The Kleene star satisfies the following laws:*

$$\begin{array}{rcll}
P^* & \equiv & \mathbf{1}+P\cdot P^* & (16)\\
P\cdot Q\leq Q & \Rightarrow & P^*\cdot Q\leq Q & (17)
\end{array}$$

For the parallel composition, we assume synchronisation over all external actions and denote it simply with $\|$ without any frame as defined in Section 2.

Proposition 4.4. *The parallel composition satisfies*

$$\begin{array}{rcll}
P\|Q & \equiv & Q\|P & (18)\\
P\|(Q\|R) & \equiv & (P\|Q)\|R & (19)\\
(P\|Q)\cdot(P'\|Q') & \leq & P\cdot P'\|Q\cdot Q' & (20)\\
P\|Q+P\|R & \leq & P\|(Q+R) & (21)\\
P\|Q\ {}_{p}{\oplus}\ P\|R & \equiv & P\|(Q\ {}_{p}{\oplus}\ R) & (22)
\end{array}$$

Notice that we cannot have equality for the interchange law (20) even with a fully synchronised $\|$. For example $\mathbf{a}\|\mathbf{a} \equiv (\mathbf{a}\cdot\mathbf{1})\|(\mathbf{1}\cdot\mathbf{a}) > (\mathbf{a}\|\mathbf{1})\cdot(\mathbf{1}\|\mathbf{a}) \equiv \mathbf{0}$ where we assume that the action a is external, hence sychronised.

Theorem 4.5. $(\mathbf{PAut},+,\cdot,\|,^*,\mathbf{deadlock},\mathbf{skip})$ *is a weak concurrent Kleene algebra.*

Definition 4.6. *A probabilistic concurrent Kleene algebra is a weak concurrent Kleene algebra with a collection of probabilistic choices* ${}_p\!\oplus$, $p \in [0,1]$, *satisfying equations (5-7,14-15,22).*

Example 4.7. *We end this section by providing an algebraic proof for the existence of a simulation between the automata in Figure 2. First, we express the least fixed point of the function* f *of Example 2.8 as promised. We prove that*

$$f(X) = (\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})\cdot X\cdot\mathbf{0}$$

using equations (10) and (14). Now, we show that the least fixed point of $f(X) = P\cdot X\cdot\mathbf{0}$ *is* $P^*\cdot\mathbf{0}$. *In fact* $f(P^*\cdot\mathbf{0}) = P\cdot P^*\cdot\mathbf{0} = (\mathbf{1}+P\cdot P^*)\cdot\mathbf{0} = P^*\cdot\mathbf{0}$ *because of equations (9), (12) and (16). Now let* Q *be a suffix point of* f *i.e.* $P\cdot Q\cdot\mathbf{0}\leq Q$, *then monotonicity and Equation (10) implies* $P\cdot Q\cdot\mathbf{0}\leq Q\cdot\mathbf{0}$. *Therefore,* $P^*\cdot Q\cdot\mathbf{0}\leq Q\cdot\mathbf{0}\leq Q$ *because of the induction law (17) and* $\mathbf{0}\leq\mathbf{1}$. *Hence* $P^*\cdot\mathbf{0}\leq Q$ *follows from Equation (10) and monotonicity of* $\cdot$.

Therefore, the left hand side automaton is simulation equivalent to

$$M = (\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})^*\cdot\mathbf{0}$$

One unfold of this automaton gives

$$\begin{array}{rcl}
M & \equiv & (\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})\cdot(\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})^*\cdot\mathbf{0}\\
 & \equiv & \texttt{stuck}\cdot\texttt{kick}\cdot(\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})^*\cdot\mathbf{0}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0}\\
 & \equiv & \texttt{stuck}\cdot\texttt{kick}\cdot(\texttt{stuck}\cdot\texttt{kick}\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0})\cdot M\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0}\\
 & \leq & (\texttt{stuck}\cdot\texttt{kick}\cdot\texttt{stuck}\cdot\texttt{kick}\cdot M\ {}_{0.2}\!\oplus\ \texttt{stuck}\cdot\texttt{kick}\cdot\texttt{tea}\cdot\mathbf{0})\ {}_{0.2}\!\oplus\ \texttt{tea}\cdot\mathbf{0}\\
 & \equiv & \texttt{stuck}\cdot\texttt{kick}\cdot\texttt{stuck}\cdot\texttt{kick}\cdot M\ {}_{0.04}\!\oplus\ (\texttt{stuck}\cdot\texttt{kick}\cdot\texttt{tea}\cdot\mathbf{0}\ {}_{0.16/0.96}\!\oplus\ \texttt{tea}\cdot\mathbf{0})\\
 & \leq & \texttt{stuck}\cdot\texttt{kick}\cdot\texttt{stuck}\cdot\texttt{kick}\cdot M\ {}_{0.04}\!\oplus\ (\texttt{stuck}\cdot\texttt{kick}\cdot\texttt{tea}\cdot\mathbf{0}+\texttt{tea}\cdot\mathbf{0})
\end{array}$$

