Weak Markovian Bisimulation Congruences and Exact CTMC-Level Aggregations for Concurrent Processes

Marco Bernardo
Dipartimento di Scienze di Base e Fondamenti – Università di Urbino – Italy

We have recently defined a weak Markovian bisimulation equivalence in an integrated-time setting, which reduces sequences of exponentially timed internal actions to individual exponentially timed internal actions having the same average duration and execution probability as the corresponding sequences. This weak Markovian bisimulation equivalence is a congruence for sequential processes with abstraction and turns out to induce an exact CTMC-level aggregation at steady state for all the considered processes. However, it is not a congruence with respect to parallel composition. In this paper, we show how to generalize the equivalence in a way that a reasonable tradeoff among abstraction, compositionality, and exactness is achieved for concurrent processes. We will see that, by enhancing the abstraction capability in the presence of concurrent computations, it is possible to retrieve the congruence property with respect to parallel composition, with the resulting CTMC-level aggregation being exact at steady state only for a certain subset of the considered processes.

1 Introduction

Several Markovian behavioral equivalences (see [1] and the references therein) have been proposed in the literature for relating and manipulating system models with an underlying continuous-time Markov chain (CTMC) [15] semantics. However, only a few of them are provided with the useful capability of abstracting from internal actions. In particular, [3] has recently addressed the case in which internal actions are exponentially timed – rather than immediate like in [9] – by defining a weak Markovian bisimulation equivalence inspired by the weak (Markovian) isomorphism of [11]. The idea is to reduce to *individual* exponentially timed internal transitions all the *sequences* of exponentially timed internal transitions that traverse states enabling *only* exponentially timed internal actions, with the reduction preserving the average duration and the execution probability of the original sequences.

From a stochastic viewpoint, this reduction amounts to replacing hypoexponentially distributed durations with exponentially distributed durations having the same expected value. As a consequence, processes related by the weak Markovian bisimulation equivalence of [3] may not possess the same transient performance measures, unless they refer to properties of the form mean time to certain events. However, those processes certainly possess the same steady-state performance measures, because the aggregation induced by the considered equivalence on the CTMC underlying each process has been shown to be exact at steady state.

The weak Markovian bisimulation equivalence of [3] is not a congruence with respect to parallel composition, a fact that limits its usefulness for compositional state space reduction purposes. The contribution of this paper is to show that compositionality can be retrieved by enhancing the abstraction capability of the considered equivalence in the presence of parallel composition. The basic idea is allowing a sequence of exponentially timed internal transitions originated from a sequential process to be reduced also in the case in which that process is composed in parallel with other processes enabling *observable* actions. Unfortunately, there is a price to pay for achieving compositionality: exactness at

M. Massink and H. Wiklicky (Eds.):Workshop on Quantitative
Aspects of Programming Languages and Systems 2012 (QAPL 2012)
EPTCS 85, 2012, pp. 122–136, doi:10.4204/EPTCS.85.9

steady state will no longer hold for all processes, but only for processes with no synchronization at all and processes whose synchronizations do not take place right before the sequences to be reduced.

This paper is organized as follows. After introducing a Markovian process calculus in Sect. 2 and recalling strong and weak Markovian bisimilarity in Sect. 3, in Sect. 4 we develop a variant of weak Markovian bisimilarity that deals with parallel composition and we investigate its congruence and exactness properties. Finally, in Sect. 5 we provide some concluding remarks.

2 Concurrent Markovian Processes

In order to study properties such as congruence of the variant (to be defined) of the weak Markovian bisimilarity of [3], we introduce typical behavioral operators through a Markovian process calculus (MPC for short). In [3], we have considered sequential processes with abstraction built from operators like the inactive process, exponentially timed action prefix, alternative composition, recursion, and hiding. Here, we include parallel composition too, so as to be able to represent concurrent processes.

As usual, we denote the internal action by τ and we assume that the resulting concurrent processes are governed by the race policy: if several exponentially timed actions are simultaneously enabled, the action that is executed is the one sampling the least duration. We also assume that the duration of an action deriving from the synchronization of two exponentially timed actions is exponentially distributed with a rate obtained by applying (like, e.g., in [10]) some commutative and associative operation denoted by $\otimes$ to the rates of the two original actions.

Definition 2.1 Let $Act_{\mathrm{M}} = Name \times \mathbb{R}_{>0}$ be a set of actions, where $Name = Name_{\mathrm{v}} \cup \{\tau\}$ is a set of action names – ranged over by a, b – and $\mathbb{R}_{>0}$ is a set of action rates – ranged over by λ, μ, γ. Let *Var* be a set of process variables ranged over by X, Y. The process language $\mathscr{PL}_{\mathrm{M}}$ is generated by the following syntax:

P	$::=$	$\underline{0}$	inactive process
	$\mid$	$<a,\lambda>.P$	exponentially timed action prefix
	$\mid$	$P+P$	alternative composition
	$\mid$	X	process variable
	$\mid$	$\mathrm{rec}\, X : P$	recursion
	$\mid$	P/H	hiding
	$\mid$	$P \,\|_S\, P$	parallel composition

where $a \in Name$, $\lambda \in \mathbb{R}_{>0}$, $X \in Var$, and $H, S \subseteq Name_{\mathrm{v}}$. We denote by $\mathbb{P}_{\mathrm{M}}$ the set of closed and guarded process terms of $\mathscr{PL}_{\mathrm{M}}$ – ranged over by P, Q. ■

In order to distinguish between process terms such as $<a,\lambda>.\underline{0} + <a,\lambda>.\underline{0}$ and $<a,\lambda>.\underline{0}$, like in [3] the semantic model $[\![P]\!]_{\mathrm{M}}$ for a process term $P \in \mathbb{P}_{\mathrm{M}}$ is a labeled multitransition system that takes into account the multiplicity of each transition, intended as the number of different proofs for the transition derivation. The multitransition relation of $[\![P]\!]_{\mathrm{M}}$ is contained in the smallest multiset of elements of $\mathbb{P}_{\mathrm{M}} \times Act_{\mathrm{M}} \times \mathbb{P}_{\mathrm{M}}$ that satisfies the operational semantic rules in Table 1 – where $\{\_ \hookrightarrow \_\}$ denotes syntactical replacement – and keeps track of all the possible ways of deriving each of its transitions.

