

Quantitative Analysis of Concurrent Reversible Computations

Andrea Marin¹ and Sabina Rossi¹

DAIS, Università Ca' Foscari Venezia, Italy
Via Torino, 155
{marin,srossi}@dais.unive.it

Abstract. Reversible computing is a paradigm of computation that extends the standard forward-only programming to reversible programming, so that programs can be executed both in the standard, forward direction, and backward, going back to past states. In this paper we present novel quantitative stochastic models for concurrent and cooperating reversible computations. More precisely, we introduce the class of ρ -reversible stochastic automata and define a semantics for the synchronization ensuring that this class of models is closed under composition. For this class of automata we give an efficient way of deriving the equilibrium distribution. Moreover, we prove that the equilibrium distribution of the composition of reversible automata can be derived as the product of the equilibrium distributions of each automaton in isolation.

1 Introduction

Reversible computing is a paradigm of computation which relies on the idea that programs can be executed both in the standard, forward direction, and backward. In contrast to traditional forward-only computations, reversible executions may restore a past state by undoing, one by one, all the previously performed operations. According to [26] a *bi-directional* execution is any program execution that carries with it the notion of a runtime choice between forward and backward execution, regardless of the granularity of the execution unit. Although still not widely used, reversible computing has a growing number of promising applications. For instance, it has been shown in [4, 5] that ideally reversible computations can be performed without loss of energy. Another application scenario is the improvement of the performance in parallel computations. Indeed, by assuming the reversibility of the computations we may increase the concurrency of the systems by allowing the local processors to execute their jobs asynchronously. In case of data dependence violations, the backward execution rolls back the processor to the execution point where the dependency violation occurred. A practical application of this idea is the popular “time warp” mechanism proposed by Jefferson in [17, 18]. Other applications include processors with speculative executions [11], debugging [8, 22], fault detection [7] and tolerance [27], database transaction rollbacks and quantum computing [25, 30] (see [26] for a survey of these application scenarios). Reversible computing can

be implemented in essentially two ways: the first consists in recording a set of checkpoints that store the state of the processor at some epochs of the computation, the second in implementing fully reversible programs where each step of the computation may be inverted. *Janus* [34, 33] is an example of a time-reversible programming language and many research efforts have been devoted to the construction of reversible hardware components.

While the functional analysis of reversible computations has been widely explored in previous works (see [4, 28, 9, 20]), the quantitative analysis is a topic that has still to be addressed. For forward-only computations, time-based quantitative analysis has been deeply studied especially in the context of systems in which some aspects are abstracted out and assumed to have a probabilistic nature (see, e.g., [2]).

In this paper we focus on the problem of defining quantitative stochastic models for concurrent and cooperating reversible computations. The stochastic processes underlying our models are continuous-time Markov processes with a discrete-state space (i.e., Continuous Time Markov Chains, CTMCs) which is a common framework for the formal specification and evaluation of quantitative properties of systems [29, 15, 13]. We focus on the derivation of the equilibrium (or steady-state) performance indices, i.e., we aim at computing the probability of observing the system in a certain state when the time elapsed since the beginning of the computation is long enough (ideally infinite). Indeed, we can imagine to repeat the computation infinite times, and the equilibrium distribution represents the probability of observing a certain state under this assumption. For models with an underlying CTMC the necessary and sufficient condition for the existence of a unique equilibrium distribution is that the chain is ergodic, i.e., its reachability graph is irreducible and the expected time elapsed from a visit to any state until the next visit to the same state is finite.

Contribution. In this paper we use stochastic automata in the style of [29] to model reversible processes. We introduce the class of ρ -reversible stochastic automata and define a semantics for the synchronisation of reversible automata that ensures that this class of models is closed under composition. For simplicity, we introduce the synchronisation semantics for pairs of automata and then we discuss how it is possible to extend it to an arbitrary number of automata. We also address the problem of the computation of the equilibrium distribution. Indeed, for general Markovian models, the derivation of the equilibrium distribution is known to be time expensive ($\mathcal{O}(n^3)$ with n being the number of model states) and prone to numerical stability problems [31] since it requires the numerical solution of a linear system of equations. For the class of reversible automata we give an efficient way of deriving the steady-state distribution which is also numerically stable since it involves only the product of floating point numbers. Even more interestingly, we prove that the equilibrium distribution of the composition of reversible automata can be derived as the product of the equilibrium distributions of each automaton in isolation. In the literature, this property is known as *product-form* [19, 3].

Related work. Formalisms for the description and the analysis of reversible computations have been proposed in [10, 21] and the references therein. The behavioural analysis of reversible computations has been studied in [28, 9, 20] where the authors address the problem of reachability and system equivalences via bisimulation. In [1] and subsequent papers, the authors propose a quantitative evaluation of the energy costs required to allow the system to reach a steady-state behaviour. Our contribution is more related to formal methods for the quantitative analysis of reversible computations and the computation of the equilibrium distribution that, to the best of our knowledge, has still to be explored. From a theoretical point of view, time-reversibility in CTMCs is mainly studied in [19] where the author introduces also a class of product-form models. In [23, 24] the authors introduce novel reversibility-based definitions for Markov chains but, differently from the present work, there is not any notion of compositionality. With respect to the above mentioned works, here we consider interacting labelled automata representing reversible computations and define a synchronisation semantics that is closed with respect to this class of computations. Moreover, we study their equilibrium distribution which, surprisingly, is proved to exhibit an unconditional product-form. This is different from the well-known quasi-reversibility based product-forms studied in [19, 12].