The second equality follows from Equations (14) and (10). The third equality follows from an unfolding of the Kleene star and the definition of M. The fourth inequality follows from Equation (15). The fifth equality follows from Equation (7) and in the last equality, we have used the fact that $P\ {}_p\!\oplus\ Q\leq P+Q$. *We use monotonicity to finally deduce that* $M\leq H$ *because* $M\leq\mathbf{run}(\{\texttt{kick},\texttt{tea},\texttt{fail}\})$ *where* $\mathbf{run}(\{a_i\}_{i=0,n})) = (\sum_{i=0}^{n} a_i)^*$ *and*

$$H = \texttt{kick}\cdot\texttt{kick}\cdot\mathbf{run}(\{\texttt{kick},\texttt{tea},\texttt{fail}\})\ {}_{0.04}\!\oplus\ (\texttt{kick}\cdot\texttt{tea}\cdot\mathbf{0}+\texttt{tea}\cdot\mathbf{0}).$$

5 Rely-Guarantee Rules

The rely/guarantee formalism provides a powerful tool for verifying a system with multiple interacting components. The concept is based on deriving the properties of the larger system through the use of inference rules on the specification of the components. We are interested in generating these inference rules algebraically.

In this section we extend the algebraic formulation of the rely-guarantee calculus in [5] to include probabilistic systems. Notice that the automata model provides a particular interpretation of rely/guarantee tuples because it is action based though the algebraic laws persist to any model satisfying the axioms. Moreover, our results are based on the definition of [5] instead of providing another interpretation of Jones' rely/guarantee components [6].

A rely/guarantee quintuple is composed of five components $P\;R\;\{U\}\;Q\;G$ where P,Q are pre/postconditions, R is a rely property and G is the guaranteed part. These components are usually algebraic specifications and can be interpreted as automata when needed where the parallel composition has a fixed frame. In the automata model, **run** denotes the automata with self-loop constructed from the external actions and the algebraic proofs are valid for every model where **run** is replaced by the unit of $\|$ (if it exists).

Definition 5.1. *A rely guarantee quintuple $P\;R\;\{U\}\;Q\;G$ holds if and only if*

$$P\cdot(R\|U)\leq Q \quad\wedge\quad U\leq G$$

In other words, if U is part of a system that satisfies the property R, then the system will satisfy the specification determined by (P,Q) and U's behaviour is determined by G.

The difference between our approach and that of Kwiatkowska et al in [13] lies in the definition of the rely/guarantee tuple, that is, the interpretations of tuple satisfaction are different. In [13], the guaranteed property is somehow part of the postcondition and they are expressed using safety properties instead of simulation relations. Our approach is an extension of the work in [5] and has a similar flavour as [8].

We now provide some simple rely/guarantee rules together with their algebraic proofs. Notice that these rules are valid for all models satisfying the axioms of probabilistic concurrent Kleene algebra. This nicely illustrates the power of algebras where they provide results that are model independent. Of course, this power comes with the disadvantage that many details are lost through abstraction. The importance of these details depends on the system and the properties to be studied.

Remind that, given a finite set of external actions $A=\{a_1,\dots,a_n\}$, we denote $\mathbf{run}(A)=(\sum_{i=1}^{n}a_i)^*$ which is similar to the run of standard CSP. Notice that for every term P constructed from A and the algebraic operators, we have $P\leq\mathbf{run}(A)$ and $P_A\|\mathbf{run}(A)=P$.

An *isolated system* is composed of two components that interact without any interference from an outside environment. An example of isolated system is given by our vending machine and user.

Proposition 5.2. *For every term P,P',Q,Q',U,U',R,R',G and G' we have the following concurrent rule for isolated system:*

$$\frac{P\;R\;\{U\}\;Q\;G \qquad P'\;R'\;\{U'\}\;Q'\;G' \qquad G\leq R' \qquad G'\leq R}{T\;\mathbf{run}\;\{U\|U'\}\;Q\;(G\|G')} \tag{23}$$

where **run** *is constructed from the external actions of $U\|U'$ and the conclusion is valid whenever $T\leq P$ and $T\leq P'$.*

Proof. We have $P\cdot(R\|U)\leq Q$ and $U\leq G$ from the interpretation of the first quintuple. The premise $G\leq R'$ implies that $U\leq R'$. Therefore, monotonicity yields $P\cdot(U\|U')\leq Q$. Hence, if $T\leq P$ and $T\leq P'$ then $T\cdot(U\|U')$ satisfies both Q. Since we assume that the external actions of U and U' coincide which is used in the parametrisation of $\|$ and $\leq$, we have $\mathbf{run}\|(U\|U')=U\|U'$ and we obtain the guaranteed part of the conclusion with monotonicity of $\|$. □

The Rule (23) implies that the quintuple $T\;\mathbf{run}\;\{U\|U'\}\;Q'\;(G\|G')$ also holds. This rule can only be used for isolated systems.