3 Strong and Weak Markovian Bisimulation Equivalences

The notion of strong bisimilarity for MPC is based on the comparison of exit rates [11, 10]. The exit rate of a process term $P \in \mathbb{P}_{\mathrm{M}}$ with respect to action name $a \in Name$ and destination $D \subseteq \mathbb{P}_{\mathrm{M}}$ is the rate at

$$
(\textsc{Pre}_{\mathrm{M}})\ \frac{}{<a,\lambda>.P \xrightarrow{a,\lambda}_{\mathrm{M}} P}
\qquad
(\textsc{Rec}_{\mathrm{M}})\ \frac{P\{\mathrm{rec}\,X : P \hookrightarrow X\} \xrightarrow{a,\lambda}_{\mathrm{M}} P'}{\mathrm{rec}\,X : P \xrightarrow{a,\lambda}_{\mathrm{M}} P'}
$$

$$
(\textsc{Alt}_{\mathrm{M},1})\ \frac{P_1 \xrightarrow{a,\lambda}_{\mathrm{M}} P'}{P_1 + P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'}
\qquad
(\textsc{Alt}_{\mathrm{M},2})\ \frac{P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'}{P_1 + P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'}
$$

$$
(\textsc{Hid}_{\mathrm{M},1})\ \frac{P \xrightarrow{a,\lambda}_{\mathrm{M}} P' \quad a \notin H}{P/H \xrightarrow{a,\lambda}_{\mathrm{M}} P'/H}
\qquad
(\textsc{Hid}_{\mathrm{M},2})\ \frac{P \xrightarrow{a,\lambda}_{\mathrm{M}} P' \quad a \in H}{P/H \xrightarrow{\tau,\lambda}_{\mathrm{M}} P'/H}
$$

$$
(\textsc{Par}_{\mathrm{M},1})\ \frac{P_1 \xrightarrow{a,\lambda}_{\mathrm{M}} P'_1 \quad a \notin S}{P_1 \,\|_S\, P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'_1 \,\|_S\, P_2}
\qquad
(\textsc{Par}_{\mathrm{M},2})\ \frac{P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'_2 \quad a \notin S}{P_1 \,\|_S\, P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P_1 \,\|_S\, P'_2}
$$

$$
(\textsc{Syn}_{\mathrm{M}})\ \frac{P_1 \xrightarrow{a,\lambda_1}_{\mathrm{M}} P'_1 \quad P_2 \xrightarrow{a,\lambda_2}_{\mathrm{M}} P'_2 \quad a \in S}{P_1 \,\|_S\, P_2 \xrightarrow{a,\lambda_1 \otimes \lambda_2}_{\mathrm{M}} P'_1 \,\|_S\, P'_2}
$$

Table 1: Structured operational semantic rules for MPC

which P can execute actions of name a that lead to D:

$$
rate(P,a,D) \;=\; \sum\{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists P' \in D.\, P \xrightarrow{a,\lambda}_{\mathrm{M}} P' \,|\!\}
$$

where $\{\!|$ and $|\!\}$ are multiset delimiters and the summation is taken to be zero if its multiset is empty. By summing up the rates of all the actions of P, we obtain the total exit rate of P, i.e., $rate_{\mathrm{t}}(P) = \sum_{a \in Name} rate(P,a,\mathbb{P}_{\mathrm{M}})$, which is the reciprocal of the average sojourn time associated with P.

Definition 3.1 An equivalence relation $\mathscr{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a Markovian bisimulation iff, whenever $(P_1,P_2) \in \mathscr{B}$, then for all action names $a \in Name$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathscr{B}$:

$$
rate(P_1,a,D) \;=\; rate(P_2,a,D)
$$

Markovian bisimilarity $\sim_{\mathrm{MB}}$ is the largest Markovian bisimulation. ■

As shown in [11, 10, 6, 7], the relation $\sim_{\mathrm{MB}}$ possesses the following properties:

- $\sim_{\mathrm{MB}}$ is a congruence with respect to all the operators of MPC as well as recursion.
- $\sim_{\mathrm{MB}}$ has a sound and complete axiomatization whose basic laws are shown below:

$(\mathscr{A}_{\mathrm{MB},1})$	$P_1 + P_2$	$=$	$P_2 + P_1$
$(\mathscr{A}_{\mathrm{MB},2})$	$(P_1 + P_2) + P_3$	$=$	$P_1 + (P_2 + P_3)$
$(\mathscr{A}_{\mathrm{MB},3})$	$P + \underline{0}$	$=$	P
$(\mathscr{A}_{\mathrm{MB},4})$	$<a,\lambda_1>.P + <a,\lambda_2>.P$	$=$	$<a,\lambda_1 + \lambda_2>.P$

The last one encodes the race policy and hence replaces the idempotency law $P + P = P$ valid for nondeterministic processes. The other laws are the usual distribution laws for the hiding operator and the expansion law for the parallel composition operator.

- $\sim_{\rm MB}$ induces a CTMC-level aggregation known as ordinary lumpability, which is exact both at steady state and at transient state.
- $\sim_{\rm MB}$ can be decided in polynomial time for all finite-state processes.

In [3], we have weakened the distinguishing power of $\sim_{\rm MB}$ by relating sequences of exponentially timed τ-actions to single exponentially timed τ-actions having the same average duration and execution probability as the sequences. Given $P \in \mathbb{P}_{\rm M}$, we say that P is stable if $P \xrightarrow{\tau,\lambda}\!\!\!\!\!\!\!/\;\;_{\rm M} P'$ for all λ and P', otherwise we say that it is unstable. In the latter case, we say that P is fully unstable iff, whenever $P \xrightarrow{a,\lambda}_{\rm M} P'$, then $a = \tau$. We denote by $\mathbb{P}_{\rm M,fu}$ and $\mathbb{P}_{\rm M,nfu}$ the sets of process terms of $\mathbb{P}_{\rm M}$ that are fully unstable and not fully unstable, respectively.

The most natural candidates as sequences of exponentially timed τ-actions to abstract are those labeling computations that traverse fully unstable states.