Plan of the paper. The paper is organized as follows. Section 2 introduces the fundamental notions of Markov chain and reversibility. In Section 3 we introduce the definition of stochastic automaton and provide the synchronization semantics. In Section 4 we present our main theoretical results about reversible automata, their closure under synchronization and their product-form solution. Section 5 concludes the paper.

2 Continuous-Time Markov Chains

Let $X(t)$ be a stochastic process taking values into a state space \mathcal{S} for $t \in \mathbb{R}^+$. $X(t)$ is *stationary* if $(X(t_1), X(t_2), \dots, X(t_n))$ has the same distribution as $(X(t_1 + \tau), X(t_2 + \tau), \dots, X(t_n + \tau))$ for all $t_1, t_2, \dots, t_n, \tau \in \mathbb{R}^+$. $X(t)$ satisfies the *Markov property* and is called *Markov process* if the conditional (on both past and present states) probability distribution of its future behaviour is independent of its past evolution until the present state.

A Continuous-Time Markov Chain (CTMC) is a Markov process with a discrete state space. A CTMC $X(t)$ is *time homogeneous* if the conditional probability $P(X(t + \tau) = s \mid X(t) = s')$ does not depend upon t , and is *irreducible* if every state in \mathcal{S} can be reached from every other state. A state in a CTMC is called *recurrent* if the probability that the process will eventually return to the same state is one. A recurrent state is called *positive-recurrent* if the expected number of steps until the process returns to it is finite. A CTMC is *ergodic* if it is irreducible and all its states are positive-recurrent. We assume that any CTMC which we deal with is *ergodic*. A process satisfying all these assumptions possesses an *equilibrium* (or *steady-state*) *distribution*, that is the *unique* collection of pos-

itive numbers $\pi(s)$ with $s \in \mathcal{S}$ such that $\lim_{t \rightarrow \infty} P(X(t) = s \mid X(0) = s') = \pi(s)$ for all $s' \in \mathcal{S}$.

The transition rate between two states s and s' is denoted by $q(s, s')$, with $s \neq s'$. The infinitesimal generator matrix \mathbf{Q} of a Markov process is such that the $q(s, s')$'s are the off-diagonal elements while the diagonal elements are formed as the negative sum of the non-diagonal elements of each row. The equilibrium distribution $\boldsymbol{\pi}$ is the unique row vector of positive numbers $\pi(s)$ with $s \in \mathcal{S}$, summing to unit and satisfying the system of global balance equations (GBEs):

$$\boldsymbol{\pi} \mathbf{Q} = \mathbf{0}. \quad (1)$$

The solution of system (1) is often unfeasible due to the large number of states of the CTMC. The analysis of an ergodic CTMC in equilibrium can be greatly simplified if it satisfies the property that when the direction of time is reversed the behaviour of the process remains the same.

Given a stationary CTMC, $X(t)$ with $t \in \mathbb{R}^+$, we call $X(\tau - t)$ its reversed process. We denote by $X^R(t)$ the reversed process of $X(t)$. It can be shown that $X^R(t)$ is also a stationary CTMC. We say that $X(t)$ is *reversible* if it is stochastically identical to $X^R(t)$, i.e., the process $(X(t_1), \dots, X(t_n))$ has the same distribution as $(X(\tau - t_1), \dots, X(\tau - t_n))$ for all $t_1, \dots, t_n, \tau \in \mathbb{R}^+$ [19].

For a stationary Markov process there exists a necessary and sufficient condition for reversibility expressed in terms of the equilibrium distribution $\boldsymbol{\pi}$ and the transition rates.

Proposition 1. (Transition rates of reversible processes [19]) *A stationary CTMC with state space \mathcal{S} and infinitesimal generator \mathbf{Q} is reversible if and only if for all $s, s' \in \mathcal{S}$ with $s \neq s'$,*

$$\pi(s)q(s, s') = \pi(s')q(s', s).$$

A reversible CTMC $X(t)$ and its dual $X^R(t)$ have the same equilibrium distribution.

The *reversed process* $X^R(t)$ of a Markov process $X(t)$ can always be defined even when $X(t)$ is not reversible. In [12] the author shows that $X^R(t)$ is a CTMC and its transition rates are defined according to the following proposition.

Proposition 2. (Transition rates of reversed process [12]) *Given the stationary CTMC $X(t)$ with state space \mathcal{S} and infinitesimal generator \mathbf{Q} , the transition rates of the reversed process $X^R(t)$, forming its infinitesimal generator \mathbf{Q}^R , are defined as follows: for all $s, s' \in \mathcal{S}$,*

$$q^R(s', s) = \frac{\pi(s)}{\pi(s')} q(s, s'), \quad (2)$$

where $q^R(s', s)$ denotes the transition rate from s' to s in the reversed process.

The equilibrium distribution $\boldsymbol{\pi}$ is the same for both the forward and the reversed process.

3 Stochastic Automata

Many high-level specification languages for stochastic discrete-event systems are based on *Markovian process algebras* [15, 6, 14] that naturally supply powerful composition operators and timed actions whose delay is governed by independent random variables with a continuous-time exponential distribution. The expressivity of such languages allows the development of well-structured specifications and efficient analyses of both qualitative and quantitative properties in a single framework. Their semantics is given in terms of stochastic automata, an extension of labelled automata with clocks that are exponentially distributed random variables.