Corollary 5.3. *The following asymmetric rule holds for isolated systems*

$$\frac{\mathbf{1}\;\mathbf{run}\;\{U\}\;\mathbf{run}\;G \qquad P'\;R'\;\{U'\}\;Q'\;G' \qquad G\leq R'}{P'\;\mathbf{run}\;\{U\|U'\}\;Q'\;(G\|G')} \tag{24}$$

When the system is not isolated, a more general rule is needed. We can show that if there exists S such that $S \leq R$, $S \leq R'$ and $S\|S \leq S$ then we can infer from the premises of Rule (23) that the quintuple

$$T \; S \; \{U\|U'\} \; Q \; (G\|G')$$

holds, where $T \leq P$ and $T \leq P'$. In other words, if the system $U\|U'$ is run within an environment that guarantees S then it satisfies the postconditions Q and Q' and guarantees $G\|G'$.

Proposition 5.4. *We have the following sequential rule:*

$$\frac{P \; R \; \{U\} \; Q \; G \qquad P' \; R' \; \{U'\} \; Q' \; G' \qquad Q \leq P' \\ (R\|U) \cdot (R'\|U') = (R \cdot R')\|(U \cdot U')}{P \; (R \cdot R') \; \{U \cdot U'\} \; Q' \; (G \cdot G')} \tag{25}$$

Proof. We have $P \cdot (R\|U) \leq Q$ and $P' \cdot (R'\|U') \leq Q'$, since $Q \leq P'$, monotonicity implies $P \cdot (R\|U) \cdot (R'\|U') \leq Q'$ and the last premise gives us $P \cdot [(R \cdot R')\|(U \cdot U')] \leq Q'$. The guaranteed part follows from monotonicity of $\cdot$. □

These rules support the construction of larger systems from the components using concurrent and/or sequential compositions. Together with these rely/guarantee rules, we will also make extensive use of Equation (15) because it provides the transport of probabilistic choices to the "upper level" of the specification automaton. That is, it allows us to write simple rely properties and postconditions of the form $(bad + good) \; {}_p\!\oplus \; good$ or even $bad \; {}_p\!\oplus \; good$ where *good* and *bad* are usually standard automata. With the testing interpretation of simulation, we conclude that the maximal probability for *bad* to happen is bounded from above by $1 - p$.

Example 5.5. *Using our running example, we have the following rely/guarantee quintuples*

$$\mathbf{1} \; \mathbf{run} \; \{M\} \; \mathbf{run} \; H \qquad \textit{and} \qquad \texttt{coin} \; H \; \{U'\} \; Q \; \mathbf{run}$$

where Q is given by the diagram in Figure 3.

The first quintuple has been established algebraically in the previous section, that is, $M \leq H$. As for the second one, it is clear that $H\|U' \leq Q'$ which can be established by direct automata calculation or using the algebra as before.

Therefore, Rule (24) implies that the quintuple

$$\texttt{coin} \; \mathbf{run} \; \{M\|U'\} \; Q \; H$$

is valid. That is, $V\|U = \texttt{coin} \cdot (M\|U') \leq Q$ [2] *which says that with probability at least* 0.96 *the user needs to kick the machine at most once to get tea. (Note we have used the fact that $(\texttt{coin} \cdot M)\|(\texttt{coin} \cdot U') \equiv \texttt{coin} \cdot (M\|U')$ which is a stronger version of the interchange law).*

[2] Notice that $M\|U'$ does not enable any transition labelled by `fail` because that action has to be synchronised. But the established property says that fail can occur only with probability at most 0.04, but not that it has to occur at all. If we wanted an explicit `fail`, we can form self loops labelled by `fail` on each state of V.

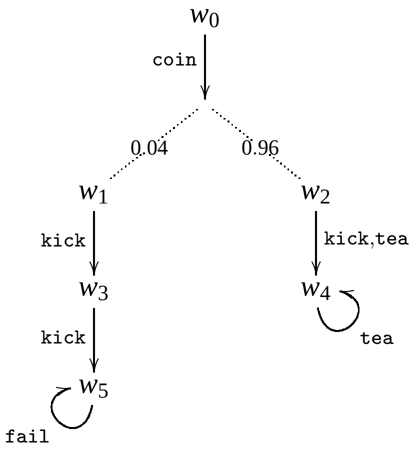

Figure 3: The postcondition for the system in the form $Q = \mathtt{coin} \cdot Q'$.

