Compositional Model Checking of product-form CTMCs

Paolo Ballarini[1] and András Horváth[2]

Abstract

Product form Markov chains are a class of compositional Markovian models that can be proved to benefit from a decomposed solution of the steady-state distribution (i.e. the steady-state distribution is given by the product of the components' steady-state distributions). In this paper we focus on the *Boucherie* product processes, a specific class of product form Continuous Time Markov Chains. We show that the compositional constraints that lead to the product form result in that class, can be exploited in the model checking problem as well, leading to a decomposed semantics for a fragment of the Continuous Stochastic Logic.

Keywords: Compositional Stochastic Model Checking, Product-form Markov Chains

1 Introduction

The Continuous Stochastic Logic (CSL) [1,2] has proved to be a valuable language for expressing performance and performability requirements over systems modelled as Continuous Time Markov Chains (CTMCs). CSL properties are verified against a CTMC model by means of appropriate model checking algorithms. Model checking procedures for finite state models are sensitive to the size of the model. A considerable amount of works aiming to increase the applicability of model checking with respect to the model's dimension can be found in literature. There are at least three different types of approach for tackling the so-called *state-space explosion problem*: compacting the state-space representation (i.e. *symbolic model checking*); reducing the state-space dimension through *abstraction*; reducing the state-space through decomposition of the original model (i.e. *compositional model checking*).

For example, *symbolic model checking*, (for both non-probabilistic [10] and probabilistic [9,6] systems), employs specific data structures (BDDs, MTBDDs) to get a compact representation of the state-space. On the other hand, *abstraction* in model checking aims to look for an *abstracted*, hence reduced, version of the model which

[1] Computing Science Department, University of Glasgow, UK, paolo@dcs.gla.ac.uk

[2] Dipartimento Informatica, Università di Torino, Italy, horvath@di.unito.it, partially supported by MIUR through PRIN project Famous and EEC project Crutial.

turns out to be equivalent to the original one from some point of view. *Compositional verification* of properties in a given temporal logic, instead, concerns the analysis of the truth of a formula when the given model is obtained by composition of a number of submodels. The goal, in that respect, is to investigate the possibility of inferring the truth of a formula ϕ by the verification of ϕ itself, or some other formulae, on the component models. Compositional approaches to model checking of non-probabilistic system have been widely studied (see, for example, CTL compositional model checking [7] and *module checking* [11]).

If lot has been done with respect to the compositional verification of non-probabilistic systems, to the best of our knowledge, the compositional verification of properties referred to probabilistic systems still remains a mainly unexplored area of research. In this paper we consider the CTMC domain of probabilistic systems and we tackle the compositional verification issue with respect to it. Specifically, we take into consideration a compositional framework for CTMCs, namely the Boucherie product process [3], in which a K-dimensional CTMC is obtained as the product of K components CTMC. The Boucherie framework is suitable to model systems in which a number of parallel processes compete over a number of mutually exclusive, shared resources. No explicit synchronisation between processes is considered apart from the implicit one due to the mutually exclusive access to shared resources. By the imposition of two constraints on the compositional rule, respectively *mutual-exclusion* and *strong blocking*, the relevant results proved by Boucherie is that the *steady-state* distribution of the product-process M is given by the product of the steady-state distribution of M's components.

In this paper we consider the CSL model checking problem for the family of Boucherie processes. We show that compositional verification of a subset of the CSL language can be performed on a Boucherie product process. Hence given a CSL formula ϕ referred to a K-dimensional process M we show how an equivalent (set of) formula(e) ϕ', that refer to some of M's components, can be derived.

The remainder of the paper is organised as follows. In Section 2 the Boucherie compositional framework is formally described and an example, which will be referred to throughout the paper, is presented. In Section 3 the CSL logic is briefly introduced and the compositional verification is proved for a subset of it. Section 4 discusses the gains we have by applying the proposed approach. Finally Section 5 summarises the work presented in this paper and illustrates guidelines for future work.

CTMC basics. We introduce the basic notions/notations that concern CTMCs. They will be used in the remainder of the paper. Given a set of atomic propositions $AP = \{a, b, c \ldots\}$ a labelled CTMC is denoted $M = (S, Q, L)$ where S is a finite set of states, $Q : S \times S \to \mathbb{R}_{\geq 0}$ is the rate matrix, with $Q(s, s) = 0$, and $L : S \to 2^{AP}$ is the labelling function. The *transition rate* $Q(s, s') > 0$ if and only if there is a transition from s to s' and the delay of a transition $s \to s'$ is governed by an exponential distribution whose parameter is the *transition rate* $Q(s, s')$. Any state s such that $Q(s, s') = 0$ for all $s' \in S$ is called *absorbing*. The sum of the outgoing *transition rates* from a state s is called the *exit rate* of s and it is denoted by $E(s) = \sum_{s' \in S} Q(s, s')$. Whenever $Q(s, s') > 0$ for more than one state s', then there is a race between different transitions from s. In such a case the probability that a transition from s to s' ($s \neq s'$) occurs within t time units is given by $P(s,s',t) = \frac{Q(s,s')}{E(s)} \cdot \left(1 - e^{-E(s)\cdot t}\right)$. $P(s,s') = \frac{Q(s,s')}{E(s)}$ represents the *embedded transition probability matrix* of M. The probability of leaving a state $s \in S$ within a time interval $I = [a,b] \subseteq \mathbb{R}_{\geq 0}$ is denoted $e_I(s) = (e^{-a\cdot E(s)} - e^{-b\cdot E(s)})$. The *steady-state* distribution for a CTMC M (i.e. the distribution that indicates the probability of being in a certain state in the long-run) is denoted π^M (or simply π whenever indicating M is not relevant in the context of the discourse). For a collection of $K > 1$ CTMCs, the subscript $_k$ will be used for referring to the k^{th} ($1 \leq k \leq K$) CTMC in the collection (e.g. $M_k = (S_k, Q_k, L_k)$, AP_k, P_k, π^{M_k}).

2 Boucherie product form

In [3], Boucherie introduced a compositional CTMC, M, suitable to model competition between concurrent processes over a number of shared resources. Such a model is described as a collection of K ergodic CTMCs, $M_k, 1 \leq k \leq K$, each of them with finite state-space, S_k, transition matrix, Q_k and unique steady-state distribution, π_k.

An index set, $I \subset \mathbb{N}_{>0}$, represents the shared resources. The set of components competing for resource i is denoted by U_i. We assume that U_i contains at least two components for every resource $i \in I$ (otherwise the resource would not be shared). The set $C_{ki} \subset \{1,\ldots,K\}$ represents the components competing with component k over resource $i \in I$ and $R_k \subset I$ indicates the resources component k is competing over. For given k and $i \in I$, A_{ki} denotes the set of states of M_k in which resource i is in use by component k. It is assumed for any component k and any two resources i and j that $A_{ki} \cap A_{kj} = \emptyset$. A_{k0} will denote the set of states where no shared resources are used by k[3]. The product process M is characterised by two conditions. **Condition 1:** each transition can change the state of one component only (i.e., there is *no synchronisation*). **Condition 2 :** resources are *mutually exclusive* and *strong blocking*, that is, if component k holds resource i then all its competitors (i.e., $k' \in C_{ki}$) are blocked.