In this paper we consider stochastic concurrent automata with an underlying continuous-time Markov chain as common denominator of a wide class of Markovian stochastic process algebra. Stochastic automata are equipped with a *composition operator* which allows a complex automaton to be constructed from simpler components. Our model draws a distinction between *active* and *passive* action types, and in forming the composition of automata only active/passive synchronisations are permitted.

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

- \mathcal{S}_P is a denumerable set of states called state space of P
- \mathcal{A}_P is a finite set of active types
- \mathcal{P}_P is a finite set of passive types
- τ denotes the unknown type
- $\rightsquigarrow_P \subseteq (\mathcal{S}_P \times \mathcal{S}_P \times \mathcal{T}_P)$ is a transition relation where $\mathcal{T}_P = (\mathcal{A}_P \cup \mathcal{P}_P \cup \{\tau\})$ and for all $s \in \mathcal{S}_P$, $(s, s, \tau) \notin \rightsquigarrow_P$ ¹
- q_P is a function from \rightsquigarrow_P to \mathbb{R}^+ such that $\forall s_1 \in \mathcal{S}_P$ and $\forall a \in \mathcal{P}_P$, $\sum_{s_2: (s_1, s_2, a) \in \rightsquigarrow_P} q_P(s_1, s_2, a) \leq 1$.

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 $q_P(s, s', a) \in \mathbb{R}^+$ is the *rate* of the transition from state s to s' with type a if $a \in \mathcal{A}_P \cup \{\tau\}$. Notice that this is indeed the apparent transition rate from s to s' relative to a [15]. If a is passive then $q_P(s, s', a) \in (0, 1]$ denotes the *probability* that the automaton synchronises on type a with a transition from s to s' . Hereafter, we assume that $q_P(s, s', a) = 0$ whenever there are no transitions with type a from s to s' . If $s \in \mathcal{S}_P$, then for all $a \in \mathcal{T}_P$ we write $q_P(s, a) = \sum_{s' \in \mathcal{S}_P} q_P(s, s', a)$. We say that P is *closed* if $\mathcal{P}_P = \emptyset$. We use the notation $s_1 \xrightarrow{a}_P s_2$ 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., $(s_1, s_2, a, p) \in \rightarrow_P$).

¹ Notice that τ self-loops do not affect the equilibrium distribution of the CTMC underlying the automaton. Moreover, the choice of excluding τ self-loops will simplify the definition of automata synchronisation.

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

Table 1: Operational rules for SA synchronisation

Definition 2. (CTMC underlying a closed automaton) *The CTMC underlying a closed 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 $s_1 \neq s_2 \in \mathcal{S}_P$,*

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

We say that a closed automaton P is ergodic (irreducible) if its underlying CTMC is ergodic (irreducible). We denote the equilibrium distribution of the CTMC underlying P by π_P .

The synchronisation operator between two stochastic automata P and Q is defined in the style of master/slave synchronisation of SANs [29] based on the Kronecker's algebra and the active/passive cooperation used in Markovian process algebra such as PEPA [15, 16].

Definition 3. (SA synchronisation) *Given two 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:*

- $\mathcal{S}_{P \otimes Q} = \mathcal{S}_P \times \mathcal{S}_Q$
- $\mathcal{A}_{P \otimes Q} = \mathcal{A}_P \cup \mathcal{A}_Q = \mathcal{P}_P \cup \mathcal{P}_Q$
- $\mathcal{P}_{P \otimes Q} = \emptyset$
- τ is the unknown type
- $\sim_{P \otimes Q}$ and $q_{P \otimes Q}$ are defined according to the rules for $\rightarrow_{P \otimes Q}$ depicted in Table 1 where $\rightarrow_{P \otimes Q}$ contains the tuples $((s_{p_1}, s_{q_1}), (s_{p_1}, s_{q_2}), a, q)$ with $((s_{p_1}, s_{q_1}), (s_{p_1}, s_{q_2}), a) \in \sim_{P \otimes Q}$ and $q = q_{P \otimes Q}((s_{p_1}, s_{q_1}), (s_{p_1}, s_{q_2}), a)$.

Notice that although we define a semantics for pairwise SA synchronisations, this can be easily extended in order to include an arbitrary finite number of pairwise cooperating automata as discussed in Section 4.1. We point out that 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 equilibrium distribution.

4 Reversible Stochastic Automata

In this section we introduce the notion of ρ -reversibility for stochastic automata. This is defined in the style of the Kolmogorov's criteria presented in [19]. We assume that for each forward action type a there is a corresponding backward type \bar{a} with $\bar{\bar{a}} = a$. Formally $\bar{\cdot}$ is a bijection (renaming) from \mathcal{T}_P to \mathcal{T}_P . In most of practical cases, $\bar{\cdot}$ is an involution, i.e., $\bar{\bar{a}} = a$ for all $a \in \mathcal{T}_P$, and hence the semantics becomes similar to the one proposed in [10]. We say that $\bar{\cdot}$ respects the active/passive types of an automaton P if $\bar{\tau} = \tau$ and for all $a \in \mathcal{T}_P \setminus \{\tau\}$ we have that $a \in \mathcal{A}_P \Leftrightarrow \bar{a} \in \mathcal{A}_P$ (or equivalently $a \in \mathcal{P}_P \Leftrightarrow \bar{a} \in \mathcal{P}_P$).