6 Related Work

This paper aims to develop an algebra that accounts for nondeterminism, probability and concurrency in a Kleene algebraic fashion [1, 5, 9, 10, 16, 17]. To the best of our knowledge, there is no algebraic structure in the style of Kleene algebras that includes nondeterminism, probability and concurrency. The algebra we develop is a mixture of concurrent [5] and probabilistic [16] Kleene algebras augmented with probabilistic choices to manipulate quantitative properties. The soundness of the algebra is established using probabilistic automata modulo simulation as in [2]. That paper provides an extensive survey of the algebraic laws for such model in the style of a recursion-free process algebra, hence there is no sequential composition (which is mandatory for the encoding of interference) nor Kleene star (which provides a meaning for terminating loops). Another related work is the quantitative Kleene coalgebra of [23]. That paper focuses on unifying various constructions of transition systems through the use of functor-coalgebras. It also provides a coalgebra composed of algebraic expressions though the main focus is on the generalisation of Kleene's correspondence between operational semantics and the expressions through the use of derivatives. Moreover, these expressions are generated from a signature that is different from the one we propose in this paper, most importantly, concurrency is not considered. Reconciling the two approaches seems very promising.

The algebraic approach to the rely/guarantee calculus of Section 5 is a straightforward generalisation of [5] but now proved to be valid for probabilistic scenario as well. Our approach is conceptually related to [8] where the rules are interpreted against probabilistic strong simulation. The precongruence of [8] is not applicable in our setting because we need explicit internal actions to construct the algebraic operators and these internal actions cannot usually be removed in presence of probability and nondeterminism. Hence, strong simulation is inadequate for an algebraic approach that should be a generalisation of Kleene algebras. The disadvantage of using weak simulation is that efficient decidability is unknown in contrast to strong simulation [8]. Though this is an interesting problem, our focus is on using the algebra for direct proof of the existence of a simulation rather than computing a simulation relation directly. Therefore, we can use theorem provers or proof assistants to handle the automation.

7 Conclusion

This paper presented a Kleene algebraic approach to systems exhibiting nondeterminism, probability and concurrency. A sound axiomatisation has been presented with respect to the set of probabilistic automata modulo probabilistic simulation. The simulation used is equivalent to the vector may testing preorder of [2] which provides the interpretation of maximal probability of failure. The algebra was constructed as a combination of probabilistic and concurrent Kleene algebras. The nondeterminisms of both algebras coincide, probability is handled by the subdistributivity law of probabilistic Kleene algebra as well as the explicit probabilistic choices, and concurrency is mainly handled by a weakening of concurrent Kleene algebra.

An important law of probability is summarised by the equation

$$P \cdot (Q \;{}_p\!\oplus\; R) \leq (P \cdot Q) \;{}_p\!\oplus\; (P \cdot R).$$

In multiple cases, it allows us to "move probabilities upward" and to write specification of the form $(bad + good) \;{}_p\!\oplus\; good$. This expression says that the probability of executing *bad* is bounded above by p because of the testing interpretation of the simulation. In fact, as shown in the vending machine example, it is also possible to write properties such as "the maximal probability of failing after k-steps is less than p".

We note finally that the rely/guarantee calculus is valid for any model satisfying the axioms. This applies to true-concurrent models as well as fragments of the establised automata models as long as they have sequential composition instead of prefixing. However, the action-based interleaving model provides a limited application of the rely/guarantee calculus. Therefore, the construction of an alternative model of the algebra using true-concurrency semantics is part of our future investigation.

References

[1] J. H. Conway (1971): *Regular Algebra and Finite Machines*. Chapman and Hall, Mathematics series.

[2] Y. Deng, R. J. van Glabbeek, M. Hennessy, C. Morgan & C. Zhang (2007): *Remarks on Testing Probabilistic Processes*. *Electr. Notes Theor. Comput. Sci.* 172, pp. 359–397. doi:10.1016/j.entcs.2007.02.013.

[3] Y. Deng & R. Van Glabbeek (2007): *Characterising testing preorders for finite probabilistic processes*. In: *In LICS07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science. IEEE Computer Society Press, Los Alamitos, CA*, pp. 313–325, doi:10.1109/LICS.2007.15.

[4] J. Desharnais, B. Möller & G. Struth (2006): *Kleene algebra with domain*. *ACM Trans. Comput. Logic* 7, pp. 798–833, doi:10.1145/1183278.1183285.

[5] C. A. R. Hoare, B. Möller, G. Struth & I. Wehrman (2011): *Concurrent Kleene Algebra and its Foundations*. Journal of Logic and Algebraic Programming 80, pp. 266–296, doi:10.1016/j.jlap.2011.04.005.

[6] C. B. Jones (1981): *Development methods for computer programs including a notion of interference*. Technical Monograph. Programming Research Group, Oxford University Computing Laboratory. Available at http://books.google.com.au/books?id=zjguSwAACAAJ.

[7] S. C. Kleene (1951): *Representation of Events in Nerve Nets and Finite Automata*. *Automata Studies.*

[8] A. Komuravelli, C. S. Pasareanu & E. M. Clarke (2012): *Assume-Guarantee Abstraction Refinement for Probabilistic Systems*. *CoRR* abs/1207.5086. Available at http://arxiv.org/abs/1207.5086, doi:10.1007/978-3-642-31424-7_25.