Under these assumptions Boucherie proved that the steady state probability of a state $s = (s_1,\ldots,s_k,\ldots,s_K)$ of the composed process, M, is of product form, i.e., it can be computed as $\pi(s_1,\ldots,s_k,\ldots,s_K) = G\prod_{i=1}^{K} \pi_i(s_i)$ where $\pi_i(s_i)$ denotes the steady state probability of state s_i in the CTMC describing component i. For the efficient calculation of the normalisation constant, G, see [12]. The state-space, S, is obtained by subtracting from the product $\Pi_{k\in K} S_k$ the set of states that represent a breach of the mutually exclusive condition (the set of these states will be denoted by ME) and the set of states that correspond to circular blocking (the set of these states will be denoted by CB). As a result S is formally defined as:

$$S = \prod_{k=1}^{K} S_k \setminus (ME \cup CB) \tag{2.1}$$

The set ME can be computed as $ME = \bigcup_{i:i\in I} \bigcup_{J:J\subset U_i,|J|\geq 2} \prod_{k=1}^{K} D_{ki}^{J}$ where $D_{ki}^{J} =$

[3] to make it more intuitive we denote non-shared resources with $i = 0$, whereas in [3] the resource index $i = 1$ is used to represent non-shared ones.

A_{ki} if $k \in J$ and $D_{ki}^{J} = S_k$ otherwise, i.e., every state in which a given resource is used by at least two components must be excluded because it violates the condition of mutual exclusion.

In order to identify the states in CB, we introduce the following notation. A given resource allocation situation will be represented by the vector $|r_1, \ldots, r_K|$ where $r_i \in I$ (i.e., component i uses resource r_i) or $r_i = 0$ (i.e., component i does not use any shared resource). Then the system is in circular blocking in $|r_1, \ldots, r_K|$ if there exists a sequence of component indices $i_1, \ldots, i_C, 2 \leq C \leq K$ such that $r_{i_1} \in U_{i_2}$, $r_{i_2} \in U_{i_3}, \ldots,$ $r_{i_{C-1}} \in U_{i_C}$, and $r_{i_C} \in U_{i_1}$. The set of vectors representing circular blocking will be denoted by CB_r. Then $CB = \bigcup_{r:r\in CB_r} \prod_{k=1}^{K} A_{kr_k}$.

The *no-synchronisation* and *strong-blocking* constraints Boucherie framework are reflected on the rate-matrix Q of M. For two states, $s = (s_1, \ldots, s_k, \ldots, s_K)$ and $s' = (s'_1, \ldots, s'_k, \ldots, s'_K)$, $Q(s, s') = 0$ if s and s' differ in more than one component; $Q(s, s') = Q_k(s_k, s'_k)$ if s and s' differ only in component k and component k is not blocked in s; $Q(s, s') = 0$ if s and s' differ only in component k and component k is blocked in s.

In this work we extend the Boucherie framework by assuming that states of component k are labelled with atomic propositions from set AP_k. For convenience we further assume that sets AP_k $(1 \leq k \leq K)$ are pair-wise disjoint. The set of atomic propositions of M is $AP = \cup_{k=1}^{K} AP_k$.

As an example Boucherie process we consider the stochastic version of the popular Dining Philosophers problem. In such a model K philosophers are sitting around a table on which K chopsticks (or forks) are disposed so that each philosopher share the chopstick on his right with his right neighbour and the chopstick on his left with his left neighbour. A philosopher behaviour is described as an infinite loop consisting of two activities: *thinking* and *eating*. In order to eat a philosopher must get both chopsticks, which he will release as soon as he starts thinking again.

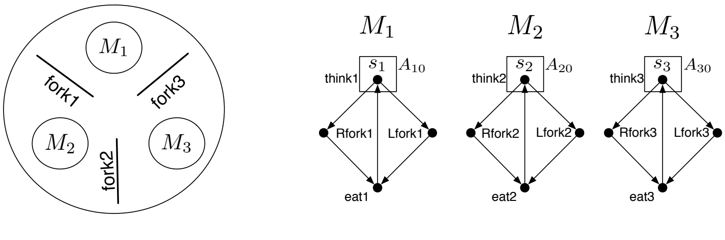

Fig. 1. Three dining philosophers and corresponding CTMCs

In this paper we refer to the case with $K = 3$ philosophers where $fork_k$, $k \in \{1, 2, 3\}$, denotes the resource shared by M_k and $M_{(k+1)mod(K)}$. Furthermore we use the resource eat_k which is shared between M_k and its two neighbours. Since we are considering 3 philosophers only then we have a single eat resource which is shared among the three of them.

Process/resource competition and corresponding CTMCs are shown in Figure 1. States of component M_k are labelled with labels $think_k$, $Rfork_k$, $Lfork_k$ and eat_k (we use t_k, Rf_k, Lf_k and e_k as abbreviations for them) according to resources occupation. Transition rates are all assumed to be 1. Philosophers' competition and the resulting state space partition is summarised as follows:

forks busy	states
all free	$(t_1t_2t_3)$
1 used	$(Lf_1t_2t_3), (Rf_1t_2t_3), (t_1Lf_2t_3).\ (t_1Rf_2t_3), (t_1t_2Lf_3), (t_1t_2Rf_3).$
2 used	$(e_1t_2t_3), (t_1e_2t_3), (t_1t_2e_3),\ (Lf_1Rf_2t_3), (Lf_1Lf_2t_3), (Rf_1Rf_2t_3),$
	$(Rf_1t_2Lf_3), (Rf_1t_2Rf_3), (t_1Lf_2Lf_3),\ (t_1Lf_2Rf_3), (t_1Rf_2Rf_3).$
3 used	$(e_1Rf_2t_3), (e_1t_2Lf_3), (Lf_1e_2t_3),\ (t_1e_2Rf_3), (Rf_1t_2e_3), (t_1Lf_2e_3).$