Definition 3.2 Let $n \in \mathbb{N}_{>0}$ and $P_1, P_2, \dots, P_{n+1} \in \mathbb{P}_{\rm M}$. A computation c of length n from P_1 to P_{n+1} having the form $P_1 \xrightarrow{\tau,\lambda_1}_{\rm M} P_2 \xrightarrow{\tau,\lambda_2}_{\rm M} \dots \xrightarrow{\tau,\lambda_n}_{\rm M} P_{n+1}$ is reducible iff $P_i \in \mathbb{P}_{\rm M,fu}$ for all $i = 1, \dots, n$. ■

If reducible, the computation c above can be reduced to a single exponentially timed τ-transition whose rate is obtained from the positive real value below:

$$probtime(c) \; = \; \left(\prod_{i=1}^{n} \frac{\lambda_i}{rate(P_i, \tau, \mathbb{P}_{\rm M})} \right) \cdot \left(\sum_{i=1}^{n} \frac{1}{rate(P_i, \tau, \mathbb{P}_{\rm M})} \right)$$

by leaving its first factor unchanged and taking the reciprocal of the second one. The value $probtime(c)$ is a measure of the execution probability of c (first factor: product of the execution probabilities of the transitions of c) and the average duration of c (second factor: sum of the average sojourn times in the states traversed by c).

The weak variant of $\sim_{\rm MB}$ defined in [3] is such that (i) processes in $\mathbb{P}_{\rm M,nfu}$ are dealt with as in $\sim_{\rm MB}$ and (ii) the length of reducible computations from processes in $\mathbb{P}_{\rm M,fu}$ to processes in $\mathbb{P}_{\rm M,nfu}$ is abstracted away while preserving the execution probability and the average duration of those computations. In the latter case, we need to lift measure *probtime* from individual reducible computations to multisets of reducible computations. Denoting by $reducomp(P, D, t)$ the multiset of reducible computations from $P \in \mathbb{P}_{\rm M,fu}$ to some P' in $D \subseteq \mathbb{P}_{\rm M}$ whose average duration is $t \in \mathbb{R}_{>0}$, we consider the following t-indexed multiset of sums of *probtime* measures:

$$pbtm(P, D) \; = \bigcup_{t \in \mathbb{R}_{>0} \text{ s.t. } reducomp(P,D,t) \neq \emptyset} \{\!| \sum_{c \in reducomp(P,D,t)} probtime(c) \, |\!\}$$

Definition 3.3 An equivalence relation $\mathcal{B} \subseteq (\mathbb{P}_{\rm M,nfu} \times \mathbb{P}_{\rm M,nfu}) \cup (\mathbb{P}_{\rm M,fu} \times \mathbb{P}_{\rm M,fu})$ is a weak Markovian bisimulation iff for all $(P_1, P_2) \in \mathcal{B}$:

- If $P_1, P_2 \in \mathbb{P}_{\rm M,nfu}$, then for all $a \in Name$ and equivalence classes $D \in \mathbb{P}_{\rm M} / \mathcal{B}$:
$$rate(P_1, a, D) \; = \; rate(P_2, a, D)$$
- If $P_1, P_2 \in \mathbb{P}_{\rm M,fu}$, then for all equivalence classes $D \in \mathbb{P}_{\rm M,nfu} / \mathcal{B}$:
$$pbtm(P_1, D) \; = \; pbtm(P_2, D)$$

Weak Markovian bisimilarity $\approx_{\rm MB}$ is the largest weak Markovian bisimulation. ■

Example 3.4 Typical cases of weakly Markovian bisimilar process terms are:

$$<\!\tau, \mu\!>.<\!\tau, \gamma\!>.Q \qquad <\!\tau, \gamma\!>.<\!\tau, \mu\!>.Q \qquad <\!\tau, \frac{\mu \cdot \gamma}{\mu + \gamma}\!>.Q$$

and:

$$<\tau,\mu>.(<\tau,\gamma_1>.Q_1 + <\tau,\gamma_2>.Q_2)$$
$$<\tau,\tfrac{\gamma_1}{\gamma_1+\gamma_2}\cdot\left(\tfrac{1}{\mu}+\tfrac{1}{\gamma_1+\gamma_2}\right)^{-1}>.Q_1 + <\tau,\tfrac{\gamma_2}{\gamma_1+\gamma_2}\cdot\left(\tfrac{1}{\mu}+\tfrac{1}{\gamma_1+\gamma_2}\right)^{-1}>.Q_2$$

and:

$$<\tau,\mu_1>.<\tau,\gamma>.Q_1 + <\tau,\mu_2>.<\tau,\gamma>.Q_2$$
$$<\tau,\tfrac{\mu_1}{\mu_1+\mu_2}\cdot\left(\tfrac{1}{\mu_1+\mu_2}+\tfrac{1}{\gamma}\right)^{-1}>.Q_1 + <\tau,\tfrac{\mu_2}{\mu_1+\mu_2}\cdot\left(\tfrac{1}{\mu_1+\mu_2}+\tfrac{1}{\gamma}\right)^{-1}>.Q_2$$

where $Q, Q_1, Q_2 \in \mathbb{P}_{\mathrm{M,nfu}}$ (see [3] for the details). ■

Similar to weak bisimilarity for nondeterministic processes, $\approx_{\mathrm{MB}}$ is not a congruence with respect to the alternative composition operator. This problem, which has to do with fully unstable process terms, can be prevented by adopting a construction analogous to the one used in [13] for weak bisimilarity over nondeterministic process terms. In other words, we have to apply the exit rate equality check also to fully unstable process terms, with the equivalence classes to consider being the ones with respect to $\approx_{\mathrm{MB}}$.