[9] D. Kozen (1994): *A completeness theorem for Kleene algebras and the algebra of regular events*. *Infor. and Comput.* 110(2), pp. 366–390, doi:10.1006/inco.1994.1037.

[10] D. Kozen (1997): *Kleene algebra with tests*. *ACM Trans. Program. Lang. Syst.* 19, pp. 427–443, doi:10.1145/256167.256195.

[11] D. Kozen (2000): *On Hoare logic and Kleene algebra with tests*. *Trans. Computational Logic* 1(1), pp. 60–76, doi:10.1145/343369.343378.

[12] D. Kozen (2003): *Kleene Algebras with Tests and the Static Analysis of Programs*. Technical Report TR2003-1915, Computer Science Department, Cornell University.

[13] M. Kwiatkowska, G. Norman, D. Parker & H. Qu (2010): *Assume-Guarantee verification for probabilistic systems*. In: *Proceedings of the 16th international conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'10, Springer-Verlag, Berlin, Heidelberg, pp. 23–37, doi:10.1007/978-3-642-12002-2_3.

[14] N. A. Lynch, R. Segala & F. W. Vaandrager (2003): *Compositionality for Probabilistic Automata*. In: *CONCUR*, pp. 204–222. doi:10.1007/978-3-540-45187-7_14.

[15] A. McIver, T. M. Rabehaja & G. Struth (2011): *On probabilistic Kleene algebras, automata and simulations*. In: *Proceedings of the 12th international conference on Relational and algebraic methods in computer science*, RAMICS'11, Springer-Verlag, Berlin, Heidelberg, pp. 264–279. Available at http://dl.acm.org/citation.cfm?id=2018285.2018305, doi:10.1007/978-3-642-21070-9_20.

[16] A. K. McIver & C. C. Morgan (2004): *Abstraction, Refinement And Proof For Probabilistic Systems (Monographs in Computer Science)*. SpringerVerlag.

[17] B. Möller (2007): *Kleene getting lazy*. *Sci. Comput. Program.* 65, pp. 195–214, doi:10.1016/j.scico.2006.01.010. Available at http://dl.acm.org/citation.cfm?id=1225317.1225705.

[18] M. O. Rabin (1976): *Probabilistic Algorithms*. Technical Report RC 6164 (#26545), IBM Research Division, San Jose, Yorktown, Zurich.

[19] A. Salomaa (1966): *Two Complete Axiom Systems for the Algebra of Regular Events*. *J. ACM* 13, pp. 158–169, doi:10.1145/321312.321326.

[20] S. Schneider (2012): *Incorporating time to an integrated formal method*. Available at http://www.nii.ac.jp/shonan/seminar017/files/2012/06/Slides1.pdf.

[21] R. Segala (1995): *A Compositional Trace-Based Semantics for Probabilistic Automata*. In: *CONCUR*, pp. 234–248. doi:10.1007/3-540-60218-6_17.

[22] R. Segala (1996): *Testing Probabilistic Automata*. In: *CONCUR*, pp. 299–314. doi:10.1007/3-540-61604-7_62.

[23] A. Silva, F. Bonchi, M. Bonsangue & J. Rutten (2011): *Quantitative Kleene coalgebras*. *Inf. Comput.* 209(5), pp. 822–849, doi:10.1016/j.ic.2010.09.007.

[24] E. W. Stark & S. A. Smolka (1996): *A Complete Axiom System for Finite-State Probabilistic Processes*. In: *In Proof, Language, and Interaction: Essays in Honour of Robin Milner*, MIT Press, pp. 571–595.

Appendix

Proposition A.1. *Simulation is a precongruence i.e. if $P \leq Q$ then $P+R \leq Q+R$, $P \cdot R \leq Q \cdot R$, $P^* \leq Q^*$, $P \;{}_p\!\oplus R \leq P \;{}_p\!\oplus R$, $P_A || R \leq Q_A || R$ and the same holds for binary operators when the order of the arguments is reversed.*

Proof. The construction of simulation in each of these cases is easy and they have been proven elsewhere [2, 22] except for sequential composition and Kleene star which we present here.

Let P, Q, R be probabilistic automata such that $S : P \to Q$ is a simulation. We show that the relation

$$S' = \{(x, \nu) \mid (x \in P \wedge \nu \in \mathscr{D}(Q) \wedge xS\nu) \vee (x \in R \wedge \nu = \delta_x)\}$$

is a simulation from $P \cdot R$ to $Q \cdot R$. Since the initial distributions of $P \cdot R$ and $Q \cdot R$ are respectively the initial distributions of P and Q, Properties (a) and (c) of Definition 3.1 are immediate for this case. As for part (b), let $x \xrightarrow{a} \mu \in \longrightarrow_{P \cdot R}$ and $(x, \nu) \in S'$. There are three cases:

- $x \xrightarrow{a} \mu \in \longrightarrow_P$, and then $(x, \nu) \in S$ and we are done because S is a simulation from P to Q and $S \subseteq S'$.
- $x \xrightarrow{a} \mu \in \longrightarrow_R$, and then $\nu = \delta_x$ and we are done because id_R is a simulation.
- $x \xrightarrow{\tau} \psi_0$ such that $x \in F_P$ and ψ_0 is the initial distribution of R. We are only interested in the case where x is reachable from ϕ_0. Since $P \leq Q$, there exist $\nu \in \mathscr{D}(Q)$ and $\nu' \in \mathscr{D}(F_Q)$ such that $xS\nu$, and $\nu \Longrightarrow \nu'$. Therefore, $\nu \Longrightarrow \psi_0$ is a valid weak transition in $Q \cdot R$ and we have $\psi_0 \overline{id_R} \psi_0$.

The dual $R \cdot P \leq R \cdot Q$ also holds using an analogous relation.

Finally, we show that Kleene star is monotonic with respect to $\leq$. Let $S : P \to Q$ be a simulation and denote by x_P, x_Q the respective initial states of P^* and Q^* and consider the relation

$$S' = S \cup \{(x_P, \delta_{x_Q})\}.$$

Properties (a) and (b) are routine. Let $x \xrightarrow{a} \mu \in \longrightarrow_{P^*}$ and $xS'\nu$, there are two cases:

- $xS\nu$, then $x \xrightarrow{a} \mu \in \longrightarrow_P$ or $x \in F_P$ and $\mu = \delta_{x_P}$.
  - If $x \xrightarrow{a} \mu \in \longrightarrow_P$ then there exists $\nu' \in \Delta(Q)$ such that $\nu \overset{\bar{a}}{\Longrightarrow} \nu'$ and $\mu S \nu'$ because S is simulation. Since $S \subseteq S'$, we deduce that $\mu S' \nu'$.
  - If $x \in F_P$ and $x \xrightarrow{\tau} \delta_{x_P}$, we can assume that x is a reachable final state of P because we consider probabilistic automata to be simulation equivalent if and only if they reachable parts are. Since S is simulation and $xS\nu$, we have $\nu \Longrightarrow \nu'$ for some $\nu' \in \mathscr{D}(F_Q)$. Therefore, $\nu \Longrightarrow \delta_{x_Q}$ and $x_P S' \delta_{x_Q}$ by definition of S'.
- $x_P S' \delta_{x_Q}$ and $x_P \xrightarrow{\tau} \mu_0$ where μ_0 is the initial distribution of P. Since S is a simulation, there exists ν_0' such that $\nu_0 \Longrightarrow \nu_0'$ and $\mu_0 \overline{S} \nu_0'$. Hence, $\delta_{x_Q} \Longrightarrow \nu_0'$ and $\mu_0 S' \nu_0'$. □

Proposition A.2. *The sequential composition satisfies Equations (8-15).*

Proof. Equations (8) and (9) are clear and (10) follows form the fact that $P \equiv Q$ iff their reachable parts are simulation equivalent.

Associativity (11) is evident because the left and right hand side automata are exactly the same.

For distributivity (12), let us write the left hand side term as $P \cdot R + Q \cdot R_c$ where R_c is a copy of R whose states are renamed to x_c for every state x of R. We construct a relation $S \subseteq (P \cup Q \cup \{z\} \cup R \cup$

$R_c) \times \mathscr{D}(P \cup Q \cup \{z\} \cup R)$ such that $S = \{(x, \delta_x), (x_c, \delta_x) \mid x \in R \wedge x_c$ is the copy of $x\} \cup id_{P+Q}$. It is easy to shown that S is a simulation and so is its inverse.

For subdistributivity (13), we consider the relation

$$S = \{(x, \delta_x), (x_c, \delta_x) \mid x \in P \wedge x_c \text{ is the copy of } x\} \cup \{(z, \mu_0)\} \cup id_Q \cup id_R$$

where z is the initial state of $P \cdot Q + P \cdot R$ and μ_0 is the initial distribution of P. It is again straightforward to prove that S is indeed a simulation.

Equation (14) is proved using the exact same simulation constructed in the case of Equation (12).

For the last equation, let

$$S = \{(x, \delta_x \; {}_p\!\oplus \; \delta_{x_c}) \mid x \in P \wedge x_c \text{ is the copy of } x\} \cup id_Q \cup id_R$$

This simulation essentially says that we carry down the probabilistic choice ${}_p\!\oplus$ on the left hand side until it needs to be resolved.