Table 1
States of the 3 Dining Philosophers CTMC

$$M_1 : \begin{cases} S_1 = A_{10} \cup A_{11} \cup A_{13} \cup A_{14} & \text{state-space} \\ A_{10} = \{think_1\} & \text{no resource states} \\ A_{11} = \{Rfork_1\} & \text{resource 1 states} \\ A_{13} = \{Lfork_1\} & \text{resource 3 states} \\ A_{14} = \{eat_1\} & \text{resource 4 states} \\ R_1 = \{fork_1, fork_3, eat\} & \text{shared resource} \\ C_{11} = \{M_2\}, C_{13} = \{M_3\}, C_{14} = \{M_2, M_3\} & \text{competing processes} \end{cases}$$

$$M_2 : \begin{cases} S_2 = A_{20} \cup A_{21} \cup A_{22} \cup A_{24} \\ A_{20} = \{think_2\} \\ A_{21} = \{Lfork_2\} \\ A_{22} = \{Rfork_2\} \\ A_{24} = \{eat_2\} \\ R_2 = \{fork_1, fork_2, eat\} \\ C_{21} = \{M_1\}, C_{22} = \{M_3\}, \\ C_{24} = \{M_1, M_3\} \end{cases} \qquad M_3 : \begin{cases} S_3 = A_{30} \cup A_{32} \cup A_{33} \cup A_{34} \\ A_{30} = \{think_3\} \\ A_{32} = \{Lfork_3\} \\ A_{33} = \{Rfork_3\} \\ A_{34} = \{eat_3\} \\ R_3 = \{fork_2, fork_3, eat\} \\ C_{32} = \{M_2\}, C_{33} = \{M_1\}, \\ C_{34} = \{M_1, M_2\} \end{cases}$$

State-space S of the composed CTMC M is obtained from (2.1) straightforwardly. The states corresponding to a breach of the mutually exclusion are the states in which at least two philosophers uses the resource *eat* or at least one of the forks is used by the two adjacent philosophers.

Also circular blocking is easy to identify in this example. The philosophers are blocked when each of them has the corresponding right or left fork, these states are given by: $CB = \big(A_{11} \times A_{22} \times A_{33}\big) \cup \big(A_{13} \times A_{21} \times A_{32}\big)$. Finally we report that for the 3 philosophers example $|ME| = 40$ and $|S| = \Pi_k|S_k| - |ME| = 64 - 40 = 24$, i.e., the composed CTMC consists of 24 states which are listed in Table 1.

3 Compositional CSL model checking for Boucherie processes

The Continuous Stochastic Logic (CSL) [1,2], is a formal language for expressing properties of a system modelled in terms of a labelled CTMC.

Given a set AP of atomic propositions, a labelled CTMC model $M = (S, Q, L)$ can be verified against properties expressed in CSL formulae. The syntax of CSL state-formulae (ϕ) and path-formulae (φ) is

$$\phi := a \mid tt \mid \neg\phi \mid \ \phi \wedge \phi \mid \mathcal{S}_{\unlhd p}(\phi) \mid \mathcal{P}_{\unlhd p}(\varphi) \tag{3.1}$$

$$\varphi := X^{I}\ \phi \mid \phi\ U^{I}\phi \tag{3.2}$$

where $a \in AP$, $p \in [0,1]$, $\unlhd \in \{<, \leq, >, \geq\}$ and $I \subseteq \mathbb{R}_{\geq 0}$ is a non empty interval.

The semantics of CSL formulae is defined in terms of two probability measures: the steady-state probability and the paths probability. $\pi^{M}(s, \phi)$ denotes the probability that, in the long-run, a state where ϕ is true has been reached given that s was the starting state. State s satisfies steady-state formula $\mathcal{S}_{\unlhd p}(\phi)$ iff $\pi^{M}(s, \phi) \unlhd p$. Similarly $Prob^{M}(s, \varphi)$ denotes the probability of paths in M with initial state s that satisfy φ (where φ is built on the time-bounded extension of the standard Next and Until path operators [8]). A path formula $\mathcal{P}_{\unlhd p}(\varphi)$ is satisfied by state s iff $Prob^{M}(s, \varphi) \unlhd p$. Formally:

$$\begin{array}{ll} s \models tt \text{ forall } s \in S & s \models \phi' \wedge \phi'' \text{ iff } s \models \phi' \wedge s \models \phi'' \\ s \models a \text{ iff } a \in L(s) & s \models \neg\phi \text{ iff } s \not\models \phi \\ s \models \mathcal{S}_{\unlhd p}(\phi) \text{ iff } \pi^{M}(s, \phi) \unlhd p & s \models \mathcal{P}_{\unlhd p}(\varphi) \text{ iff } Prob^{M}(s, \varphi) \unlhd p \end{array}$$

Below we report equation (3.3) (taken from [2]) for computing the probability measure of time-bounded Next formulae. It will be referred in the remainder.

$$Prob(s, X^{I}(\psi)) = e_{I}(s) \cdot \sum_{s' \models \psi} P(s, s') \tag{3.3}$$

We consider a K-dimensional labelled Boucherie CTMC $M = (S, Q, L)$ with states labelled according to their projections, which is: $L : S \rightarrow 2^{AP} : L(s) = \cup_{k=1}^{K} L_k(s_k)$. We take into account a restricted CSL syntax in which nesting of probabilistic operators is not permitted. The resulting syntax is given by:

$$\begin{array}{lll} \phi ::= \ \ \psi \mid \xi \mid \varphi \mid \omega \mid \phi \wedge \phi \mid \neg\phi & \varphi ::= \mathcal{P}_{\unlhd p}(X^{I}(\psi)) & \xi ::= \mathcal{S}_{\unlhd p}(\psi) \\ \psi ::= \ \ tt \mid a \mid \psi \wedge \psi \mid \neg\psi & \omega ::= \mathcal{P}_{\unlhd p}(\psi\ U\ \psi) & \end{array}$$

The state formulae ϕ can be categorised according to the atomic propositions they are built upon. A state formula, ϕ, in which all the atomic propositions belong to the same component, will be referred to as *single component formula*. A state formula, ϕ, in which the atomic propositions refer to more than one component, will be called *global formula*.

We observe that compositional semantics of simple boolean formulae (ψ) is a straightforward consequence of the "decomposed" labelling of M. For example, with respect to a 3-dimensional Boucherie CTMC we have that $(s_1, s_2, s_3) \models a_1 \wedge a_2 \wedge a_3$ if and only if $s_1 \models_1 a_1$, $s_2 \models_2 a_2$ and $s_3 \models_3 a_3$.

In the following we introduce the idea of k-move and devise its probability (Lemma 3.1). We refer to a transition $Q(s, s')$ as a k-move if it corresponds to a change of state of component k. We observe that the probability of observing a k-move in state $s = (s_1, \ldots, s_k, \ldots, s_K)$, denoted $p^{k}(s)$, is given by:

$$p^{k}(s) = \begin{cases} \frac{E_k(s_k)}{\sum_{j \in \overline{B(s)}} E_j(s_j)} & \text{if } k \in \overline{B(s)} \\ 0 & \text{if } k \in B(s) \end{cases} \tag{3.4}$$

where $B(s) = \{\hat{k} \in \{1, \ldots, K\} : \exists \hat{i} \in R_{\hat{k}}, \exists k' \in C_{\hat{k}\hat{i}} \wedge s_{k'} \in A_{k'\hat{i}}\}$ is the set of components that are blocked in s. We shall refer to states s for which $B(s) = \emptyset$ as a *globally* non-blocking states as opposed to *partially* blocking states, for which $B(s) \neq \emptyset$[4]. Note that (3.4) states that there's a null probability of observing a k-move in any state such that one amongst the competitors of k is using a resource that k is competing for. If that is not the case, instead, the probability of observing a k-move depends on how many amongst the remaining components are free to move (i.e., not blocked because one of the resource they compete for is in use by someone else). Note that if k is the only non-blocked component (i.e., $\overline{B(s)} = \{k\}$), then such probability is equal to 1.