Definition 3.5 Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. We say that P_1 is weakly Markovian bisimulation congruent to P_2, written $P_1 \simeq_{\mathrm{MB}} P_2$, iff for all action names $a \in \mathit{Name}$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\approx_{\mathrm{MB}}$:

$$\mathit{rate}(P_1, a, D) = \mathit{rate}(P_2, a, D)$$

■

As shown in [3], the relation $\simeq_{\mathrm{MB}}$ possesses the following properties:

- $\simeq_{\mathrm{MB}}$ is the coarsest congruence – with respect to all the operators of MPC other than parallel composition, as well as recursion – contained in $\approx_{\mathrm{MB}}$.
- $\simeq_{\mathrm{MB}}$ has a sound and complete axiomatization over the set of sequential process terms (i.e., process terms with no occurrences of the parallel composition operator), whose basic laws are those of $\sim_{\mathrm{MB}}$ plus the following one (which includes the various cases shown in Ex. 3.4):

$$\boxed{\begin{array}{ll}(\mathscr{A}_{\mathrm{MB},5}) & <a,\lambda>.\sum\limits_{i\in I}<\tau,\mu_i>.\sum\limits_{j\in J_i}<\tau,\gamma_{i,j}>.P_{i,j} \;=\\ & \qquad <a,\lambda>.\sum\limits_{i\in I}\sum\limits_{j\in J_i}<\tau,\tfrac{\mu_i}{\mu}\cdot\tfrac{\gamma_{i,j}}{\gamma}\cdot\left(\tfrac{1}{\mu}+\tfrac{1}{\gamma}\right)^{-1}>.P_{i,j}\end{array}}$$

  where $I \neq \emptyset$ is a finite index set, $J_i \neq \emptyset$ is a finite index set for all $i \in I$, $\mu = \sum_{i\in I}\mu_i$, and $\gamma = \sum_{j\in J_i}\gamma_{i,j}$ for all $i \in I$.
- $\simeq_{\mathrm{MB}}$ induces a CTMC-level aggregation called W-lumpability, which is exact only at steady state and performs reductions consistent with $\mathscr{A}_{\mathrm{MB},5}$. Moreover, $\simeq_{\mathrm{MB}}$ preserves transient properties expressed in terms of the mean time to certain events.
- $\simeq_{\mathrm{MB}}$ can be decided in polynomial time only for those finite-state processes that are not divergent, i.e., that have no cycles of exponentially timed τ-transitions.

4 Compositionality for Concurrent Processes

The relation $\simeq_{\mathrm{MB}}$ is not a congruence with respect to the parallel composition operator, thus restricting the usefulness for compositional state space reduction purposes of the framework developed in [3].

Example 4.1 Assuming parallel composition to have lower priority than any other operator, it holds that:

$$<a,\lambda>.<\tau,\mu>.<\tau,\gamma>.\underline{0} \simeq_{\mathrm{MB}} <a,\lambda>.<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}$$

while:

$$<a,\lambda>.<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}<a',\lambda'>.\underline{0} \not\simeq_{\mathrm{MB}} <a,\lambda>.<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}<a',\lambda'>.\underline{0}$$

First of all, we note that:

$$<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}\ \not\approx_{\rm MB}\ <\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}$$

In fact, for $a' \neq \tau$ the two process terms are not fully unstable with:

$$\begin{array}{rcl} \mathit{rate}(<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},\tau,[<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}}) & = & \mu \\ \mathit{rate}(<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},\tau,[<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}}) & = & 0 \end{array}$$

On the other hand, for $a' = \tau$ the two process terms are fully unstable with:

$$\begin{array}{rcl} \mathit{pbtm}(<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},[\underline{0}\,\|_{\emptyset}\,\underline{0}]_{\approx_{\rm MB}}) & = & \{\!|\,(\frac{\mu}{\mu+\lambda'}\cdot\frac{\gamma}{\gamma+\lambda'})\cdot(\frac{1}{\mu+\lambda'}+\frac{1}{\gamma+\lambda'}+\frac{1}{\lambda'}), \\ & & (\frac{\mu}{\mu+\lambda'}\cdot\frac{\lambda'}{\gamma+\lambda'})\cdot(\frac{1}{\mu+\lambda'}+\frac{1}{\gamma+\lambda'}+\frac{1}{\gamma}), \\ & & (\frac{\lambda'}{\mu+\lambda'})\cdot(\frac{1}{\mu+\lambda'}+\frac{1}{\mu}+\frac{1}{\gamma})\,|\!\} \\ \mathit{pbtm}(<\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},[\underline{0}\,\|_{\emptyset}\,\underline{0}]_{\approx_{\rm MB}}) & = & \{\!|\,(\frac{\frac{\mu\cdot\gamma}{\mu+\gamma}}{\frac{\mu\cdot\gamma}{\mu+\gamma}+\lambda'})\cdot(\frac{1}{\frac{\mu\cdot\gamma}{\mu+\gamma}+\lambda'}+\frac{1}{\lambda'}), \\ & & (\frac{\lambda'}{\frac{\mu\cdot\gamma}{\mu+\gamma}+\lambda'})\cdot(\frac{1}{\frac{\mu\cdot\gamma}{\mu+\gamma}+\lambda'}+\frac{1}{\frac{\mu\cdot\gamma}{\mu+\gamma}})\,|\!\} \end{array}$$

Thus:

$$[<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}} \cap [<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}} = \emptyset$$

and hence:

$$\mathit{rate}(<a,\lambda>.<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},a,[<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}}) = \lambda$$

whereas:

$$\mathit{rate}(<a,\lambda>.<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0},a,[<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}]_{\approx_{\rm MB}}) = 0$$

Also the two divergent process terms ${\rm rec}X : <\tau,\mu>.<\tau,\gamma_1>.X$ and ${\rm rec}X : <\tau,\mu>.<\tau,\gamma_2>.X$, $\gamma_1 \neq \gamma_2$, are related by $\simeq_{\rm MB}$ but this no longer holds when placing them in the context $\_\,\|_{\emptyset}\,<a',\lambda'>.\underline{0}$, $a' \neq \tau$. ■

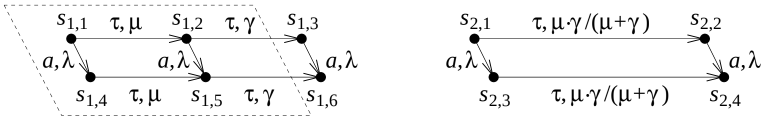
Taking inspiration from the weak isomorphism of [11], in this section we show how to retrieve full compositionality by enhancing the abstraction capability of $\simeq_{\rm MB}$ in the case of concurrent computations. The price to pay is that exactness will hold at steady state only for a certain class of processes.

4.1 Revising Weak Markovian Bisimilarity

As we have seen, $\approx_{\rm MB}$ and $\simeq_{\rm MB}$ abstract from sequences of exponentially timed τ-actions while preserving (at the computation level) their execution probability and average duration and (at the system level) transient properties expressed in terms of the mean time to certain events as well as steady-state performance measures. This kind of abstraction has been done in the simplest possible case: sequences of exponentially timed τ-actions labeling computations that traverse *fully unstable states*.

