Applying reversibility theory for the performance evaluation of reversible computations

Simonetta Balsamo[1], Filippo Cavallin[1], Andrea Marin[1], and Sabina Rossi[1]

DAIS, Università Ca' Foscari Venezia, Italy
Via Torino, 155
{balsamo,filippo.cavallin,marin,rossisab}@unive.it

Abstract. Reversible computations have been widely studied from the functional point of view and energy consumption. In the literature, several authors have proposed various formalisms (mainly based on process algebras) for assessing the correctness or the equivalence among reversible computations. In this paper we propose the adoption of Markovian stochastic models to assess the quantitative properties of reversible computations. Under some conditions, we show that the notion of time-reversibility for Markov chains can be used to efficiently derive some performance measures of reversible computations. The importance of time-reversibly relies on the fact that, in general, the process's stationary distribution can be derived efficiently by using numerically stable algorithms. This paper reviews the main results about time-reversible Markov processes and discusses how to apply them to tackle the problem of the quantitative evaluation of reversible computations.

1 Introduction

Reversible computations have two execution directions: forward, corresponding to the usual notion of computation, and backward that restores previous states of the execution. Various applications and problems related to reversible computations have been widely studied in different research areas and from different viewpoints, including functional analysis and energy consumption (see, e.g., [17, 22] and the references therein). Various formalisms and models have been proposed in the literature to represent and assess qualitative properties of reversible computations such as their correctness or if two reversible processes are equivalent in some terms. Most of the proposed approaches are based on process algebras that do not include any notion of computation time [7, 17]. We focus on the quantitative analysis and evaluation of reversible computations based on Markov stochastic processes. The dynamic behaviour of the forward and backward computation may be represented by stochastic models that include the notion of time. Hence, under certain conditions, time-reversibility of stochastic processes can be applied to assess quantitative properties of reversible computations.

Quantitative models based on Markov processes have been widely applied for the analysis and evaluation of complex systems (see e.g., [8, 5]). Markov models and formalisms have the advantage of efficient methods and algorithms for studying their behaviour. In particular, under appropriate stationary conditions one can derive the equilibrium state distribution of a continuous-time Markov chains by applying algorithms with polynomial time complexity in the process state space cardinality [25]. Several higher level formalisms that are widely applied for quantitative analysis are based on Markov processes, including Stochastic Process Algebras (SPA), Stochastic Petri Nets (SPN), Stochastic Automata Networks (SAN) and Queueing Networks (QN). Although the quantitative analysis based on these formalisms can be obtained by the direct solution of the underlying Markov chain, the state space dimension of the process in general grows exponentially with the model dimension. This is known as the state-space explosion problem and becomes intractable from the computational viewpoint as the problem size increases. In order to overcome this problem, various techniques have been proposed in the literature, including the state-space reduction by aggregating (or lumping) methods, approximation techniques, and the identification of product-form solutions for state probabilities of the Markov chain. The product-form theory provides techniques to derive the equilibrium state distribution of a complex model based on the analysis of its components in isolation. Product-form models consist of a set of interacting sub-models whose solutions are obtained by isolating them from the rest of the systems. Then, the stationary state distribution of the entire model is computed as the (normalised) product of the stationary state distributions of the sub-models. Various classes of product-form models have been defined for different formalisms and some of them can be analysed through efficient algorithms with a low polynomial complexity in the model dimension. Product-form has been widely investigated for queueing network models [14, 4]. These product-form models have simple closed-form expressions of the stationary state distributions that lead to efficient solution algorithms. For more general Markov models and by the compositionality property of Stochastic Process Algebra, the Reversed Compound Agent Theorem (RCAT) [11, 2] provides a product-form solution of a stationary CTMC defined as a cooperation between two sub-processes under certain conditions. This result gives a unified view of most of the commonly used product-forms.

The concept of time-reversibility of Markov stochastic processes has been introduced and applied to the analysis of Markov processes and stochastic networks by Kelly [16]. A reversible Markov process has the property that when the process obtained by reversing the direction of time is reversed has the same probabilistic behaviour of the original one. Early applications of these results lead to the characterisation of product-form solutions for some models with underlying time-reversible Markov process, such as closed exponential Queueing Networks [9, 4]. Also the RCAT characterisation of product-form solutions is connected to time-reversibility: the solution is based on the definition of a set of transition rates in the time-reversed process. Further notions of reversibility have been introduced in [26, 16] for dynamically reversible processes where some states of the direct and reversed processes are interchanged, and more recently the ρ-reversibility for reversible processes with arbitrary state renaming [19, 18]. Some results on properties and product-form solutions have been recently derived for this class of time-reversibility [20].