The following lemma states the relation between the embedded transition probabilities of the composed process and the embedded transition probabilities of the components.

Lemma 3.1 *The probability of a k-move from state s to s' is equal to the probability of the corresponding k-projection ($s_k \rightarrow s'_k$), weighted by the probability of observing a k-move in s. (i.e., the embedded transition matrix P of a Boucherie process M, is a factor of its k-projection).*

$$P(s, s') = P_k(s_k, s'_k) \cdot p^k(s) \tag{3.5}$$

In the rest of this section we report that in several cases it can be decided in a compositional way if a given state of the composed process satisfies a state formula or not. In particular, we take into account three types of formulae: Section 3.1 deals with single component Until formulae; Section 3.2 discusses single component Next formulae; Section 3.3 considers global Next formulae.

Essentially, we show that, given a path formula of one of the above three types, the probability of the paths satisfying it can be computed in terms of some derived formulae over the components.

For what concerns formulae involving the steady state operator, since steady state probabilities of the states in the composed process is of product form, their evaluation is straightforward.

3.1 Single component Until formulae

The following theorem provides a relation between the composed process and its components for untimed paths.

Theorem 3.2 *Consider a finite untimed path σ_k over one of the K components. The probability of observing a path σ in the composed CTMC whose k-projection is σ_k is equal to the probability of observing the path σ_k in the CTMC M_k, i.e.,*

$$Prob\{\sigma : Proj_k(\sigma) = \sigma_k\} = Prob_k\{\sigma_k\}.$$

Intuitively, Theorem 3.2 holds because in a Boucherie process a component can block another for a given amount of time but it cannot change the way the other component chooses its next state. A formal proof is given in the appendix.

[4] Note that S is accordingly partitioned.

Theorem 3.2 cannot be generalised to any timed path. Consider the model of the three philosophers and the path $\sigma = (think_1,\ think_2,\ think_3),\ [1,2],\ (Rfork_1,\ think_2,\ think_3)$. Since in state $(think_1, think_2, think_3)$ any of the philosophers can move and since either philosopher 2 and 3 can move into a state in which philosopher 1 is blocked, then the probability of σ is not equal to the probability of the path $(think_1), [1,2], (Rfork_1)$ in M_1. On the other hand, there exist paths whose probability can be computed on the single component. E.g., since only philosopher 1 can move in $(eat_1, think_2, think_3)$, we have that $Prob((eat_1, think_2, think_3), [1,2], (think_1,\ think_2, think_3)) = Prob_1((think_1), [1,2], (eat_1))$.

The following theorem is a direct consequence of Theorem 3.2.

Theorem 3.3 *For a state $s = (s_1, \ldots, s_k, \ldots, s_K)$ of a K-dimensional Boucherie CTMC and a single component untimed Until formula ω_k, then*

$$(s_1, \ldots, s_k, \ldots, s_K) \models \omega_k \iff s_k \models_k \omega_k$$

As a consequence of Theorem 3.3, a single component Until formula can simply be checked on the involved component. Finally, we briefly note that Theorem 3.3 can be generalised to nested single component path formulae as well with the restriction that there is not Next operator in the formula.

3.2 Single component Next formulae

Assume now that we have a single component Next formula $X^I(\psi_k)$ that refers to component k. The following theorem shows that checking $X^I(\psi_k)$ against the composed process is equivalent to checking a similar Next formula on component k.

Theorem 3.4 *For $s = (s_1, \ldots s_k, \ldots s_K)$ a state of a K-dimensional Boucherie CTMC, $I = [a,b] \subseteq \mathbb{R}^+$, the probability measure of a time bounded single-component Next formula $(X^I \psi_k)$ is given by:*

$$Prob\left(s, X^I(\psi_k)\right) = p^k(s) \cdot Prob_k\left(s_k, X^{\frac{1}{p^k(s)} \cdot I}(\psi_k)\right) + (1 - p^k(s)) \cdot e_I(s) \cdot Prob_k(s_k, \psi_k) \qquad (3.6)$$

where $\frac{1}{p^k(s)} \cdot I = \left[\frac{a}{p^k(s)}, \frac{b}{p^k(s)}\right]$. *is a shifted time-interval.*

Proof. see Appendix.

The above result indicates that reasoning about temporal Next properties in a Boucherie framework can be done in a decomposed fashion. In particular the verification of a single-component Next formula bounded by I against M boils down (in the worst case) to the verification of the same Next formula with a shifted bounding

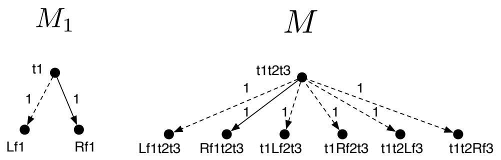

Fig. 2. Paths satisfying a Single-component Next formula

interval I'. Intuitively such time-shift can be explained as a (stochastic) compensation of the decreased concurrency: when we move the focus of our attention from a state $(s_1, \ldots, s_K)$ of M to its projection s_k on M_k, we essentially decrease the number of enabled transitions, hence the likelihood of leaving s_k within a given delay increases.

Example. Let us consider the 3 Dining Philosophers and suppose we are interested in the probability that: from the initial state $(t_1t_2t_3)$ of M, in one-step, the resource $fork_1$ gets occupied by M_1 with a delay in the interval $I = [2, 5]$. In CSL this can be expressed through the formula $\varphi_1 \equiv X^{[2,5]}(Rf_1)$. To describe the relationship between the probability measure of φ_1 wrt component M_1 (i.e. $Prob_1(t1, \varphi_1)$) and wrt the composed CTMC M (i.e. $Prob(t1t2t3, \varphi_1)$) we refer to Figure 2 which shows the unfolding of paths for M_1 and M (paths satisfying $X(Rf_1)$ are the non-dashed ones). Since $E(t_1t_2t_3) = 6$ and $e_I(t_1t_2t_3) = (e^{-2\cdot6} - e^{-5\cdot6})$, then by application of (3.3) wrt M we have that $Prob(t_1t_2t_3, X^{[2,5]}(Rf_1)) = (e^{-12} - e^{30}) \cdot 1/6$. On the other hand wrt to M_1, $E_1(t_1) = 2$ and $e_I(t_1) = (e^{-2\cdot2} - e^{-5\cdot2})$, thus from (3.3) we have that $Prob_1(t_1, X^{[2,5]}(Rf_1)) = (e^{-4} - e^{10}) \cdot 1/2$. Since component M_1 is not blocked in $t_1t_2t_3$ and the probability of a 1-move is $p^1(t_1t_2t_3) = 1/3$, then also $Prob(t_1t_2t_3, X^{[2,5]}(Rf_1)) = 1/3 \cdot 1/2(e^{-3\cdot4} - e^{-3\cdot10}) = p^1(t_1t_2t_3) \cdot Prob_1(t_1, X^{[6,15]}Rf_1)$.