In order to achieve compositionality when dealing with concurrent processes, a revision of the notion of reducible computation is unavoidable. More precisely, we need to address the case of sequences of exponentially timed τ-actions labeling computations that traverse *unstable states satisfying certain conditions*. The reason is that, if we view a system description as the parallel composition of several sequential processes, any of those processes may have local computations traversing *fully unstable local states*, but in the overall system those local states may be *part of global states that are not fully unstable*.

For instance, this is the case with the process $<\tau,\mu>.<\tau,\gamma>.\underline{0}\,\|_{\emptyset}\,<a,\lambda>.\underline{0}$, whose underlying labeled multitransition system is depicted below on the left:



As can be noted, the fully unstable local states traversed by the only local computation of the sequential process $<\tau,\mu>.<\tau,\gamma>.\underline{0}$ may become part of unstable global states that are not fully unstable if $a \neq \tau$.

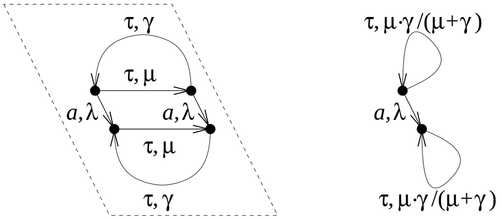
Our objective is to change the notion of reducible computation in such a way that the labeled multitransition system on the left can be regarded as being weakly Markovian bisimilar to the labeled multitransition system on the right. As can be noted, this implies that execution probabilities and average durations can only be preserved *at the level of local computations*, hence transient properties expressed in terms of the mean time to certain events can no longer be preserved at the system level.

In a concurrent setting, a sequence of exponentially timed τ-actions may be replicated due to interleaving, in the sense that it may label several computations that share no transition. The revision of the notion of reducible computation is thus based on the idea that, for each computation that traverses fully unstable *local* states and is labeled with exponentially timed τ-actions, we have to recognize – and take into account at once – *all the replicas* of that computation and pinpoint their initial and final states. In our example, there are two replicas with initial states $s_{1,1}$ and $s_{1,4}$ and final states $s_{1,3}$ and $s_{1,6}$.

In general, a one-to-one correspondence can be established between the states traversed by any two replicas by following the direction of the transitions. In our example, the pairs of corresponding states are the two initial states $(s_{1,1}, s_{1,4})$, the two intermediate states $(s_{1,2}, s_{1,5})$, and the two final states $(s_{1,3}, s_{1,6})$. We can say that *when moving vertically the current stage of the replicas is preserved*.

In addition to the exponentially timed τ-transition belonging to the replica, any two states traversed by the same replica can only possess transitions that are pairwise identically labeled. Those transitions are originated from (the local states of) sequential processes that are in parallel with (the local state of) the sequential process originating the considered reducible computation. The set of those transitions not belonging to the replica can thus be viewed as the *context* of the replica. In our example, the context of the top replica has a single transition labeled with $<a, \lambda>$, whereas the context of the bottom replica is empty. Thus, *when moving horizontally the context of each replica is preserved*, i.e., the context does not change along a replica. On the other hand, *different replicas may have different contexts*.

With regard to the identification of the boundary of the replicas of a reducible computation, there are two possibilities. One is that the final states have no exponentially timed τ-transition, as in our example. The other is that, at a certain point, each replica has an exponentially timed τ-transition back to one of the preceding states of the replica itself, as shown below with a variant of our example:



In this case, for each replica we view its return state as being its final state. In the figure above, for both replicas the final state coincides with the initial state.

The new notion of replicated reducible computation must be accompanied by an adjustment of the way measure *probtime* and multiset *pbtm* are calculated. Given a computation c of the form $P_1 \xrightarrow{\tau,\lambda_1}_{\rm M} P_2 \xrightarrow{\tau,\lambda_2}_{\rm M} \dots \xrightarrow{\tau,\lambda_n}_{\rm M} P_{n+1}$ that is reducible in the sense of Def. 3.2, the denominator of the i-th fraction occurring in each of the two factors of $probtime(c)$ can indifferently be $rate(P_i, \tau, \mathbb{P}_{\rm M})$ or $rate_{\rm t}(P_i)$: those two values coincide because $P_i \in \mathbb{P}_{\rm M,fu}$ for all $i = 1, \dots, n$. In contrast, if the reducible computation c is replicated, each of its replicas has a possibly different context and it is fundamental that $rate(P_i, \tau, \mathbb{P}_{\rm M})$ values are taken as denominators, so as to focus on τ-transitions. Since there can be τ-transitions also in the context, each destination of those exit rates needs to be a specific set $\mathscr{P}$ containing only the states traversed by the replicas rather than the generic set $\mathbb{P}_{\rm M}$. Taking into account only τ-transitions leading to states in $\mathscr{P}$ ensures *context independence* in this concurrent setting, which opens the way to the achievement of the same *probtime* value for all the replicas of a reducible computation.

We are by now ready to provide the definition of replicated reducible computation together with the revision of both *probtime* and *pbtm*. Since several reducible computations can depart from the same state (see the second and the third pair of process terms of Ex. 3.4), in general we will have to handle *replicated trees of reducible computations* rather than replicated individual reducible computations.

In the sequel, we consider $m \in \mathbb{N}_{>0}$ process terms $P_1, P_2, \dots, P_m \in \mathbb{P}_{\mathrm{M}}$ different from each other. We suppose that $P_k \xrightarrow{a_k, \lambda_k}_{\mathrm{M}} P_{k+1}$ for all $k = 1, \dots, m-1$, with P_k having a nonempty tree of computations that are locally reducible for all $k = 1, \dots, m$ (see $s_{1,1}$ and $s_{1,4}$ in our example). This tree is formalized as the set C_k^τ of all the finite-length computations starting from P_k such that each of them (i) is labeled with a sequence of exponentially timed τ-actions, (ii) traverses states that are all different with the possible exception of the final state and one of its preceding states, and (iii) shares no transitions with computations in $C_{k'}^\tau$ for all $k' \neq k$.

