Product-forms for Probabilistic Input/Output Automata

Filippo Cavallin, Andrea Marin and Sabina Rossi
DAIS - Università Ca' Foscari Venezia
Venezia, Italy
Email: {fcavalli,marin,srossi}@dais.unive.it

***Abstract*—Probabilistic I/O automata (PIOAs) provide a modelling framework that is well suited for describing and analyzing distributed and concurrent systems. They incorporate a notion of probabilistic choice as well as a notion of composition that allows one to construct a PIOA for a composite system from a collection of simpler PIOAs representing the components. Differently from other probabilistic models, the local actions of a PIOA are associated with time delays governed by independent random variables with continuous-time exponential distributions. The contribution of this paper consists in studying the product-form property for PIOAs. Our main result is the formulation of a theorem giving sufficient conditions for a composition of PIOAs to be in product-form and hence to efficiently compute its stationary probabilities.**

I. Introduction

Probabilistic Input/Output automata (PIOAs) have been introduced in [21], [22] as a formalism aimed at modelling distributed and concurrent systems in a compositional way. However, the interest for their application goes beyond the purely engineering applications [3]. PIOAs incorporate a notion of probabilistic choice and time delays for locally controlled actions. The definition of formalisms for modelling probabilistic systems has been extensively investigated in the literature both in the field of process algebras and automata theory. One of the key-factors that characterises the proposed methodologies is clearly the semantics of the composition. Giving a reasonable way of composing probabilistic systems is challenging because the probabilities that are specified within each single component have a "local" meaning. In general, they are not sufficient to describe the probabilistic behaviour of the joint model without further assumptions such as the time scale. In the PIOA model this problem is solved by associating an exponentially distributed delay parameter with each state. Intuitively, a PIOA first draws a random delay time from an independent exponentially distributed random variable and then performs the probabilistic choice. Therefore, in the composition of a collection of PIOAs, the usual *race condition policy* used in [13], [20] is applied. PIOAs communicate via input and output actions and can perform internal non-communicating transitions. The communication is seen as a message transmitted on a labelled channel (that we call synchronisation label) by the output automaton. The synchronising automaton can read the message and perform a probabilistic transition accordingly. For each PIOA the sum of the probabilities associated with output and internal transitions, called locally controlled transitions, must be 1. On the other hand, upon the reception of a message, the PIOA immediately reacts, i.e., the sum of the probabilities associated with the message-receiving transitions outgoing from each state of a PIOA must be 1 for each of them separately.

PIOAs share with many other formalisms for the quantitative analysis of computer systems the property of having an underlying Markov process that describes the model evolution, and the problem of the exponential growth of the cardinality of the state spaces which makes the derivation of the quantitative indices unfeasible even for relatively small systems. The problem of defining compositional approaches to the quantitative analysis of PIOAs has been addressed in [21] for what concerns the transient behaviour. To the best of our knowledge, the problem of defining a compositional approach for studying the stationary behaviour of PIOAs remains open. In the literature of queueing networks this problem is often associated with the so called *product-form* analysis which is described in [16] and then extended in numerous subsequent works (see, e.g., [9], [15], [2]). In the last decade the product-form approach has been successfully extended to include Markovian process algebras [10], [11]. Informally, a product-form model can be studied without constructing the stochastic process underlying the composition of the simpler components forming the systems, but these can be studied in isolation. Hence, the computational effort required to compute the stationary quantitative indices is highly reduced.

The contribution of this paper consists in studying the product-form property for PIOAs. Our main result is the formulation of a theorem giving sufficient conditions for a composition of PIOAs to be in product-form and hence to efficiently compute the stationary probabilities.

Related work. The literature about the characterisation of product-forms for various formalisms is very rich. Since the pioneering work of Kelly [16], several other works have addressed the problem of characterising the product-form of queueing networks in terms of different properties. These works have been extended with the introduction of Gelenbe's G-networks [9] whose characterisation of the product-form is surveyed in [5], [17]. Similar efforts have been devoted to stochastic Petri nets product-forms [1] and Markovian process algebra [10]. The common denominator among all these contributions is that the considered models are based on continuous-time transition rates. Fewer results are available for probabilistic models expressed in terms of probabilistic process algebras or probabilistic automata. In the latter context we mention the results by Fourneau in [6], [7] but the synchronisation semantics which is considered is different from that of PIOAs and hence the results are not directly applicable.

Structure of the paper. The paper is organized as follows: Section II is devoted to the basic notions on continuous and discrete time Markov chains. Sections III and IV introduce, respectively, the class of stochastic automata (SAs) and the class of probabilistic I/O automata (PIOAs). In Section V we study the relations between SAs and PIOAs. Section VI presents our main result that is a product-form property for PIOAs. Finally, Section VII concludes the paper.

II. Basic Notions

Let $X(t)$ be a stochastic process taking values into a countable state space $\mathcal{S}$ for $t \in \mathbb{T}$. For a continuous time (CT) stochastic process, $\mathbb{T}$ is the real line $\mathbb{R}$, while for a discrete time (DT) stochastic process, $\mathbb{T}$ is the set of integers $\mathbb{Z}$.