3.3 Global Next formulae

Let ψ be a global formula (i.e. a formula containing atomic propositions that refer to at least 2 different components) written in disjunctive normal form: $\psi = \bigvee_i \bigwedge_j a_{ij}$. Each conjunct a_{ij} is a (possibly negated) atomic proposition referring to one component (i.e., $a_{ij} \in AP_k$ for some k). We consider the Next formula $X^I(\psi)$, where $I = [a, b] \subseteq \mathbb{R}^+$ is a continuous time interval. Given a state $s = (s_1, \ldots s_K)$ of M then for every component M_k we define the (state-dependent) formula

$$\xi_k(s) = \begin{cases} \neg tt & \text{if } \forall i\ \exists j : a_{ij} \in AP_l, l \neq k \wedge s_l \not\models a_{ij} \\ tt & \text{if } \exists i\ \forall j : a_{ij} \in AP_l, l \neq k \wedge s_l \models a_{ij} \\ \bigvee_{i:\forall j', a_{ij'} \in AP_l, l\neq k, s_l \models a_{ij'}} \ \bigwedge_{j: a_{ij} \in AP_k} a_{ij} & \text{otherwise} \end{cases} \tag{3.7}$$

whose meaning is as follows. Assuming that it will be component k to move, the next state of the composed model will satisfy ψ if and only if the next state of component k will satisfy ξ_k.

Let us consider a state (s_1, s_2, s_3) of a 3-dimensional CTMC and the formula: $\psi = a_1 \wedge a_2 \wedge b_1 \wedge a_3 \ \bigvee \ a_1 \wedge c_1 \wedge c_2 \wedge d_2 \wedge d_3 \ \bigvee \ c_3 \wedge b_2$ and let assume that $s_1 \models a_1 \wedge b_1 \wedge c_1$, $s_2 \models a_2$, and $s_3 \models c_3 \wedge d_3$. If none of the processes is blocked three different moves must be considered.

- 1-move: since $s_3 \not\models a_3$ and $s_2 \not\models d_2$ and $s_2 \not\models b_2$ no matter what 1-move we consider ψ will not be satisfied in the resulting next state of the composed process. Accordingly $\xi_1(s_1, s_2, s_3) = \neg tt$.
- 2-move: thanks to component 1 and component 3, the next composed state will surely satisfy $a_1 \wedge b_1 \ \bigvee \ a_1 \wedge c_1 \wedge d_3 \ \bigvee \ c_3$. It follows that if and only if the next state of component 2 satisfies either $(c_2 \wedge d_2)$ or b_2, then the next composed state satisfies ψ. It results in $\xi_2(s_1, s_2, s_3) = (c_2 \wedge d_2) \vee b_2$.

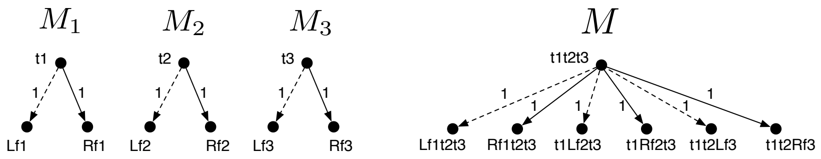

Fig. 3. Paths satisfying a Global Next formula

- 3-move: thanks to component 1 and component 2, the next composed state will surely satisfy $a_1 \wedge a_2 \wedge b_1 \bigvee a_1 \wedge c_1$. It follows that if and only if the next state of component 3 satisfies a_3, then the next composed state satisfies ψ. Accordingly $\xi_3(s_1, s_2, s_3) = a_3$.

If a k-move happens, in order to satisfy $X^I\psi$, the next state of component k has to satisfy $\xi_k(s_1, \ldots, s_K)$ and the transition has to happen within I. The resulting theorem is as follows.

Theorem 3.5 *For a state $s = (s_1, \ldots s_k, \ldots s_K)$ of a K-dimensional Boucherie CTMC, the probability measure of a time bounded global Next formula $(X^I\psi)$ is decomposed as follows:*

$$Prob(s, X^I\psi) = \sum_{k \notin B(s)} p^k(s) Prob_k(s_k, X^{\frac{1}{p^k(s)}I}\xi_k(s_1, \ldots, s_K)) \tag{3.8}$$

Proof. The proof of this theorem can be constructed as the proof of Theorem 3.4.

Example. Again let us refer to the 3 Dining Philosophers and suppose we are interested in the probability that, from the initial state, the first move of any philosopher will be to get hold of his right fork with a delay in $[2, 5]$. This is captured by the global-Next formula $\varphi \equiv (X^{[2,5]}(Rf_1 \vee Rf_2 \vee Rf3))$. Figure 3 shows the unfolding of paths generating at the initial state of each component M_k and of the composed process M (non-dashed lines denote paths satisfying $\varphi \equiv (X^{[2,5]}(Lf_1 \vee Lf_2 \vee Lf3)))$. By application of (3.3) we obtain straightforwardly

$$Prob(t_1t_2t_3, X^{[2,5]}(Rf_1 \vee Rf_2 \vee Rf3)) = 1/2 \cdot (e^{-2\cdot 6} - e^{-5\cdot 6}).$$

Now we apply theorem 3.5. Note that in the initial state $t_1t_2t_3$ every component is free to move. From (3.7) we have $\xi_1(t_1t_2t_3) = Rf_1$, $\xi_2(t_1t_2t_3) = Rf_2$ and $\xi_3(t_1t_2t_3) = Rf_3$. Since the probability of a k-move in $t_1t_2t_3$ is $p^k(t_1t_2t_3) = 1/3$ for all $k \in \{1, 2, 3\}$, the right end of (3.8) is $\sum_{k=1}^{3} 1/3 \cdot Prob_k(t_k, X^{[2\cdot 3, 5\cdot 3]}Rf_k)$. Since $Prob_k(t_k, X^{[6,15]}Rf_k) = 1/2 \cdot (e^{-6\cdot 2} - e^{-15\cdot 2})$ then we have

$$\begin{aligned}\sum_{k=1}^{3} 1/3 \cdot Prob_k(t_k, X^{[6,15]}Rf_k) = 1/2(e^{-6\cdot 2} - e^{-15\cdot 2})& \\ = Prob(t_1t_2t_3, X^{[2,5]}(Rf_1 \vee Rf_2 \vee Rf3))&\end{aligned}$$

3.4 Global (untimed) Until Formulae

In this section we consider Until formulae that refer to more than one component, like, for example, the formula $((think_1 \wedge think_2) U eat_3)$ which refer to the likelihood that philosopher M_3 wins the competition with both M_1 and M_2. Unfortunately the derivation of a compositional semantics, similar to the one demonstrated for the single-component Until and Next formulae, is not an easy task in this case. In fact, if the decomposed semantics of single-component Until formulae is a straightforward consequence of the independence of the components CTMCs, the benefit of such independence is lost when we have to look at paths that must simultaneously fulfil conditions on several components, as is the case with global Until formulae.