Definition 4. (ρ -reversible automaton) *Let P be an irreducible stochastic automaton, then P is ρ -reversible if $\rho : \mathcal{S}_P \rightarrow \mathcal{S}_P$ is a renaming (permutation) of the states and $\bar{\cdot}$ is a bijection from \mathcal{T}_P to \mathcal{T}_P that respects the active/passive typing, such that:*

1. $q(s, a) = q(\rho(s), a)$, for each state $s \in \mathcal{S}_P$;
2. for each cycle $\Psi = (s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} s_n \xrightarrow{a_n} s_1)$ in P there exists one cycle $\bar{\Psi} = (\rho(s_1) \xrightarrow{\bar{a}_n} \rho(s_n) \xrightarrow{\bar{a}_{n-1}} \dots \xrightarrow{\bar{a}_2} \rho(s_2) \xrightarrow{\bar{a}_1} \rho(s_1))$ in P such that:

$$\prod_{i=1}^n q(s_i, s_{i+1}, a_i) = \prod_{i=1}^n q(\rho(s_{i+1}), \rho(s_i), \bar{a}_i) \text{ with } s_{n+1} \equiv s_1.$$

We say that $\bar{\Psi}$ is the inverse of cycle Ψ . If ρ is the identity function we simply say that P is reversible.

Observe that the cycle $\bar{\Psi}$ is unique. This follows from the fact that, by Definition 1 of stochastic automaton, there exists at most one transition between any pair of states with a certain type $a \in \mathcal{T}_P$. We stress on the fact that this class of models does not belong to any well-known class of compositional reversible models including those studied in [19, 12].

Example 1. (Reversible Random Number Generators) In the context of concurrent simulations, the Time Warp mechanism [17, 18] is an optimistic synchronisation protocol which is used to synchronize parallel discrete event simulations allowing a process to roll back when an event occurring in another process invalidates part of its computation. The state of the process depends on a set of random numbers that are generated by a pseudo Random Number Generator (RNG). Reversible RNGs allow their state to move both forward and backward so that if the simulation process is itself reversible, the rollback can be performed without storing the process state in memory [26]. We can model a reversible RNG by a simple sequential reversible automaton as shown in Fig. 1 (a). The model consists of a denumerable set of states s_1, s_2, \dots . From each state s_i a random number is generated causing a synchronising label a_i and a transition to state s_{i+1} with rate r_f . In the backward flow, the transition occurs from state s_{i+1} to s_i with type b_i . Although it must be the case that the pseudo random number

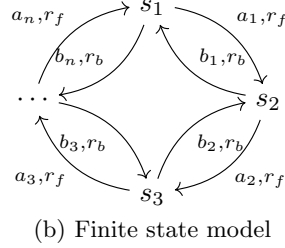
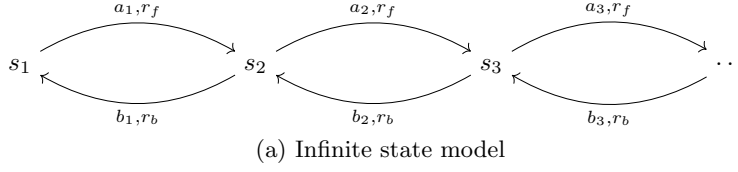


Fig. 1: Models for RNGs

generated by type a_i is the same generated in the backward transition typed b_i , the use of two different types allows the synchronising process to drive the RNG in the forward or in the backward direction. Conversely, the use of the same type would have led to a situation in which the simulator process is executing in the forward direction while the RNG process draws pseudo random numbers in the backward direction, hence introducing an undesired behaviour. The automaton is ρ -reversible with ρ being the identity function and $\bar{a}_i = b_i$, $\bar{b}_i = a_i$.

Let us now consider a cyclic RNG in which, in the forward computation, the state that follows s_n is s_1 for a given $n > 2$ as depicted in Fig. 1 (b). In this case, we have also to consider the cycle $s_1 \xrightarrow{a_1, r_f} s_2 \cdots \xrightarrow{a_{n-1}, r_f} s_n \xrightarrow{a_n, r_f} s_1$ whose inverse is $s_1 \xrightarrow{b_n, r_b} s_n \cdots \xrightarrow{b_2, r_b} s_2 \xrightarrow{b_1, r_b} s_1$. In order for the rate condition of Definition 4 to be satisfied, equation $r_f^n = r_b^n$ must hold, i.e., $r_f = r_b$ is required. In other words, the automaton is reversible if the rate of generation of random numbers is the same in the forward and backward flow.

Example 2. (Reversible computations with checkpoints) Traditional means for restoring a computation to a previous state involve checkpoints, that are fixed conditions such that when a checkpoint is reached the computation may decide to proceed forward to the next checkpoint or backward to the previous one. In these cases, differently from the model studied in Example 1, the decision about moving forward or backward is not taken in each model state but only at the fixed checkpoints. Fig. 2 shows the stochastic automaton underlying such computations. From checkpoint CK_1 the computation proceeds forward to states s_1, s_2, \dots, s_n and then reaches checkpoint CK_2 . At checkpoint CK_2 the computation can move backward to s'_n, \dots, s'_2, s'_1 and then to CK_1 . We can show that the computation is ρ -reversible with $\rho(CK_i) = CK_i$, $\rho(s_i) = s'_i$, $\rho(s'_i) = \rho(s_i)$