For a time homogeneous, continuous (CTMC) and discrete (DTMC) time Markov chain the *transition rate* and the *transition probability*, respectively, from state s to state s' are:

$$CTMC : \lim_{\tau \to 0} \frac{P(X(t+\tau) = s' \mid X(t) = s)}{\tau} = q_{s,s'}, \;\; s \neq s'$$

$$DTMC : P(X(t+1) = s' \mid X(t) = s) = p_{s,s'}$$

We denote by q_s the parameter of the exponentially distributed residence time in the state s of the CTMC.

The infinitesimal generator matrix $\mathbf{Q}$ of a CTMC is the $|\mathcal{S}| \times |\mathcal{S}|$ matrix whose off-diagonal elements are the $q_{s,s'}$'s for $s \neq s'$ and whose diagonal elements are the negative sum of the extra diagonal elements of each row.

In the discrete time case, the value $p_{s,s'}$ denotes the *one-step transition probability*. The square matrix $\mathbf{P} = (p_{s,s'})_{s,s' \in \mathcal{S}}$ is called *one-step transition matrix* and is stochastic. A discrete time Markov process is *periodic* if there exists an integer $\delta > 1$ such that $P(X(t+n) = s \mid X(t) = s) = 0$ unless n is divisible by δ; otherwise the process is *aperiodic*. An ergodic Markov chain possesses a *limiting distribution*, that is the *unique* vector $\boldsymbol{\pi}$ of positive numbers π_s with $s \in \mathcal{S}$ such that

$$\lim_{t \to \infty} P(X(t) = s \mid X(0) = s_0) = \pi_s \tag{1}$$

and $\sum_{s \in \mathcal{S}} \pi_s = 1$. The system of global balance equations (GBEs) of a Markov chain is defined as:

$$CTMC : \boldsymbol{\pi}\mathbf{Q} = \mathbf{0} \tag{2}$$

$$DTMC : \boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}. \tag{3}$$

We say that any non-trivial solution of systesms (2) and (3) are *invariant measures* of the chains, while the only one that is a probability distributions (if it exists) is the stationary distribution. Notice that for irreducible, positive recurrent and periodic DTMCs, although the limiting distribution does not exist, system (3) still admits a unique solution summing to unity that is still called stationary distribution but does not clearly correspond to the limiting distribution.

III. Stochastic Automata

Many high-level specification languages for stochastic discrete-event systems are based on labelled automata [12], [13], [20], [8], [14], [4] equipped with a composition operator and timed actions whose delays are governed by independent random variables with continuous-time exponential distributions. In this paper we refer to the model of stochastic automata presented in [18], [19] which draws a distinction between *active* and *passive* action types, and forming the composition of automata, only active/passive synchronisations are permitted. An analogue semantics is proposed for Stochastic Automata Networks (SAN) in [20].

Definition 1: (Stochastic Automaton (SA)) A stochastic automaton P is a tuple $(\mathcal{S}_P, \mathcal{T}_P, \rightsquigarrow_P, q_P)$ where

- $\mathcal{S}_P$ is a denumerable set of states called *state space* of P
- $\mathcal{T}_P$ is a finite set of action types, partitioned into disjoint sets $\mathcal{A}_P$ of *active* types, $\mathcal{P}_P$ of *passive* types and the set $\{\tau\}$ of the *unknown* or *internal* type, with the types in $\mathcal{L}_P = \mathcal{A}_P \cup \{\tau\}$ called *locally controlled*
- $\rightsquigarrow_P \subseteq \mathcal{S}_P \times \mathcal{S}_P \times \mathcal{T}_P$ is a transition relation
- q_P is a function from $\rightsquigarrow_P$ to $\mathbb{R}^+$ such that for all $s_1 \in \mathcal{S}_P$ and for all $a \in \mathcal{P}_P$, $\sum_{s_2:(s_1,s_2,a) \in \rightsquigarrow_P} q_P(s_1, s_2, a) = 1$.

In the following we denote by $\rightarrow_P$ the relation containing all the tuples of the form (s_1, s_2, a, q) where $(s_1, s_2, a) \in \rightsquigarrow_P$ and $q = q_P(s_1, s_2, a)$. We say that $q_P(s_1, s_2, a) \in \mathbb{R}^+$ is the *rate* of the transition from state s_1 to s_2 with type a if $a \in \mathcal{L}_P$. If a is passive then $q_P(s_1, s_2, a) \in (0, 1]$ denotes the *probability* that the automaton synchronises on type a with a transition from s_1 to s_2. Hereafter, we assume that $q_P(s_1, s_2, a) = 0$ whenever there are no transitions with type a from s_1 to s_2. If $s \in \mathcal{S}_P$, then for all $a \in \mathcal{T}_P$ we write $q_P(s, a) = \sum_{s' \in \mathcal{S}_P} q_P(s, s', a)$. Moreover we define $q_P(s, s') = \sum_{a \in \mathcal{L}_P} q_P(s, s', a)$ and $q_P(s) = \sum_{a \in \mathcal{L}_P} q_P(s, a)$. We say that P is *closed* if $\mathcal{P}_P = \emptyset$. We use the notation $s_1 \overset{a}{\rightsquigarrow}_P s_2$ to denote the tuple $(s_1, s_2, a) \in \rightsquigarrow_P$; we denote by $s_1 \xrightarrow{(a,r)}_P s_2$ (resp., $s_1 \xrightarrow{(a,p)}_P s_2$) the tuple $(s_1, s_2, a, r) \in \rightarrow_P$ (resp., $(s_1, s_2, a, p) \in \rightarrow_P$).