Nevertheless the possibility for a decomposed verification of global Until formulae may be suggested by observing the following property: given that φ is a global Until formula and σ a path in M that satisfies it, then, for each component k that is referred to by φ, σ_k, the k-projection of σ, satisfies a single-component Until formula φ_k which is derived from φ. E.g., if $\varphi \equiv ((think_1 \wedge think_2) U eat_3)$, the following single-component Until formulae can be derived:

$$\varphi_1 \equiv (think_1 \ U \ true), \ \varphi_2 \equiv (think_2 \ U \ true), \ \varphi_3 \equiv (true \ U \ eat_3)$$

and, $\sigma \models \varphi \implies \sigma_k \models \varphi_k$, $\forall k \in \{1, 2, 3\}$. Unfortunately, we are actually looking for a reversed type of implication, which is: we would like to prove that φ_k formulae exist such that verifying them against the components M_k is equivalent to verify φ against M. To do that, we would need to derive an *aggregation model* that describes how to combine the (local) probabilities of each φ_k, so that their combination is equivalent to the probability of φ in M. The derivation of such aggregation model is an hard task and we do not have a solution for it as yet.

To overcome that difficulty we propose an alternative approach. Rather than looking at a (decomposed) semantics equivalence for Until formulae, we see if we can exploit the inherent compositional description of a Boucherie CTMC within the CSL model checking algorithm itself.

For a state s of a labelled CTMC $M = (S, Q, L)$ the probability of satisfying an un-timed Until formula $(\phi \ U \ \psi)$ in s is computed through the following recursive function [2]:

$$Prob(s, (\phi \ \ U \ \psi)) = \begin{cases} 1 & \text{if } s \models \psi \\ \sum_{s' \in S} P(s, s') \cdot Prob(s', (\phi \ U \ \psi)) & \text{if } s \models \phi \wedge \neg\psi \\ 0 & otherwise \end{cases}$$

The above function can be adapted to a Boucherie CTMC as

$$Prob(s, (\phi \ \ U \ \psi)) = \begin{cases} 1 & \text{if } s \models \psi \\ \sum_{k \in \overline{B(s)}} \sum_{s'_k : s'_k \in S_k, s.s'_k \models \phi} & \text{if } s \models \phi \wedge \neg\psi \\ \quad p^k(s) P_k(s_k, s'_k) \cdot Prob(s.s'_k, (\phi \ U \ \psi)) & \\ 0 & otherwise \end{cases}$$

where $s.s'k$ denotes the state obtained from s by substituting s_k by s'_k.

The above equation shows that the probability of Until formulae referring to a Boucherie CTMC can be established without having to resort to the probability transition matrix P of the composed model. Note also that in the sum we consider only those states for which $s.s'_k \models \phi$ which allows for checking in a compositional manner which are those states of the composed model for which the recursion must be computed. As a result the memory requirement of the model checking algorithm for Until formulae can be lowered with respect to the “uncompositional” approach.

3.5 Timed Until formulae

We have shown in Section 3.1 that a single component untimed Until formula referring to component M_k can be verified without considering the other components of the model. When the formula is timed the way in which the other components block and hence "delay" component M_k must be taken into account.

In a given resource allocation situation, the transient behaviour of the composed model can be computed in product form. As a consequence, if a formula is satisfied only by paths along which the resource allocation does not change, then compositional verification is possible. This is rarely the case and hence we have to compute probability of path along which transient probabilities are not of product form. Even in this case however we can take advantage of the modular construction of the model for what concerns the storage of Q, the rate matrix of the composed model. In particular, entries of Q can be computed on the fly based on the rate matrices of the components, $Q_i, 1 \leq i \leq K$. This approach, which is based on Kronecker algebra (see, for example, [5] for details), results in significant memory saving if the composed model is large. Having the compact representation of the model, model checking techniques for large structured CTMCs can be applied [4].

The state space of the model can be divided into macrostates. Each macrostate is characterised by the current resource usage of the components described by a vector $|r_1, \dots, r_K|$ where $r_j = 0$ if component j does not use any shared resource and $r_j > 0$ indicates that resource r_j is in use by component j.

Then the rate matrix of the Markov chain of the composed model can be constructed as a block matrix in which block (i, j) contains transitions from states of macrostate i to states of macrostate j.

Before proceeding with the description of the blocks of Q we introduce the following notation. $Q_k^{i,j}$ is the submatrix of Q_k in which rows are selected according to states in A_{ki} and columns according to states in A_{kj}. 0_k^i and I_k^i are the zero and the identity matrix of size $|A_{ki}| \times |A_{ki}|$, respectively.

Two kinds of blocks must be distinguished because two kinds of transitions can occur: either a transition inside the macrostate or a transition that leads to a different macrostate. The block which describes the transitions inside the macrostate is given as

$$\bigoplus_{i=1}^{K} A_i \text{ with } A_i = \begin{cases} Q_i^{r_i,r_i} & \text{if component } i \text{ is not blocked} \\ 0_i^{r_i} & \text{if component } i \text{ is blocked} \end{cases} \tag{3.9}$$

where $\bigoplus$ is the Kronecker sum operator. The block that describes transitions from macrostate $r_1, \dots, r_j, \dots, r_n$ to macrostate $r_1, \dots, r_j', \dots, r_n$ is

$$\bigotimes_{i=1}^{n} B_i \quad \text{with} \quad B_i = \begin{cases} Q_i^{r_i,r_i'} & \text{if } i = j \\ I_i^{r_i} & \text{if } i \neq j \end{cases} \tag{3.10}$$

where $\bigotimes$ is the Kronecker product operator.