We further suppose that the union of $C_1^\tau, C_2^\tau, \dots, C_m^\tau$ can be partitioned into $n \in \mathbb{N}_{>0}$ *groups of replicas* each consisting of m computations from all the m sets, such that all the computations in the same group have the same length and are labeled with the same sequence of exponentially timed τ-actions. As a consequence, for all $k = 1, \dots, m$ we can write:

$$C_k^\tau \; = \; \{c_{k,i} \equiv P_{k,i,1} \xrightarrow{\tau, \lambda_{i,1}}_{\mathrm{M}} P_{k,i,2} \xrightarrow{\tau, \lambda_{i,2}}_{\mathrm{M}} \dots \xrightarrow{\tau, \lambda_{i,l_i}}_{\mathrm{M}} P_{k,i,l_i+1} \mid 1 \leq i \leq n\}$$

where $P_{k,i,1} \equiv P_k$ is the initial state and $l_i \in \mathbb{N}_{>0}$ is the length of the computation for all $i = 1, \dots, n$.

Definition 4.2 The family of computations $\mathscr{C}^\tau = \{C_1^\tau, C_2^\tau, \dots, C_m^\tau\}$ is said to be generally reducible, or g-reducible for short, iff <u>either</u> $m = 1$ and for all $i = 1, \dots, n$:

- $P_{1,i,j} \in \mathbb{P}_{\mathrm{M,fu}}$ for all $j = 1, \dots, l_i$;
- $P_{1,i,l_i+1} \in \mathbb{P}_{\mathrm{M,nfu}}$ or $P_{1,i,l_i+1} \equiv P_{1,i,j}$ for some $j = 1, \dots, l_i$;

<u>or</u> $m \geq 1$, with $P_{1,i,j} \in \mathbb{P}_{\mathrm{M,nfu}}$ for all $i = 1, \dots, n$ and $j = 1, \dots, l_i$ when $m = 1$, and for all $i = 1, \dots, n$:

- For all $k = 1, \dots, m$, $j = 1, \dots, l_i$, and $<a, \lambda> \in Act_{\mathrm{M}}$:

  1. [Deviation from the replica] If $P_{k,i,j} \xrightarrow{a, \lambda}_{\mathrm{M}} P'$ with $P' \not\equiv P_{k,i,j+1}$, then:

     a. [change of replica via context] either $P' \equiv P_{k',i,j}$ for some $k' = 1, \dots, m$;

     b. [change of computation] or $P' \equiv P_{k,i',j'}$ with $a = \tau$ and $\lambda = \lambda_{i',j'-1}$ for some $i' = 1, \dots, n$ other than i and some $j' = 2, \dots, l_{i'+1}$.

  2. [Context preservation along the replica] For all $k' = 1, \dots, m$, it holds that $P_{k,i,j} \xrightarrow{a, \lambda}_{\mathrm{M}} P_{k',i,j}$ iff $P_{k,i,j'} \xrightarrow{a, \lambda}_{\mathrm{M}} P_{k',i,j'}$ for all $j' = 1, \dots, l_i$.

  3. [Stage preservation across replicas] For all $i' = 1, \dots, n$ other than i and $j' = 2, \dots, l_{i'+1}$, it holds that $P_{k,i,j} \xrightarrow{a, \lambda}_{\mathrm{M}} P_{k,i',j'}$ iff $P_{k',i,j} \xrightarrow{a, \lambda}_{\mathrm{M}} P_{k',i',j'}$ for all $k' = 1, \dots, m$.

- [Termination] One of the following holds:

  $\overline{4}$. Whenever there exists $\lambda_{i,l_i+1} \in \mathbb{R}_{>0}$ such that $P_{k,i,l_i+1} \xrightarrow{\tau, \lambda_{i,l_i+1}}_{\mathrm{M}} P_{k,i,l_i+2}$ for all $k = 1, \dots, m$, then at least one of conditions 1, 2, and 3 above is not satisfied by P_{k',i,l_i+1} for some $k' = 1, \dots, m$.

  $\widetilde{4}$. There is no $\lambda_{i,l_i+1} \in \mathbb{R}_{>0}$ such that $P_{k,i,l_i+1} \xrightarrow{\tau, \lambda_{i,l_i+1}}_{\mathrm{M}} P_{k,i,l_i+2}$ for all $k = 1, \dots, m$.

  $\widehat{4}$. $P_{k,i,l_i+1} \equiv P_{k,i,j}$ for all $k = 1, \dots, m$ and some $j = 1, \dots, l_i$. ■

Some comments are now in order:

- In the case that $m = 1$ and all the traversed states are fully unstable (see the “either” option), Def. 4.2 coincides with Def. 3.2 except for the fact that the former considers a tree of computations whilst the latter considers a single computation.
- The case $m = 1$ with $P_{1,i,j} \in \mathbb{P}_{\mathrm{M,nfu}}$ for every $i = 1,\dots,n$ and $j = 1,\dots,l_i$ happens when all the sequential process terms in parallel with the one originating the tree of locally reducible computations repeatedly execute a single action (selfloop transition), thus causing no replica of the tree to be formed. Both this case and the case $m \geq 2$ are subject to conditions 1, 2, 3, and 4.
- Condition 1 establishes that each transition deviating (see $P' \not\equiv P_{k,i,j+1}$) from the replica of the considered computation of $\mathscr{C}^\tau$:
  - either is a vertical transition of the context that preserves the current stage of the replicas and hence causes the passage to the corresponding state of another replica ($k' \neq k$) or to the same state of the same replica ($k' = k$, meaning that one of the sequential process terms in parallel with the one originating the considered computation repeatedly executes a single action);
  - or is a transition belonging to some other computation in $\mathscr{C}^\tau$ starting from the same process term P_k as the considered computation.