Definition 2: (CTMC underlying a closed SA) The CTMC underlying a closed stochastic automaton P, denoted $X_P(t)$, is defined as the CTMC with state space $\mathcal{S}_P$ and infinitesimal generator matrix $\mathbf{Q}$ defined as: for all $s_1, s_2 \in \mathcal{S}_P$: $q_{s_1,s_2} = \sum_{a,r:(s_1,s_2,a,r) \in \rightarrow_P} r$ with $s_1 \neq s_2$.

We say that a closed automaton P is *ergodic* (*irreducible*) if its underlying CTMC is ergodic (irreducible). We denote the stationary distribution of the CTMC underlying P by $\boldsymbol{\pi}_P$.

The synchronisation operator between two stochastic automata P and Q is defined in the style of the master/slave synchronisation of SANs and the active/passive cooperation used in Markovian process algebra such as PEPA.

Definition 3: (SA synchronisation) Given two stochastic automata P and Q such that $\mathcal{A}_P = \mathcal{P}_Q$ and $\mathcal{A}_Q = \mathcal{P}_P$ we define the automaton $P \otimes Q$ as follows:

$$\frac{s_{p_1} \xrightarrow{(a,r)}_P s_{p_2} \quad s_{q_1} \xrightarrow{(a,p)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(a,pr)}_{P\otimes Q} (s_{p_2}, s_{q_2})} \; (a \in \mathcal{A}_P = \mathcal{P}_Q) \qquad \frac{s_{p_1} \xrightarrow{(a,p)}_P s_{p_2} \quad s_{q_1} \xrightarrow{(a,r)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(a,pr)}_{P\otimes Q} (s_{p_2}, s_{q_2})} \; (a \in \mathcal{P}_P = \mathcal{A}_Q)$$

$$\frac{s_{p_1} \xrightarrow{(\tau,r)}_P s_{p_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(\tau,r)}_{P\otimes Q} (s_{p_2}, s_{q_1})} \qquad \frac{s_{q_1} \xrightarrow{(\tau,r)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(\tau,r)}_{P\otimes Q} (s_{p_1}, s_{q_2})}$$

TABLE I
OPERATIONAL RULES FOR SA SYNCHRONISATION

- $\mathcal{S}_{P\otimes Q} = \mathcal{S}_P \times \mathcal{S}_Q$
- $\mathcal{T}_{P\otimes Q} = \mathcal{A}_{P\otimes Q} \cup \{\tau\}$ where $\mathcal{A}_{P\otimes Q} = \mathcal{A}_P \cup \mathcal{A}_Q = \mathcal{P}_P \cup \mathcal{P}_Q$ and $\mathcal{P}_{P\otimes Q} = \emptyset$
- $\rightsquigarrow_{P\otimes Q}$ and $q_{P\otimes Q}$ are defined according to the rules for $\rightarrow_{P\otimes Q}$ depicted in Table I: indeed, the $\rightarrow_{P\otimes Q}$ contains the tuples $((s_{p_1}, s_{q_1}),(s_{p_2}, s_{q_2}), a, q)$ with $((s_{p_1}, s_{q_1}),(s_{p_2}, s_{q_2}), a) \in \rightsquigarrow_{P\otimes Q}$ and $q = q_{P\otimes Q}((s_{p_1}, s_{q_1}),(s_{p_2}, s_{q_2}), a)$.

This semantics can be easily extended in order to include an arbitrary finite number of cooperating automata (see [19]). The assumption that an automaton obtained by a cooperation does not have passive types, ensures that the resulting automaton has an underlying CTMC and then we can study its stationary distribution.

IV. PROBABILISTIC INPUT/OUTPUT AUTOMATA

In [21], [22], the authors define the class of *probabilistic I/O automata* (PIOA) which provide a model for distributed or concurrent systems that incorporates a notion of probabilistic choice together with a composition operator. This model is based on a combination of reactive and generative transitions. In a reactive system, probabilities are distributed over the outgoing transitions labelled with the same action, i.e., actions are treated as being provided by the environment and there are no probabilistic assumptions about the behavior of the environment. On the other hand, in a generative system, probabilities are distributed over all outgoing transitions, i.e., actions are treated as locally generated by the system.

In a probabilistic I/O automaton for every input action there is a reactive transition. Moreover, it is assumed that each input action is enabled in each state of a PIOA. The output and internal actions (called locally controlled actions) are treated generatively. At most one generative probabilistic transition gives the local behavior of each state. A delay rate parameter δ is also added to each state.