As an example we consider the model of the dining philosophers in a slightly more complicated version. In particular, every state of the philosopher will be represented by 2 states of the corresponding Markov chain. The rate matrix of the chain that corresponds to philosopher $i, 1 \leq i \leq 3$, denoted by Q_i is

$$\begin{vmatrix} 0 & q_{i,12} & q_{i,13} & q_{i,14} & q_{i,15} & q_{i,16} & 0 & 0 \\ q_{i,21} & 0 & q_{i,23} & q_{i,24} & q_{i,25} & q_{i,26} & 0 & 0 \\ 0 & 0 & 0 & q_{i,34} & 0 & 0 & q_{i,37} & q_{i,38} \\ 0 & 0 & q_{i,43} & 0 & 0 & 0 & q_{i,47} & q_{i,48} \\ 0 & 0 & 0 & 0 & 0 & q_{i,56} & q_{i,57} & q_{i,58} \\ 0 & 0 & 0 & 0 & q_{i,65} & 0 & q_{i,67} & q_{i,68} \\ q_{i,71} & q_{i,72} & 0 & 0 & 0 & 0 & 0 & q_{i,78} \\ q_{i,81} & q_{i,82} & 0 & 0 & 0 & 0 & q_{i,87} & 0 \end{vmatrix}$$

In states 1 and 2 the philosopher thinks, in states 3-4 has only the left fork, in states 5 and 6 has only the right fork and in state 7 and 8 eats.

Let us consider a few blocks of Q. The transitions inside the macrostate in which all the philosophers think can be written as $Q_1^{0,0} \bigoplus Q_2^{0,0} \bigoplus Q_3^{0,0}$ where $Q_i^{0,0}$ is given in (3.11).

The transition inside the macrostate in which philosopher 1 has his left fork and the other two are in thinking phase can be computed as $Q_1^{3,3} \bigoplus 0_2^0 \bigoplus Q_3^{0,0}$ where we have a zero matrix because philosopher 2 is blocked and $Q_1^{3,3}$ is as in (3.11).

The transitions that takes from the macrostate in which everybody thinks into the macrostate in which philosopher 1 has his right (resource 1) fork are given by $Q_1^{0,1} \bigotimes I_2^0 \bigotimes I_3^0$ with $Q_1^{0,1}$ given in (3.11).

$$Q_i^{0,0} = \begin{vmatrix} 0 & q_{i,12} \\ q_{i,21} & 0 \end{vmatrix} \quad Q_1^{3,3} = \begin{vmatrix} 0 & q_{1,34} \\ q_{1,43} & 0 \end{vmatrix} \quad Q_1^{0,1} = \begin{vmatrix} q_{1,15} & q_{1,16} \\ q_{1,25} & q_{1,26} \end{vmatrix} \tag{3.11}$$

Let us consider now the transition matrix of the embedded DTMC of the process, P. Note that P heavily depends on the transition rates of the individual components because "fast" components make moves with higher probability than "slow" components. From the embedded DTMCs of the components, $P_i, 1 \leq i \leq K$, the transition rates cannot be recovered. For this reason P cannot be built based on the individual embedded DTMCs. We can still however compute entries of P without storing the whole matrix based on the fact that $P = I - (\mathrm{diag}(Q))^{-1}Q$ where $\mathrm{diag}(Q)$ is the diagonal matrix of Q.

4 On the gain of decomposed model checking

So far we have seen that the verification of a certain type of CSL formulae against a Boucherie CTMC M is equivalent to the verification of some derived formulae against the component CTMCs M_k. The obvious advantage of such an approach is that if the components processes are smaller than the composed one (which almost always is the case), then the verification of large Boucherie processes through model checking can benefit of that.

Although a precise evaluation of the advantage of such decomposed approach depends very much on the considered model and the formulae to check, we can make some general considerations. If verifying a formula ϕ against M is equivalent to verifying a derived formula ϕ' against M_k, then the gain is given by the state-space dimension difference $|S| - |S_k|$: the larger the difference the greater the gain. By definition of the Boucherie CTMC state-space (2.1) we observe that the number of states in M (hence the difference $|S| - |S_k|$) depends on the following.

- K: **the dimension of the Boucherie framework** (i.e., the number of components): the larger the dimension, the greater the gain.
- C: **the level of competition of the framework**: this, in turn, depends on both: the number of shared resources (I), and the distribution of competition amongst them (i.e. the sets A_{ki}). We may say that a resource that is shared by 2 processes has a lower level of competition that a resource which is shared by 3 processes. Similarly we may say that a framework in which processes occupy shared resources in single states only, has a lower level of competition than a framework in which processes hold shared resources through a number of different states. As a result the larger the level of competition of a framework, the greater is the gain.

Let us consider the example of the 3 Dining Philosophers and suppose we are interested in determining what is the probability that Philosopher 1 will eat (at some point), which corresponds to the CSL single-component formula $\varphi_1 \equiv (\mathit{true}\ U\ \mathit{eat}_1)$. Because of Theorem 3.2 we know that it is sufficient to verify φ_1 directly on the 4-states CTMC M_1, rather than against the 24- states M, which implies a about 83% gain in state-space. Finally we observe that, in some cases, the verification of ϕ against M corresponds to the verification of a number of ϕ'_k against M_k (e.g Global Next). To evaluate the gain in this case we need to consider the number of ϕ'_k we have to verify on each component. If the decomposition of ϕ results in n_k formulae ϕ'_k that has to be verified against component M_k then the gain of decomposed verification, in this case, is: $(|S| - \sum_{k\in\{1\ldots K\}} n_k \cdot |S_k|)$. For example, as we have seen before, the Global-Next formula $\varphi \equiv (X^{[2,5]}(Rf_1 \vee Rf_2 \vee Rf3))$ is decomposed into the following 3 single-component formulae: $\varphi_1 \equiv (X^{[6,15]}(Rf_1))$, $\varphi_2 \equiv (X^{[6,15]}(Rf_2))$ and $\varphi_3 \equiv (X^{[6,15]}(Rf_3))$ each of which has to be verified against the component it is referring to. Hence, the gain of the decomposed semantics, in this case, is $24 - 3 \cdot 4 = 12$ (i.e. 50% gain). Note that more significant figures would be obtained for more complex/realistic systems: the 3 Dining Philosophers model is indeed just a toy example.

5 Conclusion

We have studied a compositional approach to the probabilistic model checking problem. More precisely we have considered a family of compositional (product- form) CTMCs (i.e. the so-called Boucherie process), and we have shown how CSL model checking verification can be "decomposed" for (certain) formulae that are referred to such CTMCs. The obvious implication of such results is that the complexity of CSL model checking can be reduced by some order of magnitudes. The compositional results herein demonstrated are referred to CSL non-nested path-formulae only: the extension to nested path-formulae is part of our future goals. Furthermore we have argued that a decomposed CSL semantics cannot be derived for timed Until formulae because the transient behaviour of the Boucherie process is not of product form. However it seems possible to develop transient analysis technique that provides the transient behaviour based on computation on the single components. This technique which we aim to study in the future could lead to developing a compositional verification method for the segment of CSL formulae that we did not deal with in this paper.

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, Vol. 1:pp. 162–170, 2000.

[2] C. Baier, B. Haverkort, H. Hermann, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. on Software Eng.*, Vol. 29(6):pp. 524–541, June 2003.