In this paper we survey the main results about time-reversible Markov processes and discuss how to apply them to address the problem of quantitative evaluation of reversible computations. We recall the definition of time reversibility for continuous time Markov processes, the main properties and its application for quantitative analysis. We present an abstract model of continuous time Markov chain for representing and performance evaluating reversible parallel computations. Taking advantage of the process reversibility, the stationary distribution of the model can be efficiently derived by using numerically stable algorithms. In particular we present some product-form results of reversible synchronising automata by applying ρ-reversibility to the underlying Markov process.

The paper is organised as follows. Section 2 introduces the notation for Markov processes and presents the time-reversibility definitions and criteria. The application of ρ-reversible Markov process to model reversible computations is presented in Section 3, where we discuss the modelling assumptions and applications of the quantitative analysis. Section 4 presents an abstract model based on continuous time Markov chains and Stochastic Automata for synchronising parallel reversible computations. We discuss the application of ρ-reversibility and the derivation of product-form solution of ρ-reversible synchronised automata that represent reversible computations, and an application example.

2 Theoretical background

Let $X(t)$ with $t \in \mathbb{R}$ be a Continuous Time Markov Chain (CTMC) with state space $\mathcal{S}$. Then, assuming that the process is irreducible, an *invariant measure* of the CTMC is a collection of positive real numbers $g(s)$ for all $s \in \mathcal{S}$ that satisfies the system of *Global Balance Equations* (GBE):

$$g(s) \sum_{s' \in \mathcal{S}} q(s, s') = \sum_{s' \in \mathcal{S}} g(s') q(s', s)\,, \tag{1}$$

or equivalently $\mathbf{gQ} = \mathbf{0}$. If the CTMC is ergodic there exists a unique invariant measure $\pi(s)$ which is also a probability distribution over $\mathcal{S}$, i.e., $\sum_{s \in \mathcal{S}} \pi(s) = 1$ and this is the steady-state distribution of the CTMC. The Markov chain $X(t)$ is *stationary* if $P\{X(0) = s\} = \pi(s)$ for all $s \in \mathcal{S}$. In the following two paragraphs we introduce the notion of time-reversibility for stationary Markov chains in the continuous time setting (for the discrete case see [16, 18]).

2.1 Time reversibility for CTMCs

Given a stationary CTMC, $X(t)$ with $t \in \mathbb{R}$, we call $X(\tau - t)$ its reversed process for all $\tau \in \mathbb{R}$. 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. Given a state renaming function

ρ (a bijection from $\mathcal{S}$ to $\mathcal{S}$), we say that $X(t)$ is *ρ-reversible* if it is stochastically identical to $X^R(t)$ modulo the state renaming ρ [19, 18]. Intuitively, an external observer is not able to distinguish $X(t)$ from $X^R(t)$ once the state renaming function ρ is applied to rename the states. Notice that if ρ is the identity then we simply say that $X(t)$ is *reversible*, whereas if ρ is an involution, then we say that $X(t)$ is *dynamically reversible* [26, 16].

We can decide if a CTMC is ρ-reversible in two ways: the first involves the steady-state distribution of the CTMC, while the latter is based on an extended formulation of Kolmogorov's criteria [16], i.e., requires the analysis of the cycles in the reachability graph.

Lemma 1. *Given a stationary CTMC $X(t)$ with state space $\mathcal{S}$, if there exists a collection of positive real numbers $\boldsymbol{\pi}$ summing to unity and a bijection ρ from $\mathcal{S}$ to $\mathcal{S}$ such that:*

$$q_s = q_{\rho(s)} \qquad \text{for all } \; s \in \mathcal{S} \tag{2}$$

$$\pi(s)q(s,s') = \pi(\rho(s'))q(\rho(s'),\rho(s)) \quad \text{for all } \; s,s' \in \mathcal{S}, s \neq s' \tag{3}$$

then $X(t)$ is ρ-reversible and $\pi(s)$ is its steady-state distribution.

Equation (2) states that the residence time of a state and its renaming must be equal. Notice that this condition is trivially satisfied if ρ is the identity, i.e., $X(t)$ is reversible. The set of equations (3) are called *detailed balance equations*. In case the renaming function ρ is known, it is possible to use the detailed balance equations to compute the chain's steady-state distribution instead of the more complex GBE.

Lemma 2. *Given a stationary CTMC $X(t)$ with state space $\mathcal{S}$ and let ρ be a renaming on $\mathcal{S}$. $X(t)$ is* ρ-reversible *with respect to ρ if and only if for every finite sequence $s_1, s_2, \ldots s_n \in S$,*

$$\begin{aligned} &q(s_1,s_2)q(s_2,s_3)\cdots q(s_{n-1},s_n)q(s_n,s_1) = \\ &\qquad q(\rho(s_1),\rho(s_n))q(\rho(s_n),\rho(s_{n-1}))\cdots q(\rho(s_3),\rho(s_2))q(\rho(s_2),\rho(s_1)) \end{aligned} \tag{4}$$

and Equation (2) holds for all $s \in \mathcal{S}$.