- By construction of the simulation, we have $\mu_0 \overline{S} (\mu_0 \; {}_p\!\oplus \; \mu_{0c})$ where μ_0 and μ_{0c} are the respective initial distributions of P and P_c.
- Let $x \xrightarrow{a} \mu \in \longrightarrow_{P \cdot (Q \; {}_p\!\oplus R)}$ and $xS\nu$, there are three cases
  - $x \xrightarrow{a} \mu \in \longrightarrow_P$, therefore $\nu = \delta_x \; {}_p\!\oplus \; \delta_{x_c}$ and $\nu \xrightarrow{\overline{a}} \mu \; {}_p\!\oplus \; \mu_c$ where μ_c is the copy of μ.
  - $x \xrightarrow{a} \mu \in \longrightarrow_Q \cup \longrightarrow_R$, then we are done because $id_Q \cup id_R \subseteq S$.
  - $x \xrightarrow{\tau} \mu_{0Q} \; {}_p\!\oplus \; \mu_{0R}$ and $x \in F_P$, then $x \xrightarrow{\tau} \mu_{0Q}$ and $x \xrightarrow{\tau} \mu_{0R}$ are valid transitions of $P \cdot Q$ and $P_c \cdot R$. But $\nu = \delta_x \; {}_p\!\oplus \; \delta_{x_c}$ because $x \in P$, therefore $\nu \xrightarrow{\overline{\tau}} \mu_{0Q} \; {}_p\!\oplus \; \mu_{0R}$.
- Let $xS\nu$ and x is a final state. By definition of ${}_p\!\oplus$, $x \in F_Q \cup F_R$ and hence $\nu = \delta_x \in \mathscr{D}(F_Q) \cup \mathscr{D}(F_R)$.

□

Before we prove that the Kleene star satisfies the usual unfold and left induction law of probabilistic Kleene algebra, let us introduce the notion of unfolding which will simplify the proof of the induction law considerably. It is essentially a cleaner version of our construction in [15]. We denote **unfold**(P) the unfold of any automaton P [14], that is, the automaton

$$(\mathrm{Path}(P), \longrightarrow, \mu_0, F)$$

where

$$\longrightarrow = \{\alpha \xrightarrow{a} \mu \mid \alpha \in \mathrm{Path}(P) \wedge \exists \mu' \in \mathscr{D}(P) : \mathrm{last}(\alpha) \xrightarrow{a} \mu' \in \longrightarrow_P \wedge \mu(\alpha a x) = \mu'(x)\}$$

and

$$F = \{\alpha \in \mathrm{Path}(P) \mid \mathrm{last}(\alpha) \in F_P\}.$$

and μ_0 is the initial distribution of P. This construction provides us with an automaton whose states are finite paths in P and there is a transition between two paths α, α' iff $\alpha' = \alpha a x$ where $a \in \Sigma_\tau$ and $x \in P$. Such a transition is labelled by a. It is now easy to show that the relation $\{(\alpha, \delta_{\mathrm{last}(\alpha)}) \mid \alpha \in \mathrm{Path}(P)\}$ is a simulation from **unfold**(P) to P and the inverse is also a simulation from P to **unfold**(P) [14].

Proposition A.3. *Kleene star satisfies Equation (16) and the induction law (17).*

Proof. Let u be the initial state of $\mathbf{1}+P\cdot P^*$ and v be the initial state of P^*. Since we add only one state and some transition in the construction of P^*, we denote $x_* \in P^*$ the state corresponding to $x \in P$. To prove Equation (16), we consider the relation

$$S = \{(x_*, \delta_{x_*}), (x_*, \delta_x) \mid x \in P\} \cup \{(v, \delta_v), (v, \delta_u)\}$$

from P^* to $\mathbf{1}+P\cdot P^*$. We now prove that S is a simulation.

- For the initial distribution, we have $vS\delta_u$.
- Let $y \xrightarrow{a} \mu$ be a valid transition in P^* and $yS\nu$. There are two cases
  - $y = v$ and the transition is $v \xrightarrow{\tau} \mu_0$ where μ_0 is the initial distribution of P. If $\nu = \delta_v$ then we are done because $\{(v,\delta_v)\} \cup \{(x_*, \delta_{x_*}) \mid x \in P\} = id_{P^*}$. Else, $\nu = \delta_u$ and $u \xrightarrow{\tau} \mu_0$ is a valid transition in $\mathbf{1}+P\cdot P^*$.
  - $y = x_*$ for some $x \in P$ and:
    - $x_* \xrightarrow{a} \mu_*$ is the copy of a transition of P. Therefore, if $\nu = \delta_{x_*}$ then the same transition belongs to $P\cdot P^*$. If $\nu = \delta_x$ then $x \xrightarrow{a} \mu$ is a transition of P and $\mu_* \overline{S} \mu$.
    - or, $x_* \xrightarrow{\tau} \delta_v$ and in this case, if $\nu = \delta_{x^*}$ then that transition belongs to $P\cdot P^*$ again, else $\nu = \delta_x$ and $x \in F_P$. Therefore, $\delta_x \xrightarrow{\overline{\tau}} \delta_v$ is a lifted transition in $P\cdot P^*$.
- The conservation of final state is obvious because $F_{P^*} = \{v\}$ and $u \xrightarrow{\tau} \delta_z$ where z is the final state of $\mathbf{1}$ in $\mathbf{1}+P\cdot P^*$.