Definition 4: (Probabilistic I/O Automaton (PIOA)) A probabilistic I/O automaton P is a tuple $(\mathcal{S}_P, \mathcal{T}_P, \rightsquigarrow, \mu_P, \delta_P)$ where

- $\mathcal{S}_P$ is a denumerable set of states called *state space* of P
- $\mathcal{T}_P$ is a finite set of actions, partitioned into disjoint sets $\mathcal{O}_P$ of *output* actions, $\mathcal{I}_P$ of *input* actions and the set $\{\tau\}$ of the *unknown* or *internal* action, with the actions in $\mathcal{L}_P = \mathcal{O}_P \cup \{\tau\}$ called *locally controlled*
- $\rightsquigarrow_P \subseteq \mathcal{S}_P \times \mathcal{S}_P \times \mathcal{T}_P$ is a transition relation
- μ_P is the transition probability function from $\rightsquigarrow_P$ to $(0, 1]$ such that
  - for all $s_1 \in \mathcal{S}_P$ and for all $a \in \mathcal{I}_P$, $\sum_{s_2:(s_1,s_2,a)\in\rightsquigarrow_P} \mu_P(s_1, s_2, a)=1$
  - for all $s_1 \in \mathcal{S}_P$, if there exists $a \in \mathcal{L}_P$ and $s_2 \in \mathcal{S}_P$ such that $(s_1, s_2, a) \in\rightsquigarrow_P$, then $\sum_{a\in\mathcal{O}_P\cup\{\tau\}} \sum_{s_2:(s_1,s_2,a)\in\rightsquigarrow_P} \mu_P(s_1, s_2, a)=1$
- δ_P is the state delay function from $\mathcal{S}_P$ to $[0, \infty)$ which is required to satisfy the following condition:
  - for all $s \in \mathcal{S}_P$, $\delta_P(s) > 0$ if and only if there exists $a \in \mathcal{L}_P$ and $s' \in \mathcal{S}_P$ such that $(s, s', a) \in\rightsquigarrow_P$.

We assume that $\mu_P(s_1, s_2, a) = 0$ whenever $(s_1, s_2, a) \not\in\rightsquigarrow_P$. Notice that any PIOA satisfies the following *input-enabledness* condition: for all $s \in \mathcal{S}_P$ and for all $a \in \mathcal{I}_P$, there exists a state $s' \in \mathcal{S}_P$ such that $(s, s', a) \in\rightsquigarrow_P$.

If $s \in \mathcal{S}_P$, then for all $a \in \mathcal{T}_P$ we write $\mu_P(s, a) = \sum_{s'\in\mathcal{S}_P} \mu_P(s, s', a)$. Moreover we define $\mu_P(s, s') = \sum_{a\in\mathcal{T}_P} \mu_P(s, s', a)$ and $\mu_P(s) = \sum_{a\in\mathcal{T}_P} \mu_P(s, a)$.

The state delay function δ_P is explained as follows: upon arrival in a state s, the PIOA P chooses randomly the length of time it will spend in that state before executing its next locally controlled (internal or output) transition. The random choice is made, independently of the other PIOAs in the system, according to an exponential holding time distribution whose mean is the reciprocal $1/\delta_P(s)$ of the delay parameter $\delta_P(s)$ associated with that state. If no locally controlled actions are enabled in this state then $\delta_P(s) = 0$.

In the following we denote by $\rightarrow_P$ the relation containing all the tuples of the form (s_1, s_2, a, μ) where $(s_1, s_2, a) \in\rightsquigarrow_P$ and $\mu = \mu_P(s_1, s_2, a)$. We say that $\mu_P(s, s', a) \in [0, 1]$ is the *transition probability* from state s to s' with type a. We say that P is *closed* if $\mathcal{I}_P = \emptyset$. We use the notation $s_1 \overset{a}{\rightsquigarrow}_P s_2$ to denote the tuple $(s_1, s_2, a) \in\rightsquigarrow_P$; we denote by $s_1 \xrightarrow{(a,\mu)}_P s_2$ the tuple $(s_1, s_2, a, \mu) \in\rightarrow_P$.

Definition 5: (DTMC underlying a closed PIOA) The DTMC underlying a closed PIOA P, denoted $Y_P(t)$, is defined as the DTMC with state space $\mathcal{S}_P$ and transition probability matrix $\mathbf{P}$ defined as: for all $(s_1, s_2) \in \mathcal{S}_P$,

$$p_{s_1,s_2} = \sum_{a,\mu:(s_1,s_2,a,\mu)\in\rightarrow_P} \mu \,.$$

We say that a closed automaton P is *ergodic* (*irreducible*) if

$$\frac{s_{p_1} \xrightarrow{(a,\mu_1)}_P s_{p_2} \quad s_{q_1} \xrightarrow{(a,\mu_2)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(a,\mu_1\mu_2)}_{P\otimes Q} (s_{p_2}, s_{q_2})} \quad (a \in \mathcal{I}_{P\otimes Q})$$

$$\frac{s_{p_1} \xrightarrow{(a,\mu_1)}_P s_{p_2} \quad s_{q_1} \xrightarrow{(a,\mu_2)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(a,\Delta_1\mu_1\mu_2)}_{P\otimes Q} (s_{p_2}, s_{q_2})} \quad (a \in \mathcal{O}_P) \qquad \frac{s_{p_1} \xrightarrow{(\tau,\mu)}_P s_{p_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(\tau,\Delta_1\mu)}_{P\otimes Q} (s_{p_2}, s_{q_1})} \qquad \text{with} \quad \Delta_1 = \frac{\delta_P(s_{p_1})}{\delta_P(s_{p_1}) + \delta_Q(s_{q_1})}$$

$$\frac{s_{p_1} \xrightarrow{(a,\mu_1)}_P s_{p_2} \quad s_{q_1} \xrightarrow{(a,\mu_2)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(a,\Delta_2\mu_1\mu_2)}_{P\otimes Q} (s_{p_2}, s_{q_2})} \quad (a \in \mathcal{O}_Q) \qquad \frac{s_{q_1} \xrightarrow{(\tau,\mu)}_Q s_{q_2}}{(s_{p_1}, s_{q_1}) \xrightarrow{(\tau,\Delta_2\mu)}_{P\otimes Q} (s_{p_1}, s_{q_2})} \qquad \text{with} \quad \Delta_2 = \frac{\delta_Q(s_{q_1})}{\delta_P(s_{p_1}) + \delta_Q(s_{q_1})}$$