Analogously to Kolmogorov's criteria for reversible chains, Lemma 2 requires to check Equation (4) for all the (minimal) cycles of the CTMC and can be a useful tool for proving the ρ-reversibility of a CTMC. A consequence of ρ-reversibility is that $\pi(s) = \pi(\rho(s))$ for all $s \in \mathcal{S}$.

3 Modelling reversible computations with ρ-reversible Markov processes

Reversible computations are characterised by the fact that they have two execution directions: the forward and the backward that restores past states of the

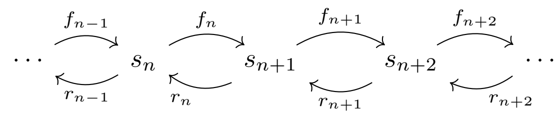

Fig. 1: Model for a reversible sequential computation.

computation. Our idea of the implementation of purely reversible computations[1] is similar to that considered in [22], i.e., the code being executed is naturally reversible. For instance, the programmer may have used Janus [27] which is a programming language for reversible computations or a subset of a standard language equipped with a reversible compiler.

3.1 Modelling reversible programming structures

In this section we describe a modelling methodology for the reversible programming structures such as sequences, branches, cycles and sequences with checkpoints.

Sequential computations. The simplest reversible computation is the reversible sequential one shown in Figure 1 where s_i are the states of the computation and the arc labels denote the transition rates, f standing for forward rates and r for the reversed ones. In this model every state can be restored in one step. For each state we define a probabilistic law that decides if the computation will proceed in the forward or backward direction. In practice these probabilities can be derived by the statistical analysis of the software execution or by the knowledge of the intrinsic law that governs the probability of proceeding in one direction or the opposite.

Assume that the residence time in state s_n is exponentially distributed with rate $f_n + r_{n-1}$, then the probability of a forward transition given that $X(t)$ is in state s_n is $f_n/(f_n + r_{n-1})$ and the probability of a backward transition is $r_{n-1}/(f_n + r_{n-1})$. This follows from the properties of the exponential random variable (see, e.g., [24]) and the so called *race policy.*

If the Markov chain depicted in Figure 1 is ergodic then it is reversible. The ergodicity is trivially satisfied if there exist lower and higher boundary states. The former is a state that does not allow a backward computation while the latter is a state that does not allow a forward computation. According to Lemma 2 we have that the forward cycle $s_n \xrightarrow{f_n} s_{n+1} \xrightarrow{r_n} s_n$ has itself as inverse cycle and therefore the conditions of Lemma 2 are satisfied.

[1] By purely reversible computations we mean those computations in which each step can be undone and there are no segments in which the execution direction is forward only.

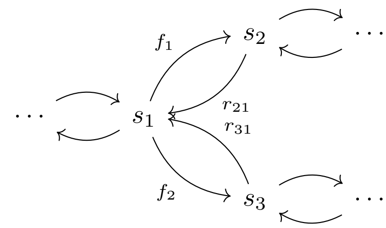

Fig. 2: Model for a reversible branch.

Branches. Branches can be modelled in a similar way to the one used for the sequential computations. Notice that, as commonly done in stochastic modelling, we model the branch by means of the probabilistic behaviour of the executed process. Although a modelling approach taking into account the detailed description of the system state is theoretically possible, in many cases this is not practically feasible due to the high cardinality that would be reached by the state space. Suppose that state s_1 is associated with a branch that proceeds to state s_2 with probability p and to s_3 with probability $1-p$ (see Figure 2). In this case, let $1/f$ be the expected residence time in state s_1, then the transition rates are $f_1 = fp$ and $f_2 = f(1-p)$. Following the reasoning proposed in the previous paragraph on sequential computations, it is easy to see that the conditions of Lemma 2 are satisfied by choosing ρ as the identity.

Cycles. Cycles can be modelled as long as each transition they consist of can be undone. Let us consider the model of Figure 3 where the computation at state s_1 can proceed by entering the cycle s_1, s_2, s_3, s_4 or by moving to state s_5. The probability of entering the cycle given that the computation will proceed in the forward direction is $f_1/(f_1 + f_1')$ and the number of (forward) iterations are geometrically distributed. Modelling the exact number of iterations of the cycles is possible but, in general, will drastically increase the number of model states. Let us focus on the cycle s_1, s_2, s_3, s_4 and its inverse s_1, s_4, s_3, s_2. If we apply Lemma 2 with ρ being the identity function, we notice that the conditions are satisfied for the cycles consisting of two states (e.g., s_3, s_4, s_3) but we need also to consider the cycle s_1, s_2, s_3, s_4 whose inverse is s_1, s_4, s_3, s_2 that originates a rate-condition for the ρ-reversibility: $f_1 f_2 f_3 f_4 = r_1 r_2 r_3 r_4$. *In general, in cycles, the product of the forward rates must be equal to the product of the corresponding backward rates.* This is trivially satisfied if the time required to perform a forward computation follows the same distribution of that required to undo it.