[3] R. Boucherie. A characterisation of independence for competing Markov chains with applications to stochastic Petri nets. *IEEE Trans. on Software Eng.*, Vol. 20(7):pp. 536–544, 1994.

[4] P. Buchholz, J.-P. Katoen, P. Kemper, and C. Tepper. Model-checking large structured Markov chains. *J. Log. Algebr. Program.*, 56(1-2):69–97, 2003.

[5] P. Buchholz and P. Kemper. Kronecker based matrix representations for large markov models. In *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 256–295, 2004.

[6] C. Baier, E. Clarke, V. Hartonas-Garmhausen, M. Kwiatkowska, and M. Ryan. Symbolic model checking for probabilistic processes. In *Proc. of the 24th Int. Coll. on Automata, Languages and Programming*, volume 1256 of *LNCS*, pages pp. 430–440. Springer-Verlag, 1997.

[7] E. Clarke, D. Long, and K. McMillan. Compositional model checking. In *Proceedings of the Fourth Annual Symposium on Logic in computer science*, pages 353–362, Piscataway, NJ, USA, 1989. IEEE Press.

[8] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, Vol. 8(2):pp. 244–263, 1986.

[9] J.-P. Katoen, M. Kwiatkowska, G. Norman, and D. Parker. Faster and symbolic CTMC Model-Checking. In *Proceedings of the Joint International Workshop PAPM-PROBMIV*, volume 2165 of *LNCS*. L. de Alfaro S. Gilmore, 2001.

[10] J.R. Burch, E.M. Clarke, D.L. Dill, K.L. McMillan, and J. Hwang. Symbolic model cheking: 10^{20} states and beyond. In *In Proc. of LICS '90*. IEEE Computer Society Press, 1990.

[11] O. Kupferman and M. Y. Vardi. Module checking. In *CAV '96: Proceedings of the 8th International Conference on Computer Aided Verification*, pages 75–86, London, UK, 1996. Springer-Verlag.

[12] M. Sereno. Computational algorithms for product-form of competing markov chains. In *10th International Workshop on Petri Nets and Performance Models (PNPM 2003), Urbana, Illinois, USA*, pages 93–102. IEEE Press, sep 2003.

A Proofs

Proof of Theorem 3.2. On condition that the state of the composed process is $(s_1', \dots, s_k', \dots, s_K') \in S$, the probability that the first k-move leads the process to state $(s_1'', \dots, s_k'', \dots, s_K'')$ will be denoted by $F_k((s_1'', \dots, s_k'', \dots, s_K'')|(s_1', \dots, s_k', \dots, s_K'))$. On condition that the state of the composed process is $(s_1', \dots, s_k', \dots, s_K')$, the probability that the first k-move is the nth move will be denoted by $F_{k,n}((s_1', \dots, s_k', \dots, s_K'))$.

In any state of the composed process component k is either blocked or it can make a move. By construction of the Boucherie process

- component k will make a move in the future with probability one, i.e., $\sum_{i=1}^{\infty} F_{k,i}((s_1', \dots, s_k', \dots, s_K')) = 1$
- assuming that it is component k to make the move, it takes a move according to its own infinitesimal generator Q_k.

Based on the above facts

$$\begin{aligned} F_k((s_1'', \dots, s_k'', \dots, s_K'')|(s_1', \dots, s_k', \dots, s_K')) &= \sum_{i=1}^{\infty} F_{k,i}((s_1', \dots, s_k', \dots, s_K')) \frac{Q_k(s_k', s_k'')}{E_k(s_k')} \\ &= \frac{Q_k(s_k', s_k'')}{E_k(s_k')} \sum_{i=1}^{\infty} F_{k,i}((s_1', \dots, s_k', \dots, s_K')) = \frac{Q_k(s_k', s_k'')}{E_k(s_k')}, \end{aligned}$$

i.e., the next state of component k after the first k-move does not depend on the state of the other components. Further, the distribution of the next state of component k in the composed CTMC is identical to the distribution of the next state in the single CTMC which implies the theorem. □

Proof of Theorem 3.4. The probability of satisfying a time bounded next formula $(X^I \psi_k)$ in state s is given by: $Prob(s, X^I(\psi_k)) = e_I(s) \cdot \sum_{s' \models \psi_k} P(s, s')$ or, equivalently:

$$Prob\left(s, X^I(\psi_k)\right) \quad = \quad \left(e^{-a \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)} - e^{-b \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)}\right) \quad \cdot \quad \sum_{s' \models \psi_k} P(s, s'). \tag{A.1}$$

In order to calculate $Prob(s, X^I(\psi_k))$ the following cases have to be considered.

i- $k \in \overline{B(s)}$, $s_k \not\models_k \psi_k$: in this case the successors of s that may satisfy ψ_k are only those corresponding to a k-move. Hence, by application of Lemma 3.1 into (A.1) we have:

$$\begin{aligned} Prob(s, X^I(\psi_k)) &= (e^{-a \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)} - e^{-b \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)}) \cdot p^k(s) \cdot \sum_{s'_k \models_k \psi_k} P_k(s_k, s'_k) = \\ & p^k(s) \cdot \left(e^{-a \cdot \frac{E_k(s_k)}{p^k(s)}} - e^{-b \cdot \frac{E_k(s_k)}{p^k(s)}}\right) \cdot \sum_{s'_k \models_k \psi_k} P_k(s_k, s'_k) = p^k(s) \cdot Prob_k\left(s_k, X^{\frac{1}{p^k(s)}[a,b]}(\psi_k)\right). \end{aligned}$$

ii- $k \in \overline{B(s)}$, $s_k \models_k \psi_k$: in this case every successor of s that corresponds to a non-k-move satisfies ψ_k whereas a subset of those corresponding to a k-move satisfy ψ_k. Hence,

$$\begin{aligned} Prob\left(s, X^I(\psi_k)\right) &= \left(e^{-a \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)} - e^{-b \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)}\right) \cdot \\ & \left[\left(p^k(s) \cdot \sum_{s'_k \models_k \psi_k} P_k(s_k, s'_k)\right) + \left(\sum_{\bar{k} \neq k} p^{\bar{k}}(s) \cdot \sum_{s'_{\bar{k}} \in S_{\bar{k}}} P_{\bar{k}}(s_{\bar{k}}, s'_{\bar{k}})\right)\right] \\ & = p^k(s) \cdot Prob_k\left(s_k, X^{\frac{1}{p^k(s)}[a,b]}(\psi_k)\right) + (1 - p^k(s)) \cdot e_I(s). \end{aligned}$$

iii- $k \in B(s)$, $s_k \models_k \psi_k$: since component k is blocked, the path formula is satisfied if the composed process makes a move in I. We have that

$$Prob\left(s, X^I(\psi_k)\right) = \left(e^{-a \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)} - e^{-b \cdot \sum_{j \in \overline{B(s)}} E_j(s_j)}\right).$$

i, *ii* and *iii* demonstrate (3.6). □