TABLE II
OPERATIONAL RULES FOR PIOA SYNCHRONISATION

its underlying CTMC is ergodic (irreducible). We denote the stationary distribution of the CTMC underlying P by $\boldsymbol{\pi}_P$.

Two probabilistic I/O automata P and Q are called *compatible* if the corresponding sets of output actions are disjoint, i.e., $\mathcal{O}_P \cap \mathcal{O}_Q = \emptyset$. In [21], [22] the authors define the composition of a finite collection of compatible PIOAs. The pairwise composition of compatible PIOAs is as follows.

Definition 6: (PIOA Synchronisation) Given two compatible probabilistic I/O automata P and Q we define the automaton $P \otimes Q$ as follows:

- $\mathcal{S}_{P\otimes Q} = \mathcal{S}_P \times \mathcal{S}_Q$
- $\mathcal{T}_{P\otimes Q}$ is partitioned into the disjoint sets $\mathcal{O}_{P\otimes Q} = \mathcal{O}_P \cup \mathcal{O}_Q$, $\mathcal{I}_{P\otimes Q} = (\mathcal{I}_P \cup \mathcal{I}_Q) \setminus \mathcal{O}_{P\otimes Q}$ and $\{\tau\}$
- $\rightsquigarrow_{P\otimes Q}$ and $q_{P\otimes Q}$ are defined according to the rules for $\rightarrow_{P\otimes Q}$ depicted in Table II: indeed, the relation $\rightarrow_{P\otimes Q}$ contains the tuples $((s_{p_1}, s_{q_1}),(s_{p_2}, s_{q_2}), a, \mu)$ with $((s_{p_1}, s_{q_1}),(s_{p_2}, s_{q_2}), a) \in\rightsquigarrow_{P\otimes Q}$ and $\mu = \mu_{P\otimes Q}((s_{p_1}, s_{q_1}), (s_{p_2}, s_{q_2}), a)$
- $\delta_{P\otimes Q} = \delta_P + \delta_Q$, i.e, for all $(s_p, s_q) \in \mathcal{S}_{P\otimes Q}$, $\delta_{P\otimes Q}(s_p, s_q) = \delta_P(s_p) + \delta_Q(s_q)$.

If s is a state of P, then $\delta_P(s)$ is a positive real number corresponding to the *delay rate* in state s. It is the parameter of an exponentially distributed random varible, determining the time that the automaton waits in state s until it generates one of its locally controlled actions. When computing the distribution on locally controlled actions for $P \otimes Q$, the component distributions are joined in one such that any probability of P is multiplied with the normalization factor $\dfrac{\delta_P}{\delta_P + \delta_Q}$ and any probability of Q is multiplied with the normalization factor $\dfrac{\delta_Q}{\delta_P + \delta_Q}$. The normalization factor models a racing policy between the states of P and Q for generating their own locally controlled actions. For instance, the value $\dfrac{\delta_P(s_p)}{\delta_P(s_p) + \delta_Q(s_q)}$ is the probability that the state s_p has less waiting time left then the state s_q and therefore wins the race and generates one of its own local actions.

V. DISCRETIZATION OF A SA INTO A PIOA

In this section we present a method to transform a stochastic automaton P into a probabilistic one P^D. Each state of P^D is equipped with a *delay rate* representing the waiting time between each change of the system. We show that this discretization is indeed a bijection from the class of SAs to the class of PIOAs. Then, we prove that, under the assumption that all the input actions synchnonises with the output ones in the PIOA model, the discretization respects the synchronisation in the sense that, given two sochastic automata, the composition of the corresponding discretized automata coincides with the discretization of the composition of the two stochastic automata.

Definition 7: (Discretization of a SA into a PIOA) Given a stochastic automaton $P = (\mathcal{S}_P, \mathcal{T}_P, \rightsquigarrow_P, q_P)$, the discretization of P is the probabilistic I/O automaton $P^D = (\mathcal{S}_{P^D}, \mathcal{T}_{P^D}, \rightsquigarrow_{P^D}, \mu_{P^D}, \delta_{P^D})$ defined as follows:

- $\mathcal{S}_{P^D} = \mathcal{S}_P$
- $\mathcal{T}_{P^D}$ is partitioned into the disjoint sets $\mathcal{O}_{P^D} = \mathcal{A}_P$, $\mathcal{I}_{P^D} = \mathcal{P}_P$ and $\{\tau\}$
- $\rightsquigarrow_{P^D} = \rightsquigarrow_P$
- $\delta_{P^D}(s) = \sum_{a\in\mathcal{A}_P} q_P(s, a)$ with $s \in \mathcal{S}_P$
- μ_{P^D} is the transition probability function from $\rightsquigarrow_{P^D}$ to $(0, 1]$ such that
  - for all $s_1, s_2 \in \mathcal{S}_P$ and for all $a \in \mathcal{P}_P = \mathcal{I}_{P^D}$, $\mu_{P^D}(s_1, s_2, a) = q_P(s_1, s_2, a)$ (since $q_P(s_1, s_2, a)$ is indeed a probability)
  - for all $s_1, s_2 \in \mathcal{S}_P$ and for all $a \in \mathcal{L}_P = \mathcal{L}_{P^D}$, $\mu_{P^D}(s_1, s_2, a) = q_P(s_1, s_2, a)/\delta_{P^D}(s_1)$.