Sequences with checkpoints. In the previous paragraphs we have shown how it is possible to model reversible sequential computations, branches and cycles by using a reversible CTMC, i.e., by taking the identify as ρ function. In the context of modelling reversible computations, the notion of ρ-reversibility is important because it allows the specification of atomic sequences that can be only fully

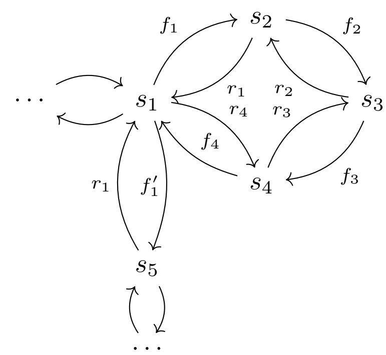

Fig. 3: Model for a reversible cycle.

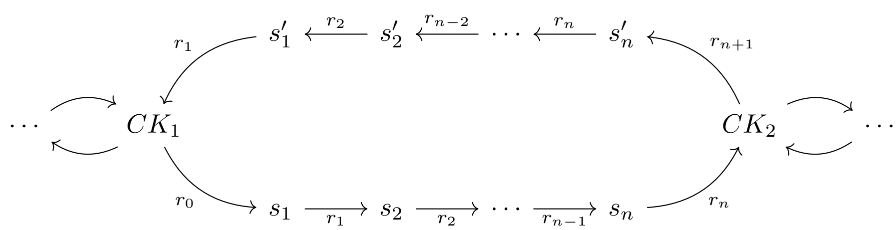

Fig. 4: Model for a reversible computation with checkpoints.

reversed. For instance, consider a system atomic transaction whose correctness is tested at a certain checkpoint. If an invalid state is detected, then all the operations performed by the transaction must be undone. An example of such a computation is shown in Figure 4. In order to prove the ρ-reversibility of the model, we define function ρ as:

$$\rho(s) = \begin{cases} CK_1 & \text{if } s = CK_1 \\ CK_2 & \text{if } s = CK_2 \\ s'_i & \text{if } s = s_i \quad 1 \le i \le n \\ s_i & \text{if } s = s'_i \quad 1 \le i \le n \end{cases} .$$

By Lemma 2 we observe that the residence time of s_i must have the same expectation of that of its ρ-renaming, s'_i (and vice versa). Therefore, we have the rate condition for the ρ-reversibility whose interpretation is that the time required to perform an operation in the transaction must follow the same probabilistic law of that required to undo it. For what concerns the cycle analysis, observe that $CK_1, s_1, \dots, s_n, CK_2, s'_n, \dots, s'_1, CK_1$ has itself as inverse and hence the condition (4) is satisfied.

3.2 Modelling assumptions and steady-state

In this section we discuss two crucial points of the modelling technique that we propose: How does the exponential assumption of the distribution of the state residence time impact on the expressiveness of this modelling framework? How do we interpret the steady-state distribution of Markov chains in terms of quantitative properties of the reversible computations?

The exponential assumption can be relaxed by using distributions whose coefficient of variation may be higher or lower than that of the exponential. This is achieved by splitting a state whose residence time is not exponential into a set of micro-states each of which has an exponential residence time. Coxian random variables are formed by exponential stages and can approximate any distribution with rational Laplace transform with arbitrary accuracy (see, e.g., [15]). The literature proposing algorithms to fit data statistics to a distribution by means of a combination of the exponential stages is very rich (e.g., [6, 21]).

Informally, the steady-state distribution of a CTMC is the probability of observing a given state when the time elapsed since the first observation is very large (the time required to reach the stationary behaviour depends on the magnitude of the second eigenvalue of the infinitesimal generator). For instance, in stationary reversible simulations [22] the state of the process after a period of warm up, is independent of its initial conditions and hence our framework can be applied easily. The assumption that each state transition can be undone includes the transitions that take the model to the state encoding the result of the computation. As a consequence, it is not obvious how the steady-state distribution can give an idea about the time required to obtain the result in a reversible computation. In stochastic analysis this problem is connected to the computation of the (moments of the) distribution of the time to absorption. Basically, we assume that once the chain enters in one of the states encoding the result, then they cannot leave them. Unfortunately, to the best of our knowledge, time-reversibility does not help in the exact computation of the distribution of the time to absorption. Nevertheless, approximating methods which may take advantage from the process' reversibility are available and are quite accurate when the expected computation time is much higher than the expected transition delays of the model (see, e.g., [3, 1]). The steady-state distribution may also be interpreted as the fraction of a large number of processes which are in a given state (in the long-run) once they are run in parallel and they restart their computation after terminating it.

4 Cooperation of reversible parallel computations

In this section we present an abstract model based on continuous time Markov chains for the performance evaluation of reversible parallel computations. Differently from those functional models that represent explicitly the parallel composition of reversible computations, we do not consider here any notion of causality. Instead we present a stochastic model for analysing the dynamic behaviour of those computations that can be realized in a reversible fashion, where the underlying conditional probabilities play the role of causality.