With the similar reasoning, it holds that the inverse of S is a simulation from $\mathbf{1}+P\cdot P^*$ to P^*.

We now prove the induction law (17). We can assume that P is loop-free by unfolding it and therefore $1+P\cdot\mathbf{unfold}(P^*)$ is again loop-free and simulation equivalent to P^*. Let $\mathscr{F}(X) = \mathbf{1}+P\cdot X$. Since $P\cdot\mathbf{0} \le P$, we show easily by induction that $\mathbf{unfold}(\mathscr{F}^n(\mathbf{0})) \trianglelefteq \mathbf{unfold}(\mathscr{F}^{n+1}(\mathbf{0}))$ where $\trianglelefteq$ is the inclusion of automata i.e. $X \trianglelefteq Y$ if the state space of X is a subset of the state space of Y, transitions of X are transitions of Y and $F_X \subseteq F_Y$. We can then construct a limit automta $\lim_n \mathscr{F}^n(\mathbf{0}) = \mathscr{F}^*(\mathbf{0})$ obtained as the countable union of component by component (the set of states is the union of the sets of states, the set of transitions is the union of sets of transitions,...). Since P has no cycle, it follows that $\mathscr{F}^*(\mathbf{0}) = \mathbf{unfold}(P^*)$.

Now assume that $P\cdot Q \le Q$, then $(\mathbf{1}+P\cdot\mathbf{0})\cdot Q \le (\mathbf{1}+P)\cdot Q \le Q$ and by induction, $\mathscr{F}^n(\mathbf{0})\cdot Q \le Q$ for every $n \in \mathbb{N}$. Moreover, since $\mathbf{unfold}(\mathscr{F}^n(\mathbf{0})) \trianglelefteq \mathbf{unfold}(\mathscr{F}^{n+1}(\mathbf{0}))$, we have $\mathbf{unfold}(\mathscr{F}^n(\mathbf{0}))\cdot Q \trianglelefteq \mathbf{unfold}(\mathscr{F}^{n+1}(\mathbf{0}))\cdot Q$ and since $F_{\mathbf{unfold}(\mathscr{F}^n(\mathbf{0}))} \subseteq F_{\mathbf{unfold}(\mathscr{F}^{n+1}(\mathbf{0}))}$ (inclusion of final states), $\lim_n(\mathbf{unfold}(\mathscr{F}^n(\mathbf{0}))\cdot Q) = \mathscr{F}^*(\mathbf{0})\cdot Q$ (the two automaton are equal by construction). Hence $\mathscr{F}^*(\mathbf{0})\cdot Q \le Q$. □

Proposition A.4. *The parallel composition satisfies Equations (18-22).*

Proof. Equations (18), (21) and (22) are proven in [2].

For the associativity, remind that when the frame is fixed then there is a standard simulation between $P\|(Q\|R)$ and $(P\|Q)\|R$ by associating each tuple $(x,(y,z))$ to $((x,y),z)$. That simulation is lifted to $(P\times(Q\times R))\times\mathscr{D}((P\times Q)\times R)$ using point distributions and dually.

As for the interchange law (20), we consider the injection

$$S = \{((x,y), \delta_{(x,y)}) \mid (x,y) \in (P\times Q)\cup(P'\times Q')\}$$

from $U = (P\|Q)\cdot(P'\|Q')$ to $V = P\cdot P'\|Q\cdot Q'$.

- Using the definition of $\|$ and $\cdot$, we deduce that the initial distributions of U and V are the same.

- Let $(x,y)S\delta_{(x,y)}$ and $(x,y) \xrightarrow{a} \mu \in \longrightarrow_U$. There are three cases:
  - $(x,y) \in P \times Q$ and $\mu = \mu_P \times \mu_Q \in \mathscr{D}(P \times Q)$. In all three cases in the definition of $\|$, we have $(x,y) \xrightarrow{a} \mu_P \times \mu_Q \in \longrightarrow_V$.
  - $(x,y) \in P' \times Q'$ and $\mu = \mu_{P'} \times \mu_{Q'} \in \mathscr{D}(P' \times Q')$. This is the same as the previous case.
  - $(x,y) \in F_P \times F_Q$ and the transition is $(x,y) \xrightarrow{\tau} \mu_{0P'} \times \mu_{0Q'}$ where $\mu_{0P'}, \mu_{0Q'}$ are the respective initial distributions of P', Q'. Since $x \in F_P$, $x \xrightarrow{\tau} \mu_{0P'} \in \longrightarrow_{PP'}$ and similarly for $y \in F_Q$. Therefore, $(x,y) \xrightarrow{\tau} \mu_{0P'} \times \delta_y \xrightarrow{\overline{\tau}} \mu_{0P'} \times \mu_{0Q'}$ i.e. $(x,y) \Longrightarrow \mu_{0P'} \times \mu_{0Q'}$ is a weak lifted transition in V.
- Finally, $F_U = F_V$. □