Notice that $\delta_{P^D}(s)$ is the sum of the transition rates of the active actions of P outgoing from s. If s has only passive actions outgoing from it then $\delta_{P^D}(s) = 0$.

The discretization is a bijective function from the set of all SAs to the set of all PIOAs. The invese of the discretization function allows one to transform a probabilistic I/O automaton P^D into the unique stochastic automaton P having P^D as the corresponding discretization.

Proposition 1: The discretization transformation of Definition 7 is a bijection from the set of all SAs to the set of all PIOAs.

Corollary 1: Let P be a closed SA and P^D be the corresponding discretized PIOA (according to Definition 7). Then

- for all $s \in \mathcal{S}_P$, $\delta_{P^D}(s) = q_P(s)$
- for all $s \in \mathcal{S}_P$ and for all $a \in \mathcal{T}_P$, $q_P(s, s', a) = \mu_P(s, s', a)\, \delta_{P^D}(s)$.

The stationary distribution of a closed SA P can be derived from that of the corresponding discretized atomaton P^D and vice versa.

Theorem 1: Let P be a closed irreducible SA and P^D be the corresponding discretized PIOA (defined according to Definition 7). Let $\mathcal{S} = \mathcal{S}_P = \mathcal{S}_{P^D}$.

- If $\boldsymbol{\pi}_{P^D}$ is an invariant measure of the DTMC underlying P^D then $\boldsymbol{\pi}_P$ defined by
$$\pi_P(s) = \frac{\pi_{P^D}(s)\delta^{-1}_{P^D}(s)}{\sum_{s\in\mathcal{S}} \pi_{P^D}(s)\delta^{-1}_{P^D}(s)} \tag{4}$$
for all $s \in \mathcal{S}$, is the stationary distribution of the CTMC underlying P, assuming its ergodicity.
- If $\boldsymbol{\pi}_P$ is an invariant measure of the CTMC underlying P then $\boldsymbol{\pi}_{P^D}$ defined by
$$\pi_{P^D}(s) = \frac{\pi_P(s)\delta_{P^D}(s)}{\sum_{s\in\mathcal{S}} \pi_P(s)\delta_{P^D}(s)} \tag{5}$$
for all $s \in \mathcal{S}$, is the stationary distribution of the DTMC underlying P^D, assuming its ergodicity.

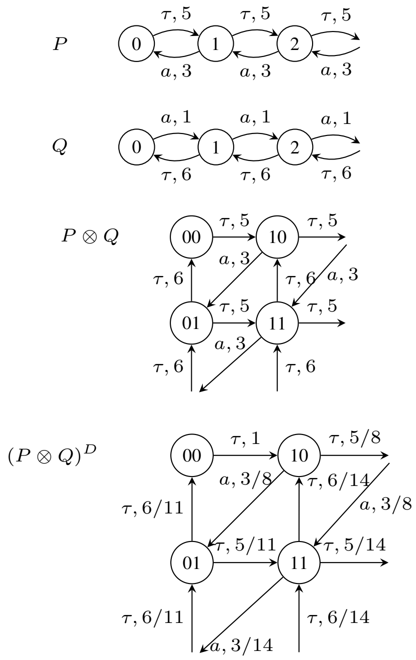
The discretization function respects the synchronisation operator when we assume that all the input actions synchnonise with the output ones in the PIOA model.

Proposition 2 (Discretization respects the synchronisation): Let P and Q be two SAs and P^D and Q^D be the corresponding discretized PIOAs. Assume that $\mathcal{I}_P = \mathcal{O}_Q$ and $\mathcal{I}_Q = \mathcal{O}_P$. Then
$$(P \otimes Q)^D = P^D \otimes Q^D.$$

Example 1: Consider the stochastic automata P and Q depicted in Table III with $\mathcal{A}_P = \mathcal{P}_Q = \{a\}$ and $\mathcal{P}_P = \mathcal{A}_Q = \emptyset$. Let $P \otimes Q$ be the composition of P and Q defined according to Definition 3 and $(P \otimes Q)^D$ be the probabilistic I/O automaton corresponding to the discretization of $P \otimes Q$. The discretizations of P and Q are depicted in Table IV together with their probabilistic composition. One can verify that $(P \otimes Q)^D = P^D \otimes Q^D$.

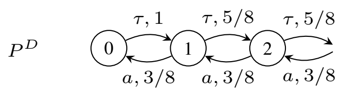
VI. Product-forms for PIOAs

In this section we present our product-form result for probabilistic I/O automata. In particular we prove that the stationary distribution of the composition of two PIOAs, P and Q, can be computed without constructing the stochastic process underlying the whole system $P \otimes Q$, but it can be derived from the the stationary distribution of the two components in isolation.