4.1 Labelled stochastic automata and synchronisation

We introduce the notion of labelled stochastic automaton as a model for synchronizing computations. In the definition of stochastic automata we distinguish between *active* and *passive* action types, and only active/passive synchronisations are allowed when forming the composition of automata.

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, named* 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 $\mathcal{T}_P = (A_P \cup \mathcal{P}_P \cup \{\tau\})$ and for all $s \in \mathcal{S}_P$, $(s, s, \tau) \notin \rightsquigarrow_P$*[2]
- q_P *is a function from $\rightsquigarrow_P$ to $\mathbb{R}^+$ such that $\forall s_1 \in \mathcal{S}_P$ and $\forall a \in \mathcal{P}_P$, $\sum_{s_2:(s_1,s_2,a)\in\rightsquigarrow_P} q_P(s_1, s_2, a) \le 1$.*

Hereafter, we denote by $\to_P$ the relation defined as

$$\to_P = \{(s_1, s_2, a, q) \mid (s_1, s_2, a) \in\rightsquigarrow_P \text{ and } q = q_P(s_1, s_2, a)\}.$$

We will use the notation $s_1 \overset{a}{\rightsquigarrow}_P s_2$ to denote the tuple $(s_1, s_2, a) \in\rightsquigarrow_P$; moreover we denote by $s_1 \xrightarrow{(a,r)}_P s_2$ (resp., $s_1 \xrightarrow{(a,p)}_P s_2$) the tuple $(s_1, s_2, a, r) \in\to_P$ (resp., $(s_1, s_2, a, p) \in\to_P$).

For $s, s' \in \mathcal{S}_P$ and for $a \in \mathcal{A}_P \cup \{\tau\}$, $q_P(s, s', a) \in \mathbb{R}^+$ denotes the *rate* of the transition from s to s' with type a. For $s, s' \in \mathcal{S}_P$ and for $a \in \mathcal{P}_P$, $q_P(s, s', a) \in (0, 1]$ denotes the *probability* that the automaton synchronises on type a with a transition from s to s'. In the following, we adopt the convention that $q_P(s, s', a) = 0$ whenever there are no transitions with type a from s to s'. For $s \in \mathcal{S}_P$ and for $a \in \mathcal{T}_P$ we write $q_P(s, a) = \sum_{s'\in\mathcal{S}_P} q_P(s, s', a)$. We say that the automaton P is *closed* if $\mathcal{P}_P = \emptyset$.

Every closed automaton has an underlying continuous time Markov chain as defined below.

Definition 2. (CTMC underlying a closed automaton) *Given a closed automaton P, we denote by $X_P(t)$ the CTMC underlying P, whose state space is $\mathcal{S}_P$ and whose infinitesimal generator matrix $\mathbf{Q}$ is as follows: for all $s_1 \neq s_2 \in \mathcal{S}_P$,*

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

[2] We exclude the τ self-loops from the definition of stochastic automaton in order to simplify the semantics of synchronisation. Indeed, the τ self-loops are irrelevant for the equilibrium distribution of the CTMC underlying the automaton.

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

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

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

Table 1: Operational rules for SA synchronisation

A closed automaton P is said to be ergodic *(*irreducible*) if its underlying CTMC is ergodic (irreducible). The equilibrium distribution of the CTMC underlying the automaton P is denoted by $\boldsymbol{\pi}_P$.*

Stochastic automata can be composed throughout a synchronisation operator which is defined in the style of the master/slave synchronisation of SANs [23] based on the Kronecker's algebra and the active/passive cooperation operation used in Markovian process algebra such as PEPA [12, 13].

Definition 3. (SA synchronisation) *Let P and Q be two stochastic automata and assume that $\mathcal{A}_P = \mathcal{P}_Q$ and $\mathcal{A}_Q = \mathcal{P}_P$. The parallel composition of P and Q is the automaton $P \otimes Q$ defined 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*
– $\rightsquigarrow_{P\otimes Q}$ *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* $((s_{p_1}, s_{q_1}),(s_{p_1}, s_{q_2}), a) \in \rightsquigarrow_{P\otimes Q}$ *and* $q = q_{P\otimes Q}((s_{p_1}, s_{q_1}), (s_{p_1}, s_{q_2}), a)$.

Notice that, according to the above definition, an automaton obtained by a composition does not have passive types. This is reasonable if we consider the fact that in this case the resulting automaton has an underlying CTMC and then we can study its equilibrium distribution. In [20] we show that this semantics for pairwise SA synchronisations can be easily extended in order to include an arbitrary finite number of pairwise cooperating automata.

4.2 Reversible Stochastic Automata

We now introduce the notion of *ρ-reversibility* for stochastic automata. We present a definition in the style of the Kolmogorov's criteria stated in [16].