$$\begin{aligned}
& - \mathcal{S}_S = \mathcal{S}_P \\
& - \mathcal{A}_S = \mathcal{A}_P \text{ and } \mathcal{P}_S = \mathcal{P}_P \text{ if } a \in \mathcal{A}_P \cup \{\tau\} \\
& - \mathcal{A}_S = \mathcal{A}_P \cup \{a\} \text{ and } \mathcal{P}_S = \mathcal{P}_P \setminus \{a\} \text{ if } a \in \mathcal{P}_P \\
& - \rightsquigarrow_S = \rightsquigarrow_P \\
& - q_S(s_1, s_2, b) = \begin{cases} q_P(s_1, s_2, b) & \text{if } b \neq a \\ q_P(s_1, s_2, b) \cdot k & \text{if } b = a \end{cases}
\end{aligned}$$

Intuitively, the rescaling of a passive type a with a factor k_a should be interpreted as if the automaton is synchronising with an event that occurs according to an independent homogeneous Poisson process with rate k_a and hence can be seen as a way to close an open automaton. The rescaling of an active type b by a factor $k_b \leq 1$ should be interpreted as the reduction of the rates of the transitions with type b due to the fact that a cooperating automaton is ready to synchronise on b with a state independent probability k_b . If $k_b > 1$ we interpret this as a speed up of the active transitions, e.g., because the synchronising automaton models the fact that one component performs part of the work that is associated with the synchronising transition.

Since $\bar{\cdot}$ is a permutation of the labels, we denote by $[a]$ the orbit of type a , i.e., $a \in [a]$, $\bar{a} \in [a]$, $\bar{\bar{a}} \in [a]$ and so on. When $\bar{\cdot}$ is an involution then for all a we have that $[a]$ is either a singleton or contains two types. Notice that if $\bar{\cdot}$ respects the active/passive types of P , then for all a we have that the elements of $[a]$ are either all active or all passive. Notice that $[\tau] = \{\tau\}$.

Proposition 4. (ρ -reversible scaled automaton) *If P is an ergodic ρ -reversible automaton, then for all $a \in \mathcal{T}_P$, the automaton $P' = P\{b \cdot k, b \in [a]\}$ is also ρ -reversible. Moreover, $\pi_P(s) = \pi_{P'}(s)$ for all states $s \in \mathcal{S}_P$.*

According to Proposition 4 the ergodicity and the equilibrium distribution of a ρ -reversible automaton does not depend on the rescaling of all the types belonging to an orbit of $\bar{\cdot}$. As a consequence, if the automaton is open and we close it by rescaling, its equilibrium distribution and ergodicity does not change with the rescaling factor. Henceforth, we will talk about equilibrium distribution and ergodicity of open automata in the sense that they are the same for any closure obtained by rescaling.

The following relationship states that ρ -reversibility for stochastic automata implies the reversibility of the underlying CTMC when the renaming function ρ is the identity.

Proposition 5. (CTMC-reversibility) *Let P be an ergodic automaton. If P is ρ -reversible and ρ is the identity function then its underlying CTMC is reversible.*

The opposite is, in general, not true.

Example 3. (Automaton ρ -reversibility and CTMC reversibility) Consider the automaton depicted in Fig. 3 (a). Since it has only one active type a (indeed, in this case, $\bar{a} = a$), the underlying CTMC can be trivially derived and it is identical to the model in Fig. 3 (a) where the action type a is not present. Since the

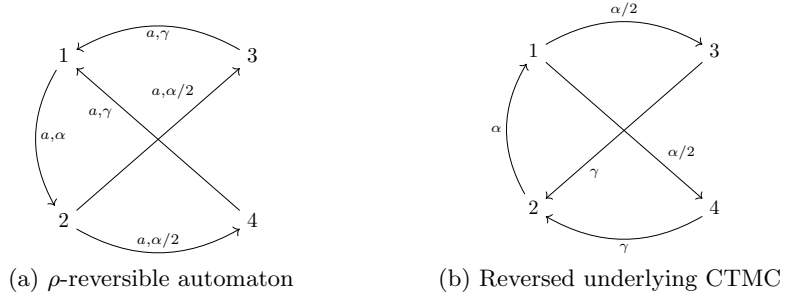


Fig. 3: A ρ -reversible automaton with an underlying non-reversible CTMC.

CTMC underlying the automaton is ergodic, we may construct the time-reversed CTMC displayed in Fig. 3 (b). We notice that the two CTMCs are different and hence the automaton is not reversible, however we can prove that under the renaming $\rho = \{1 \rightarrow 2, 2 \rightarrow 1, 3 \rightarrow 4, 4 \rightarrow 3\}$ the labelled automaton in Fig. 3 (a) satisfies the conditions of Definition 4, i.e., it is ρ -reversible.

The following theorems are important to tackle the state space explosion when studying a network of synchronising automata. Theorem 2 states that the synchronisation of ρ -reversible automata is still ρ -reversible and therefore networks of more than two automata can be defined by combining pairs of automata. Notice that operator \otimes among ρ -reversible automata inherits the associativity from Kronecker's operator of the Stochastic Automata [29] or from the synchronisation operator of PEPA [15]. Theorem 3 states that the composition of two ρ -reversible automata has an equilibrium distribution that can be derived by the analysis of the isolated cooperating automata (i.e., without generating the joint state space and solving the system of global balance equations). Notice that this analysis, differently from those based on the concepts of quasi-reversibility [19, 12] and reversibility, does not require a re-parameterisation of the cooperating automata, i.e., the expressions of the equilibrium distributions of the isolated automata are *as if* their behaviours are stochastically independent although they are clearly not.