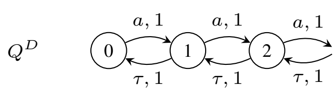


with $\delta(00) = 5$, $\delta(10) = 8$, $\delta(01) = 11$, $\delta(11) = 14$

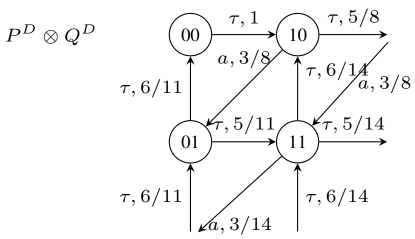
TABLE III
Example of $(P \otimes Q)^D$



with $\delta(0) = 5$, $\delta(1) = 8$, $\delta(2) = 8$



with $\delta(0) = 0$, $\delta(1) = 6$, $\delta(2) = 6$



with $\delta(00) = 5$, $\delta(10) = 8$, $\delta(01) = 11$, $\delta(11) = 14$.

TABLE IV
Example of $P^D \otimes Q^D$

We first introduce a closure operation over probabilistic I/O automata that allows us to assign to all the transitions with the same input action a the same transition probability λ.

Definition 8: (PIOA closure) The closure of a probabilistic I/O automaton P with respect to an input action $a \in \mathcal{I}_P$ and $\lambda \in \mathbb{R}^+$, written $P^C = P\{a \leftarrow \lambda\}$, is the PIOA defined as:

- $\mathcal{S}_{P^C} = \mathcal{S}_P$
- $\mathcal{T}_P$ is partitioned into the disjoint sets $\mathcal{O}_{P^C} = \mathcal{O}_P \cup \{a\}$, $\mathcal{I}_{P^C} = \mathcal{I}_P \setminus \{a\}$ and $\{\tau\}$
- $\rightsquigarrow_{P^C} = \rightsquigarrow_P$
- μ_{P^C} is the transition probability function from $\rightsquigarrow_{P^C}$ to $(0, 1]$ such that
  - for all $s_1, s_2 \in \mathcal{S}_P$ such that $(s_1, s_2, a) \in \rightsquigarrow_P$, $\mu_{P^C}(s_1, s_2, a) = \mu_P(s_1, s_2, a)\frac{\lambda}{\delta_P(s_1)+\lambda}$
  - for all $s_1, s_2 \in \mathcal{S}_P$ and for all $b \in \mathcal{I}_P \setminus \{a\}$, $\mu_{P^C}(s_1, s_2, b) = \mu_P(s_1, s_2, b)$
  - for all $s_1, s_2 \in \mathcal{S}_P$ and for all $b \in \mathcal{L}_P$, $\mu_{P^C}(s_1, s_2, b) = \mu_P(s_1, s_2, b)\frac{\delta_P(s_1)}{\delta_P(s_1)+\lambda}$
- δ_{P^C} is such that for all $s \in \mathcal{S}_P$, $\delta_{P^C}(s) = \delta_P(s) + \lambda$.

Several closures can be specified by susequently applying Definition 8. It is easy to prove that the order in which the closures are applied is irrelevant and then, if $\mathcal{I}_P = \{a_1, \ldots, a_n\}$ and $\{\lambda_1, \ldots, \lambda_n\}$ is a set of positive real numbers then we write $P\{a_i \leftarrow \lambda_i\}_{a_i \in \mathcal{I}_P}$ for $((P\{a_1 \leftarrow \lambda_1\}) \cdots)\{a_n \leftarrow \lambda_n\}$.

We are now ready to prove the product-form theorem.

Theorem 2: (Product-forms for PIOA) Let P and Q be two probabilistic I/O automata such that $\mathcal{O}_P = \mathcal{I}_Q$ and $\mathcal{O}_Q = \mathcal{I}_P$. Let $\{a_1, \ldots, a_n\} = \mathcal{O}_P \cup \mathcal{O}_Q$. If there exists a set of positive real numbers $\{\lambda_1, \ldots, \lambda_n\}$ such that $P^C = P\{a_i \leftarrow \lambda_i\}_{a_i \in \mathcal{I}_P}$ and $Q^C = Q\{a_i \leftarrow \lambda_i\}_{a_i \in \mathcal{I}_Q}$ satisfy the following equations:

- for all $s_p \in \mathcal{S}_P$ and for all $a_i \in \mathcal{O}_P$,

$$\delta_{P^C}(s_p) \sum_{s'_p \in \mathcal{S}_{P^C}} \mu_{P^C}(s'_p, s_p, a_i) \frac{\pi_{P^C}(s'_p)}{\pi_{P^C}(s_p)} \frac{\delta_P(s'_p)}{\delta_{P^C}(s'_p)} = \lambda_i \qquad (6)$$