We assume the existence of a bijection (renaming) $\overleftarrow{\cdot}$ from $\mathcal{T}_P$ to $\mathcal{T}_P$ such that for each forward action type a there is a corresponding backward action type $\overleftarrow{a}$ with $\overleftarrow{\tau} = \tau$. In most of practical cases, $\overleftarrow{\cdot}$ is an involution, i.e., $\overleftarrow{\overleftarrow{a}} = a$ for all $a \in \mathcal{T}_P$, and hence the semantics becomes similar to the one proposed in [7]. We say that $\overleftarrow{\cdot}$ respects the active/passive types of an automaton P if $\overleftarrow{\tau} = \tau$ and for all $a \in \mathcal{T}_P \setminus \{\tau\}$ we have that $a \in \mathcal{A}_P \Leftrightarrow \overleftarrow{a} \in \mathcal{A}_P$ (or equivalently $a \in \mathcal{P}_P \Leftrightarrow \overleftarrow{a} \in \mathcal{P}_P$).

The notion of ρ-reversible automaton is defined as follows.

Definition 4. (ρ-reversible automaton) *Let P be an irreducible stochastic automaton. Assume that*

- *$\rho : \mathcal{S}_P \to \mathcal{S}_P$ is a renaming (permutation) of the states, and*
- *$\overleftarrow{\cdot}$ is a bijection from $\mathcal{T}_P$ to $\mathcal{T}_P$ that respects the active/passive typing.*

We say that P is ρ-reversible if

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

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

We say that $\overleftarrow{\Phi}$ is the inverse *of cycle Φ. If ρ is the identity function we simply say that P is* reversible*.*

Notice that the inverse cycle $\overleftarrow{\Phi}$ of a cycle Φ is unique. This can be easly derived 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 $a \in \mathcal{T}_P$.

The following theorem states that any ρ-reversible automaton satisfies a set of detailed balance equations similar to those presented in Lemma 1.

Theorem 1. (Detailed balance equations) *If P is ergodic and ρ-reversible then for each pair of states $s, s' \in \mathcal{S}_P$, and for each type $a \in \mathcal{T}_P$, we have*

$$\pi_P(s) q(s, s', a) = \pi_P(s') q(\rho(s'), \rho(s), \overleftarrow{a}) .$$

The next proposition says that the states of an ergodic ρ-reversible automaton have the same equilibrium probability of the corresponding image under ρ.

Proposition 1. (Equilibrium probability of the renaming of a state) *If P is an ergodic and ρ-reversible automaton then for all $s \in \mathcal{S}_P$,*

$$\pi_P(s) = \pi_P(\rho(s)) .$$

4.3 Product-form result

It is well-known that the cardinality of the state space of complex systems can grow exponentially with the structure of the model. Even worse, the numerical algorithms for deriving the equilibrium distribution become numerically unstable and prohibitive in terms of computation time. In this section we present the product-form result for networks of ρ-reversible synchronising automata. First we prove that the parallel composition of ρ-reversible automata is still ρ-reversible. Then, based on this result, we prove that the equilibrium distribution of the composition of two ρ-reversible automata can be derived from the equilibrium distribution of the cooperating automata considered in isolation (i.e., without generating the joint state space and solving the system of global balance equations). The analysis in isolation requires to set a rate for the passive transitions. To this aim, in [20] we prove that, thanks to the *rescaling property* of ρ-reversible automata, we can choose an arbitrary positive constant.

Theorem 2. (Closure under ρ-reversibility) *Let P_1 and P_2 be two ρ_1- and ρ_2-reversible automata with respect to the same function $\overleftharpoon{\cdot}$ on the action types. Then, the composition $P_1 \otimes P_2$ is ρ-reversible with respect to the same $\overleftharpoon{\cdot}$, where, for all $(s_1, s_2) \in \mathcal{S}_{P_1} \times \mathcal{S}_{P_2}$,*

$$\rho(s_1, s_2) = (\rho_1(s_1), \rho_2(s_2)) \,. \tag{5}$$

The next theorem provides the product-form result for networks of ρ-reversible stochastic automata. In order to understand the relevance of this result, consider a set of M cooperating automata and assume that each automaton has a finite state space of cardinality N. The state space of the network may have the size of the Cartesian product of the state space of each single automaton, i.e., in the worst case, its cardinality is 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 $\mathcal{O}(N^{3M})$. For ρ-reversible automata, by applying Theorem 1, we can efficiently compute the equilibrium distribution of each automaton in linear time on the cardinality of the state space, and by Theorem 3 the complexity of the computation of the joint equilibrium distribution is $\mathcal{O}(NM)$.