  These two facts together imply the maximality of $\mathscr{C}^\tau$, because taking into account deviating transitions causes all replicas to be included. In addition, they prevent process terms like $<\tau,\mu>.(<\tau,\gamma>.\underline{0} + <a,\lambda>.\underline{0}) + <a,\lambda>.\underline{0}$ and $<\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} + <a,\lambda>.\underline{0}$ – which do not contain occurrences of parallel composition ($m = 1$) and have no fully unstable states when $a \neq \tau$ – from being deemed to be equivalent.
- Condition 2 is related to condition 1.a and ensures that the context of a replica is preserved along each state traversed by the replica.
- Condition 3 is related to condition 1.b and ensures that any transition belonging neither to the considered computation nor to its context (i.e., belonging to some other computation in $\mathscr{C}^\tau$) is present at the same stage of each replica of the considered computation.
- The three variants of condition 4 establish the boundary of the replicas of the considered computation in a way that guarantees the maximality of the length of the replicas themselves under (i) conditions 1, 2, and 3, (ii) the constraint that all of their transitions are labeled with exponentially timed τ-actions, (iii) and the constraint that all the traversed states are different with the possible exception of the final state and one of its preceding states.

Let $\mathit{initial}(\mathscr{C}^\tau) = \{P_k \mid 1 \leq k \leq m\}$ and $\mathit{final}(\mathscr{C}^\tau) = \{P_{k,i,l_i+1} \mid 1 \leq k \leq m, 1 \leq i \leq n\}$ be the sets of initial states and final states of the computations in $\mathscr{C}^\tau$. In order to avoid interferences between the computations in $C_1^\tau, C_2^\tau, \dots, C_m^\tau$ and the transitions belonging to the context of those computations, for any computation $c_{k,i}$ in $\mathscr{C}^\tau$ we consider the following context-free measure:

$$\mathit{probtime}_{\mathrm{cf}}(c_{k,i}) \;=\; \left(\prod_{j=1}^{l_i} \frac{\lambda_{i,j}}{\mathit{rate}(P_{k,i,j},\tau,\mathscr{P}_k)}\right) \cdot \left(\sum_{j=1}^{l_i} \frac{1}{\mathit{rate}(P_{k,i,j},\tau,\mathscr{P}_k)}\right)$$

where $\mathscr{P}_k = \{P_{k,i',j'} \mid 1 \leq i' \leq n, 2 \leq j' \leq l_{i'+1}\}$. In this way, all replicas of the same computation will have the same $\mathit{probtime}_{\mathrm{cf}}$ measure, as shown below.

Proposition 4.3 Whenever $\mathscr{C}^\tau$ is g-reducible, then for all $k,k' = 1,\dots,m$ and $i = 1,\dots,n$:

$$\mathit{probtime}_{\mathrm{cf}}(c_{k,i}) \;=\; \mathit{probtime}_{\mathrm{cf}}(c_{k',i}) \qquad \blacksquare$$

Moreover, we replace the generic multiset $\mathit{pbtm}(P,D)$ with the more specific multisets $\mathit{pbtm}_{\mathrm{cf}}(P_k, D \cap \mathit{final}(\mathscr{C}^\tau))$ for all $P_k \in \mathit{initial}(\mathscr{C}^\tau)$. The latter multisets are based on $\mathit{probtime}_{\mathrm{cf}}$ instead of $\mathit{probtime}$ as well as on $\mathit{reducomp}_{\mathrm{cf}}$ instead of $\mathit{reducomp}$, where $\mathit{reducomp}_{\mathrm{cf}}(P_k, D \cap \mathit{final}(\mathscr{C}^\tau), t)$ is the multiset of computations identical to those in C_k^τ that go from P_k to $D \cap \mathit{final}(\mathscr{C}^\tau)$ and have average duration t. We point out that computations of length zero are not considered as $t \in \mathbb{R}_{>0}$, so that whenever $P_k \in \mathit{initial}(\mathscr{C}^\tau) \cap D \cap \mathit{final}(\mathscr{C}^\tau)$, then the calculation of $\mathit{pbtm}_{\mathrm{cf}}(P_k, D \cap \mathit{final}(\mathscr{C}^\tau))$ does take into account computations identical to those in C_k^τ going from P_k to itself.

Proposition 4.4 Whenever $\mathscr{C}^\tau$ is g-reducible, then for all $k, k' = 1, \dots, m$:

$$\mathit{pbtm}_{\mathrm{cf}}(P_k, \mathit{final}(\mathscr{C}^\tau)) = \mathit{pbtm}_{\mathrm{cf}}(P_{k'}, \mathit{final}(\mathscr{C}^\tau)) \quad \blacksquare$$

We are finally ready to introduce the revised definition of weak Markovian bisimilarity.

Definition 4.5 An equivalence relation $\mathscr{B}$ over $\mathbb{P}_{\mathrm{M}}$ is a g-weak Markovian bisimulation iff, whenever $(P_1, P_2) \in \mathscr{B}$, then:

- For all visible action names $a \in \mathit{Name}_{\mathrm{v}}$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathscr{B}$:
$$\mathit{rate}(P_1, a, D) = \mathit{rate}(P_2, a, D)$$
- If P_1 is not an initial state of any g-reducible family of computations, then P_2 is not an initial state of any g-reducible family of computations either, and for all equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathscr{B}$:
$$\mathit{rate}(P_1, \tau, D) = \mathit{rate}(P_2, \tau, D)$$
- If P_1 is an initial state of some g-reducible family of computations, then P_2 is an initial state of some g-reducible family of computations too, and for all g-reducible families of computations $\mathscr{C}_1^\tau$ with $P_1 \in \mathit{initial}(\mathscr{C}_1^\tau)$ there exists a g-reducible family of computations $\mathscr{C}_2^\tau$ with $P_2 \in \mathit{initial}(\mathscr{C}_2^\tau)$ such that for all equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\mathscr{B}$:
$$\mathit{pbtm}_{\mathrm{cf}}(P_1, D \cap \mathit{final}(\mathscr{C}_1^\tau)) = \mathit{pbtm}_{\mathrm{cf}}(P_2, D \cap \mathit{final}(\mathscr{C}_2^\tau))$$

G-weak Markovian bisimilarity $\approx_{\mathrm{MB,g}}$ is the largest g-weak Markovian bisimulation. ■

Example 4.6 The process terms mentioned in each of the three cases of Ex. 3.4 are still related by $\approx_{\mathrm{MB,g}}$. Note that each of those process terms is the only initial state of a g-reducible family of computations composed by a single computation (first case) or a single tree of computations (second and third case) traversing only fully unstable states, thus $m = 1$ and the "either" option of Def. 4.2 applies. ■