- for all $s_q \in \mathcal{S}_P$ and for all $a_i \in \mathcal{O}_P$,

$$\delta_{Q^C}(s_q) \sum_{s' \in \mathcal{S}_{Q^C}} \mu_{Q^C}(s'_q, s_q, a_i) \frac{\pi_{Q^C}(s'_q)}{\pi_{Q^C}(s_q)} \frac{\delta_Q(s'_q)}{\delta_{Q^C}(s'_q)} = \lambda_i \qquad (7)$$

then for all the states $(s_p, s_q) \in \mathcal{S}_{P \otimes Q}$ belonging to an irreducible class:

$$\pi_{P \otimes Q}(s_p, s_q) \propto \pi_{P^C}(s_p)\pi_{Q^C}(s_q) \frac{\delta_P(s_p) + \delta_Q(s_q)}{\delta_{P^C}(s_p)\delta_{Q^C}(s_q)}\,. \qquad (8)$$

VII. Conclusion

This paper has addressed the problem of the compositional stationary analysis of PIOAs in a similar fashion of what has been done for the transient in [21]. We have derived a product-form theorem for PIOA. Since we have enlighed the strong relations between the stochastic automata in the style of SANs and the PIOAs, it is interestign to address the problem of relating Theorem 2 with the results known for the stochastic counterpart. It is important to notice that the product-form that appears in our theorem is the solution of a DTMC and is *not* equal to that of the corresponding SAN. In fact the discretisation procedure described in Section V does not preserve the steady-state distribution of automata. Moreover, given a PIOA, its steady-state distribution is independent of the δs associated with the states (in the same way the stationary distribution of the embedded chain of a CTMC is independent of its residence times). As a consequence if a PIOA is in product-form, we can easily compute the stationary distributions of all the corresponding stochastic automata which are defined for arbitrary definitions of the δ functions.

References

[1] S. Balsamo, P. G. Harrison, and A. Marin. Methodological Construction of Product-form Stochastic Petri-Nets for Performance Evaluation. *Journal of System and Software*, 85(7):1520–1539, 2012.

[2] E. Barbierato, G.-L. Dei Rossi, M. Gribaudo, M. Iacono, and A. Marin. Exploiting product forms solution techniques in multiformalism modeling. *Electr. Notes Theor. Comput. Sci.*, 296:61–77, 2013.

[3] E. Bartocci, F. Corradini, M. R. Di Berardini, E. Entcheva, S. A. Smolka, and R. Grosu. Modeling and simulation of cardiac tissue using hybrid I/O automata. *Theoretical Compututer Science*, 410(33-34):3149–3165, 2009.

[4] M. Bugliesi, L. Gallina, S. Hamadou, A. Marin, and S. Rossi. Behavioural equivalences and interference metrics for mobile ad-hoc networks. *Performance Evaluation*, 73:41–72, 2014.

[5] X. Chao, M. Miyazawa, and M. Pinedo. *Queueing networks: customers, signals and product form solutions*. Wiley, 1999.

[6] J.-M. Fourneau. Discrete time Markov chains competing over resources: Product form steady-state distribution. In *Proc of the 5th International Conference on the Quantitative Evaluaiton of Systems (QEST'08)*, pages 147–156, 2008.

[7] J.-M. Fourneau. Collaboration of discrete-time Markov chains: Tensor and product form. *Performance Evaluation*, 67(9):779–796, 2010.

[8] L. Gallina, S. Hamadou, A. Marin, and S. Rossi. A probabilistic energy-aware model for mobile ad-hoc networks. In *Proc. of ASMTA'11*, volume 6751 of *LNCS*, pages 316–330, 2011.

[9] E. Gelenbe. Random neural networks with negative and positive signals and product form solution. *Neural Computation*, 1(4):502–510, 1989.

[10] P. G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, 2003.

[11] P. G. Harrison and A. Marin. Product-forms in multi-way synchronizations. *Computer Journal*, 57(11):1693–1710, 2014.

[12] H. Hermanns, U. Herzog, and J. P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002.

[13] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge Press, 1996.

[14] J. Hillston, A. Marin, C. Piazza, and S. Rossi. Contextual lumpability. In *Proc. of VALUETOOLS'13*, pages 194–203, 2013.

[15] E. Gelenbe J.M. Fourneau. Random neural networks with multiple classes of signals. *Neural Computation*, 11(4):953–963, 1999.

[16] F. Kelly. *Reversibility and stochastic networks*. Wiley, New York, 1979.

[17] A. Marin. Product-form in G-networks. *Probability in the Engineering and Informational Sciences*, 2016. Available online.

[18] A. Marin and S. Rossi. Lumping-based equivalences in Markovian automata and applications to product-form analyses. In *Proc. of the 12th International Conference on Quantitative Evaluation of SysTems (QEST'15)*, volume 9259 of *LNCS*, pages 160–175, 2015.

[19] A. Marin and S. Rossi. Quantitative analysis of concurrent reversible computations. In *Proc. of FORMATS'15*, volume 9268 of *LNCS*, pages 206–221, 2015.

[20] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Performance Evaluation Review*, 13(2):147–154, 1985.

[21] E.W. Stark and S.A. Smolka. Compositional analysis of expected delays in networks of probabilistic I/O automata. In *Proc. of the 13th Annual IEEE Symposium on Logic in Computer Science (LICS'98)*, pages 466–477, 1998.

[22] S.H. Wu, S.A. Smolka, and E.W. Stark. Composition and behaviors of probabilistic I/O automata. *Theoretical Computer Science*, 176:1–38, 1997.