Theorem 3. (Product-form solution) *Let P_1 and P_2 be two ergodic ρ_1- and ρ_2-reversible automata with respect to the same function $\overleftharpoon{\cdot}$ on the action types, and let $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ be the equilibrium distributions of the CTMCs underlying P_1 and P_2, respectively. If $P_1 \otimes P_2$ is ergodic on the state space given by the Cartesian product of the state spaces of P_1 and P_2, then for all $(s_1, s_2) \in \mathcal{S}_{P_1} \times \mathcal{S}_{P_2}$,*

$$\pi(s_1, s_2) = \pi_1(s_1)\pi_2(s_2) \tag{6}$$

where $\boldsymbol{\pi}$ is the equilibrium distribution of the CTMC underlying $P_1 \otimes P_2$. In this case we say that the composed automaton exhibits a product-form solution*.*

Notice that this analysis, differently from those based on the concepts of quasi-reversibility [16, 10] 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.

4.4 Example

In this section we describe a model for the parallel composition of two reversible computations. Consider the stochastic automata P_1 and P_2 depicted in Fig. 5. P_1 and P_2 communicate on the reversible channels a, b and c. Channel a is unreliable, i.e., a packet sent from P_1 to P_2 is recevied by P_2 with probability p and lost with probability $1-p$. P executes its computations in the forward ($s_0 \to s_1 \to s_2 \to s_3 \to s_4 \to s_5$ or $s_0 \to s_1 \to s_2 \to s_3 \to s_4 \to s_6$) or backward ($s_5 \to s_4 \to s'_3 \to s'_2 \to s_1 \to s_0$ or $s_6 \to s_4 \to s'_3 \to s'_2 \to s_1 \to s_0$) direction. It has two checkpoints modelled by states s_1 and s_4. P_2 moves from t_0 to t_1 or t_2 with a probabilistic choice upon the synchronisation with type a. P_1 is ρ_1-reversible with $\rho_1(s_i) = s_i$ for $i = 0, 1, 4, 5, 6$ and $\rho_1(s_i) = s'_i$ and $\rho_1(s'_i) = s_i$ for $i = 2, 3$, while P_2 is ρ_2-reversible where ρ_2 is the identity function. Notice that $a, \overleftarrow{a}, b, c \in \mathcal{A}_{P_1} = \mathcal{P}_{P_2}$ and $\overleftarrow{b}, \overleftarrow{c} \in \mathcal{A}_{P_2} = \mathcal{P}_{P_1}$.

We assume that the model encodes the result of the computation in the states (s_5, t_2), (s_5, t_4), (s_6, t_2), (s_6, t_4). We aim to compute the equilibrium probability of these four states that represents the fraction of time that the process spends in the states that encode the desired result.

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 $\pi_1(s_1)$ by using the detail balance equation and we obtain:

$$\pi_1(s_0)\lambda(1-p) = \pi_1(s_1)\mu(1-p)\,,$$

which gives $\pi_1(s_1) = \pi_1(s_0)\lambda/\mu$. Then, we derive $\pi_1(s_2)$ using the detailed balance equation with s_1 and obtain: $\pi_1(s_2) = \pi_1(s_0)\lambda\gamma_1/(\mu\gamma_2)$. By Proposition 1 we immediately have $\pi_1(s'_2) = \pi_1(s_2)$. Then we derive $\pi_1(s'_3) = \pi_1(s_3) = \pi_1(s_0)\lambda\gamma_1/(\mu\gamma_3)$, $\pi_1(s_4) = \pi_1(s_0)\lambda\gamma_1/(\mu\gamma_4)$, $\pi_1(s_5) = \pi_1(s_0)\lambda\gamma_1\nu q/(\mu\gamma_4)$ and $\pi_1(s_6) = \pi_1(s_0)\lambda\gamma_1\nu(1-q)/(\mu\gamma_4)$. It remains to derive $\pi_1(s_0)$ that is computed by normalising the probabilities. We can apply the same approach to derive the equilibrium distribution of P_2, obtaining:

$$\pi_2(t_1) = \pi_2(t_3) = \pi_2(t_0)\frac{1}{2}\,, \quad \pi_2(t_2) = \pi_2(t_0)\frac{1}{2\alpha}\,, \quad \pi_2(t_4) = \pi_2(t_0)\frac{1}{2\beta}\,.$$

Again, by normalising the probabilities, we obtain $\pi_2(t_0)$. By applying Theorem 3 we can now easily derive the desired result:

$$\pi(s_5,t_2) + \pi(s_5,t_4) + \pi(s_6,t_2) + \pi(s_6,t_4) = \\ \pi_1(s_5)\pi_2(t_2) + \pi_1(s_5)\pi_2(t_4) + \pi_1(s_6)\pi_2(t_2) + \pi_1(s_6)\pi_2(t_4)\,.$$

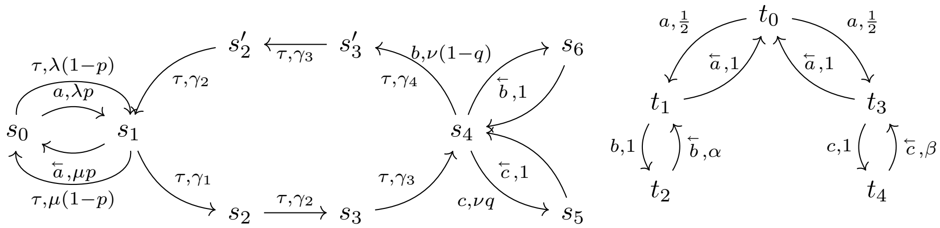

Fig. 5: Models for P_1 and P_2

Notice that we have not build the joint state space and also that the automata P_1 and P_2 are not independent. For example, when P_2 is in state t_1 and P_1 is in checkpoint s_4, P_2 moves to t_2 only if P_1 decides neither to roll back to checkpoint s_1 nor to move to s_5.