Theorem 2. (Closure under ρ -reversibility) *Let P and Q be two ρ_P - and ρ_Q -reversible automata with respect to the same function τ on the action types. Then, the automaton $P \otimes Q$ is $\rho_{P \otimes Q}$ -reversible with respect to the same τ , where, for all $(s_1, s_2) \in \mathcal{S}_P \times \mathcal{S}_Q$,*

$$\rho_{P \otimes Q}(s_p, s_q) = (\rho_P(s_p), \rho_Q(s_q)). \quad (3)$$

Theorem 3 plays a pivotal role in the theory developed in this paper. Indeed, if we have a set of M cooperating automata the cardinality of the state space may have the size of the Cartesian product of the state space of each single automaton. Assuming that each automaton has a finite state space of cardinality N , the joint

state space has, in the worst case, a cardinality of N^M . Since the computation of the equilibrium distribution of a CTMC requires the solution of the linear system of global balance equations, its complexity is $\mathcal{O}(N^{3M})$. In case of ρ -reversible automata, the steady-state distribution can be computed efficiently by means of Theorem 1 in linear time on the cardinality of the state space for each automaton, and hence by Theorem 3 the complexity of the computation of the joint equilibrium distribution is $\mathcal{O}(NM)$.

Theorem 3. (Product-form solution) *Let P and Q be two ergodic ρ_P - and ρ_Q -reversible automata with respect to the same function $\bar{\cdot}$ on the action types, and let π_P and π_Q be the equilibrium distributions of the CTMCs underlying P and Q , respectively. If $S = P \otimes Q$ is ergodic on the state space given by the Cartesian product of the state spaces of P and Q , then for all $(s_p, s_q) \in \mathcal{S}_P \times \mathcal{S}_Q$,*

$$\pi_S(s_p, s_q) = \pi_P(s_p)\pi_Q(s_q). \quad (4)$$

In this case we say that the composed automaton S exhibits a product-form solution.

It is worth of stressing on the fact that the cooperating automata are *not* stochastically independent. Indeed Theorem 3 holds only for the equilibrium distribution of the joint model, i.e., when $t \rightarrow \infty$. This is coherent with the literature on product-forms of stochastic models, i.e., stochastic independence clearly implies product-form but the opposite is not true.

Example 4. We consider the model for a reversible computation shown in Fig. 4. P and Q communicate on an unreliable channel, i.e., a packet sent from P to Q is received by Q with probability p and lost with probability $1 - p$. P executes its computation in the forward ($s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5$) or backward ($s_5 \rightarrow s_4 \rightarrow s'_3 \rightarrow s'_2 \rightarrow s_1 \rightarrow s_0$) direction. It has two checkpoints modelled by states s_1 and s_4 and the synchronisations with Q occur on the transitions from s_0 to s_1 (and its dual from s_1 to s_0) and from s_4 to s_5 (and its dual from s_5 to s_4). Q moves from s_0 to s_1 or s_2 with a probabilistic choice upon the synchronisation with type a . Notice that when P is executing in the backward direction also Q is rolling back because of the synchronising type \bar{a} . Assume that the model encodes the result of the computation in state (s_5, s_4) or (s_5, s_5) (where the first component of the state is associated with P and the second with Q). We aim to compute the equilibrium probability of these two states that represents the fraction of time that the process spends in the states that encode the desired result. Notice that $a, b, \bar{a}, \bar{b} \in \mathcal{A}_P = \mathcal{P}_Q$.

Let us define the involution $\bar{\cdot}$ as: $\bar{a} = \bar{a}$, $\bar{b} = \bar{b}$. Moreover, let $\rho_P(s_i) = s_i$ for $i = 0, 1, 4, 5$ and $\rho_P(s_i) = s'_i$ and $\rho_P(s'_i) = s_i$ for $i = 2, 3$, while $\rho_Q(s_i)$ is the identity for all $i = 0, \dots, 5$. We can prove that P and Q are ρ -reversible with respect to ρ_P and ρ_Q , respectively, and $\bar{\cdot}$.

Now we use Theorem 1 to derive the equilibrium distribution of the isolated automata. Let us consider an arbitrary state in P , say s_0 . We can immediately derive $\pi_P(s_1)$ by using the detail balance equation and we obtain:

$$\pi_P(s_0)\lambda(1 - p) = \pi_P(s_1)\mu(1 - p),$$

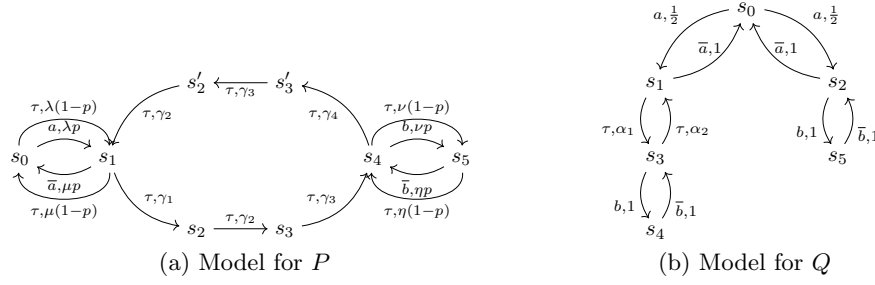


Fig. 4: A model for two communicating programs

which gives $\pi_P(s_1) = \pi_P(s_0)\lambda/\mu$. Then, we derive $\pi_P(s_2)$ using the detailed balance equation with s_1 and obtain: $\pi_P(s_2) = \pi_P(s_0)\lambda\gamma_1/(\mu\gamma_2)$. By Proposition 3 we immediately have $\pi_P(s'_2) = \pi_P(s_2)$. Then we derive $\pi_P(s'_3) = \pi_P(s_3) = \pi_P(s_0)\lambda\gamma_1/(\mu\gamma_3)$, $\pi_P(s_4) = \pi_P(s_0)\lambda\gamma_1/(\mu\gamma_4)$ and $\pi_P(s_5) = \pi_P(s_0)\lambda\gamma_1\beta/(\mu\gamma_4\eta)$. Notice that there may be more than one candidate detailed balance equation that can be applied to derive the equilibrium distribution of a state, but Theorem 1 ensures that this can be arbitrarily chosen. It remains to derive $\pi_P(s_0)$ that is computed by normalising the probabilities, i.e.:

$$\pi_P(s_0) = \frac{\gamma_2\gamma_3\gamma_4\eta\mu}{\beta\gamma_1\gamma_2\gamma_3\lambda + \gamma_1(\gamma_2\gamma_3 + 2(\gamma_2 + \gamma_3)\gamma_4)\lambda\eta + \gamma_2\gamma_3\gamma_4\eta(\lambda + \mu)}.$$

We can apply the same approach to derive the equilibrium distribution of Q , obtaining:

$$\pi_Q(s_5) = \pi_Q(s_2) = \pi_Q(s_1) = \pi_Q(s_0)\frac{1}{2}, \quad \pi_Q(s_3) = \pi_Q(s_4) = \pi_Q(s_0)\frac{\alpha_1}{2\alpha_2}.$$

and by normalising the probabilities $\pi_Q(s_0) = 2\alpha_2/(2\alpha_1 + 5\alpha_2)$. By Theorem 3 we have:

$$\pi_{P \otimes Q}(s_5, s_4) + \pi_{P \otimes Q}(s_5, s_5) = \pi_P(s_5)\pi_Q(s_4) + \pi_P(s_5)\pi_Q(s_5).$$

Notice that we have not build the joint state space and also that the automata P and Q are not independent. For example, when Q is in state s_2 and P is in checkpoint s_4 , Q moves to s_5 only if P decides not to roll back to checkpoint s_1 and the communication between P and Q is succesfull.

4.1 Synchronisation of an arbitrary number of automata

Since Definition 3 considers only cooperations of two stochastic automata, in this section we discuss how it is possible to define networks with an arbitrary number of synchronising ρ -reversible automata. The semantics we refer to when we deal with multi-way synchronisations is an instance of that presented in [29,

15]. Informally, the automata synchronise on a set of types L , i.e., the activities with type in L are carried out only jointly, while those outside are carried out independently. It is well-known that this synchronisation semantics is associative. It remains to prove that the results on ρ -reversible automata proposed here, are applicable also for this multi-way synchronisation semantics.

Let P be a ρ -reversible automaton, and $a \notin \mathcal{T}_P$, then also the automaton P^{+a} (P^{-a}) is ρ -reversible where P^{+a} (P^{-a}) is identical to P but has active (passive) type a as self-loop in each state with rate (probability) 1. Moreover, let $b \in \mathcal{A}_P$, and let P^{*b} be identical to P with the exception that label b is passive (we are assuming that for all $s_1 \in \mathcal{S}_P$ we have $\sum_{s_2 \in \mathcal{S}_P} q_P(s_1, s_2, b) \leq 1$). Then, assume we want to define a network of ρ_{P_i} -reversible automata P_1, \dots, P_M . Let τ be defined on all the types in $\cup_{i=1}^M \mathcal{T}_{P_i}$, then we can proceed as follows. Consider the automata P_1 and P_2 and define the automata $P_1^{+a_i - a_j}$ where $a_i \in \mathcal{P}_{P_2} \setminus \mathcal{A}_{P_1}$, $a_j \in \mathcal{A}_{P_2} \setminus \mathcal{P}_{P_1}$ and $P_2^{+a_k - a_h}$, where a_k and a_h are defined analogously. Then, the automaton $P_{12} = P_1^{+a_i - a_j} \otimes P_2^{+a_k - a_h}$ is well-defined according to Definition 3, is ρ -reversible and by Theorem 3 its steady-state probability is in product-form. In order to make P_{12} synchronise with P_3 we define P_{12}^{*b} , for all $b \in (\mathcal{P}_{P_1} \setminus \mathcal{A}_{P_2}) \cup (\mathcal{P}_{P_2} \setminus \mathcal{A}_{P_1})$ and repeat the procedure for the synchronisation of P_{12}^{*b} with P_3 . Notice that this procedure gives the same semantics of the master/slave synchronisation of SAN. Here, the advantage of proceeding pairwise is that we can iteratively apply Theorem 3 to derive the equilibrium distribution of the joint process very efficiently.