5 Conclusion

In this paper we have proposed an abstract modelling framework for the quantitative analysis of reversible computations. The main idea is to exploit the time-reversibility property of Markov processes in order to provide a computationally efficient way of deriving the desired performance indices. We have shown that, under some conditions, the proposed approach is suitable to be applied for a compositional formalism based on labelled stochastic automata. As a consequence the advantages (reduction of time-complexity and improvement of algorithms' numerical stability) of time-reversibility are applicable also for the analysis of the cooperation of automata that are proved to have product-form steady-state distributions.

References

1. J. R. Artalejo, A. Economou, and M. J. Lopez-Herrero. The maximum number of infected individuals in SIS epidemic models: computational techniques and quasi-stationary distributions. *J. of Comput. Appl. Math.*, 233:2563–2574, 2010.
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. A. D. Barbour. Quasi-stationary distributions in markov population processes. *Adv. Appl. Prob.*, 8:296–314, 1976.
4. F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM*, 22(2):248–260, 1975.
5. M. Bugliesi, L. Gallina, S. Hamadou, A. Marin, and S. Rossi. Behavioural equivalences and interference metrics for mobile ad-hoc networks. *Perform. Eval.*, 73:41–72, 2014.

6. G. Casale and E. Smirni. KPC-toolbox: fitting markovian arrival processes and phase-type distributions with MATLAB. *SIGMETRICS Performance Evaluation Review*, 39(4):47, 2012.
7. V. Danos and J. Krivine. Reversible communicating systems. In *Proc. of Int. Conf. on Concurrency Theory (CONCUR)*, pages 292–307, 2004.
8. L. Gallina, S. Hamadou, A. Marin, and S. Rossi. A probabilistic energy-aware model for mobile ad-hoc networks. In *Analytical and Stochastic Modeling Techniques and Applications - 18th International Conference, ASMTA 2011, Venice, Italy, June 20-22, 2011. Proceedings*, pages 316–330, 2011.
9. W. J. Gordon and G. F. Newell. Cyclic queueing networks with exponential servers. *Oper. Res.*, 15(2):254–265, 1967.
10. P. G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, 2003.
11. P. G. Harrison. Reversed processes, product forms and a non-product form. *Elsevier Linear Algebra and Its Applications*, 386:359–381, 2004.
12. J. Hillston. *A Compositional Approach to Performance Modelling.* Cambridge Press, 1996.
13. J. Hillston, A. Marin, C. Piazza, and S. Rossi. Contextual lumpability. In *Proc. of Valuetools 2013 Conf.* ACM Press, 2013.
14. J. R. Jackson. Jobshop-like queueing systems. *Management Science*, 10:131–142, 1963.
15. K. Kant. *Introduction to Computer System Performance Evaluation.* McGraw-Hill, 1992.
16. F. Kelly. *Reversibility and stochastic networks.* Wiley, New York, 1979.
17. I. Lanese, C. Antares Mezzina, and F. Tiezzi. Causal-consistent reversibility. *Bulletin of the EATCS*, 114, 2014.
18. A. Marin and S. Rossi. On discrete time reversibility modulo state renaming and its applications. In *Proc. of Valuetools 2014 Conf.*, 2014.
19. A. Marin and S. Rossi. On the relations between lumpability and reversibility. In *Proc. of MASCOTS'14*, pages 427–432, 2014.
20. A. Marin and Sabina Rossi. *Quantitative Analysis of Concurrent Reversible Computations*, volume 9268 of *LNCS*, pages 206–221. Springer-Verlag, 2015.
21. A. Mészáros, J. Papp, and M. Telek. Fitting traffic traces with discrete canonical phase type distributions and markov arrival processes. *Applied Mathematics and Computer Science*, 24(3):453–470, 2014.
22. K.S. Perumalla. *Introduction to reversible computing.* CRC Press, 2013.
23. B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. *SIGMETRICS Perf. Eval. Rev.*, 13(2):147–154, 1985.
24. S. M. Ross. *Stochastic Processes.* John Wiley & Sons, 2nd edition, 1996.
25. W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, UK, 1994.
26. P. Whittle. *Systems in stochastic equilibrium.* John Wiley & Sons Ltd., 1986.
27. 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.