5 Conclusion

In this paper we studied a class of stochastic models, named ρ -reversible automata, as a novel formalism for the quantitative analysis of reversible computations. Similarly to reversible processes [19], ρ -reversible automata satisfy a system of detailed balance equations which provide an efficient technique for the computation of their equilibrium distribution. ρ -reversible automata are equipped with a synchronisation operator similar to that of [29, 15] which is associative and the class of ρ -reversible automata is closed under the synchronising operator. Moreover, ρ -reversible automata always exhibit a product-form solution (which is in general different from those known from the literature) allowing one to compute the joint equilibrium distribution as the product of the equilibrium distributions of the synchronising sub-components considered in isolation. We prove that the equilibrium distribution of any ρ -reversible stochastic automaton is insensitive to any ρ -reversible context. Therefore, our theory allows for the definition of system components whose equilibrium performance indices are independent of their context.

Acknowledgments

Work partially supported by the MIUR Project CINA: “Compositionality, Interaction, Negotiation, Autonomicity for the future ICT society”.

References

1. G. Bacci, V. Danos, and O. Kammar. On the statistical thermodynamics of reversible communicating processes. In *Proc. of Int. Conf. on Algebra and Coalgebra in Computer Science (CALCO) 2011*, pages 1–18, 2011.
2. C. Baier, E.M. Hahn, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking for performability. *Math. Structures in Comp. Sci.*, 23(S.I. 04), 2013.
3. S. Balsamo and A. Marin. Performance engineering with product-form models: efficient solutions and applications. In *Proc. of ICPE*, pages 437–448, 2011.
4. C. Bennett. Logical reversibility of computations. *IBM J. Res. Dev.*, 17(6):525–532, 1973.
5. C. Bennett. Thermodynamics of computation. *Int. J. of Physics*, 21:905–940, 1982.
6. M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202:1–54, 1998.
7. P.G. Bishop. Using reversible computing to achieve fail-safety. In *Proc. of 8th Int. Symp. on Soft. Reliability Eng.*, pages 182–191, 1997.
8. B. Boothe. Efficient algorithms for bidirectional debugging. *SIGPLAN Not.*, 35(5):299–310, 2000.
9. L. Cardelli and C. Laneve. Reversibility in massive concurrent systems. *Scientific Annals of Computer Science*, 21(2):175–198, 2011.
10. V. Danos and J. Krivine. Reversible communicating systems. In *Proc. of Int. Conf. on Concurrency Theory (CONCUR)*, pages 292–307, 2004.
11. M. Dubois, M. Annavaram, and P. Stenstrom. *Parallel Computer Organization and Design*. Cambridge Press, 2012.
12. P. G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, 2003.
13. H. Hermanns. *Interactive Markov Chains*. Springer, 2002.
14. H. Hermanns, U. Herzog, and J. P. Katoen. Process algebra for performance evaluation. *Theor. Comput. Sci.*, 274(1-2):43–87, 2002.
15. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge Press, 1996.
16. J. Hillston, A. Marin, C. Piazza, and S. Rossi. Contextual lumpability. In *Proc. of Valuetools 2013 Conf.* ACM Press, 2013.
17. D.R. Jefferson. Virtual time. *ACM Trans. on Programming Languages and Systems*, 7(3):404–425, 1985.
18. D.R. Jefferson and P. Reiher. Supercritical speedup. In *Proc. of the 24th Annual Simulation Symp.*, pages 159–168, 1991.
19. F. Kelly. *Reversibility and stochastic networks*. Wiley, New York, 1979.
20. I. Lanese, M. Lienhardt, C. A. Mezzina, A. Schmitt, and J.-B. Stefani. Concurrent flexible reversibility. In *Proc. of ESOP’13*, pages 370–390. Springer-Verlag, 2013.
21. I. Lanese, C. Antares Mezzina, and F. Tiezzi. Causal-consistent reversibility. *Bulletin of the EATCS*, 114, 2014.
22. J. Lee. Dynamic reverse code generation for backward execution. *Elect. notes in Theor. Comp. Sci.*, 174(4):37–54, 2007.
23. A. Marin and S. Rossi. Autoreversibility: exploiting symmetries in Markov chains. In *Proc. of MASCOTS 2013*, pages 151–160. IEEE Computer Society, 2013.
24. A. Marin and S. Rossi. On the relations between lumpability and reversibility. In *Proc. of MASCOTS’14*, pages 427–432, 2014.

25. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, New York, NY, USA, 2000.
26. K.S. Perumalla. *Introduction to reversible computing*. CRC Press, 2013.
27. K.S. Perumalla and A.J. Park. Reverse computation for rollback-based fault tolerance in large parallel systems. *Cluster Computing*, 16(2):303–313, 2013.
28. I. Phillips and I. Ulidowski. Reversing algebraic process calculi. *Journal of Logic and Algebraic Programming*, 73:70–96, 2007.
29. B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Perf. Eval. Rev.*, 13(2):147–154, 1985.
30. E.G. Rieffel and W.H. Polak. *Quantum Computing: a Gentle Introduction*. MIT Press, 2011.
31. W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, UK, 1994.
32. P. Whittle. *Systems in stochastic equilibrium*. John Wiley & Sons Ltd., 1986.
33. T. Yokoyama. Reversible computation and reversible programming languages. *Elect. notes in Theor. Comp. Sci.*, 253(6):71–81, 2010.
34. T. Yokoyama and R. Glück. A reversible programming language and its invertible self-interpreter. In *Proc. of the 2007 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-based Program Manipulation*, pages 144–153, New York, NY, USA, 2007. ACM.