Example 4.7 Let us reconsider the two process terms at the beginning of Ex. 4.1. Now we have:

$$<a,\lambda>.<\tau,\mu>.<\tau,\gamma>.\underline{0} \approx_{\mathrm{MB,g}} <a,\lambda>.<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}$$

and:

$$<a,\lambda>.<\tau,\mu>.<\tau,\gamma>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0} \approx_{\mathrm{MB,g}} <a,\lambda>.<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}$$

because it holds that:

$$<\tau,\mu>.<\tau,\gamma>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0} \approx_{\mathrm{MB,g}} <\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}$$

In fact, for $a' \neq \tau$ the two process terms are the initial states of two g-reducible families of computations $\mathscr{C}_1^\tau$ and $\mathscr{C}_2^\tau$, respectively, each composed of two replicas – the first one having context $\{<a',\lambda'>\}$ and final state $\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}$ and the second one having empty context and final state $\underline{0} \,\|_{\emptyset}\, \underline{0}$ – with:

$$\begin{aligned}\mathit{pbtm}_{\mathrm{cf}}(<\tau,\mu>.<\tau,\gamma>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}, D \cap \mathit{final}(\mathscr{C}_1^\tau)) &= \{\!|\, \tfrac{1}{\mu} + \tfrac{1}{\gamma} \,|\!\} \\ \mathit{pbtm}_{\mathrm{cf}}(<\tau,\tfrac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}, D \cap \mathit{final}(\mathscr{C}_2^\tau)) &= \{\!|\, \tfrac{\mu+\gamma}{\mu\cdot\gamma} \,|\!\}\end{aligned}$$

whenever D contains the final state $\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}$, as the way of calculating $\mathit{probtime}_{\mathrm{cf}}$ and $\mathit{pbtm}_{\mathrm{cf}}$ does not take the context into account.

For $a' = \tau$, in addition to $\mathscr{C}_1^{\tau}$ and $\mathscr{C}_2^{\tau}$, the two process terms are the initial states of two further g-reducible families of computations $\mathscr{C}_1'^{\tau}$ and $\mathscr{C}_2'^{\tau}$, respectively, each composed of two replicas of length 1 labeled with $<a', \lambda'>$. In this case:

$$\begin{array}{rcl} \mathit{pbtm}_{\mathrm{cf}}(<\tau,\mu>.<\tau,\gamma>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}, D \cap \mathit{final}(\mathscr{C}_1'^{\tau})) & = & \{\!|\, \frac{1}{\lambda'} \,|\!\} \\ \mathit{pbtm}_{\mathrm{cf}}(<\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}, D \cap \mathit{final}(\mathscr{C}_2'^{\tau})) & = & \{\!|\, \frac{1}{\lambda'} \,|\!\} \end{array}$$

whenever D contains the two $\approx_{\mathrm{MB,g}}$-equivalent final states $<\tau,\mu>.<\tau,\gamma>.\underline{0} \,\|_{\emptyset}\, \underline{0}$ and $<\tau, \frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} \,\|_{\emptyset}\, \underline{0}$. The two divergent process terms at the end of Ex. 4.1 are not related by $\approx_{\mathrm{MB,g}}$ because $\gamma_1 \neq \gamma_2$; hence, they no longer result in a disruption of compositionality when placed in the context $\_ \,\|_{\emptyset}\, <a',\lambda'>.\underline{0}$. ■

We conclude by showing that there exists a relationship between $\approx_{\mathrm{MB,g}}$ and $\approx_{\mathrm{MB}}$ only for process terms that have no cycles of exponentially timed τ-actions. The reason of this limitation is that $\approx_{\mathrm{MB,g}}$ imposes checks on those cycles that are not always performed by $\approx_{\mathrm{MB}}$, like, e.g., in the case of the two divergent process terms $\mathrm{rec}X : <\tau,\gamma_1>.X$ and $\mathrm{rec}X : <\tau,\gamma_2>.X$ where $\gamma_1 \neq \gamma_2$.

Proposition 4.8 Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$ be not divergent. Then:

$$P_1 \approx_{\mathrm{MB}} P_2 \Longrightarrow P_1 \approx_{\mathrm{MB,g}} P_2$$ ■

4.2 Congruence Property

The investigation of the compositionality of $\approx_{\mathrm{MB,g}}$ with respect to MPC operators leads to results analogous to those for $\approx_{\mathrm{MB}}$ [3], plus the achievement of congruence with respect to parallel composition.

Proposition 4.9 Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Whenever $P_1 \approx_{\mathrm{MB,g}} P_2$, then:

1. $<a,\lambda>.P_1 \approx_{\mathrm{MB,g}} <a,\lambda>.P_2$ for all $<a,\lambda> \in \mathit{Act}_{\mathrm{M}}$.
2. $P_1/H \approx_{\mathrm{MB,g}} P_2/H$ for all $H \subseteq \mathit{Name}_{\mathrm{v}}$.
3. $P_1 \,\|_S\, P \approx_{\mathrm{MB,g}} P_2 \,\|_S\, P$ and $P \,\|_S\, P_1 \approx_{\mathrm{MB,g}} P \,\|_S\, P_2$ for all $S \subseteq \mathit{Name}_{\mathrm{v}}$ and $P \in \mathbb{P}_{\mathrm{M}}$. ■

The relation $\approx_{\mathrm{MB,g}}$ is not a congruence with respect to the alternative composition operator due to fully unstable process terms: for instance, it holds that $<\tau,\mu>.<\tau,\gamma>.\underline{0} \approx_{\mathrm{MB,g}} <\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0}$ whereas $<\tau,\mu>.<\tau,\gamma>.\underline{0} + <a,\lambda>.\underline{0} \not\approx_{\mathrm{MB,g}} <\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} + <a,\lambda>.\underline{0}$. In fact, if it were $a \neq \tau$, then we would have:

$$\begin{array}{rcl} \mathit{rate}(<\tau,\mu>.<\tau,\gamma>.\underline{0} + <a,\lambda>.\underline{0}, \tau, [\underline{0}]_{\approx_{\mathrm{MB,g}}}) & = & 0 \\ \mathit{rate}(<\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} + <a,\lambda>.\underline{0}, \tau, [\underline{0}]_{\approx_{\mathrm{MB,g}}}) & = & \frac{\mu\cdot\gamma}{\mu+\gamma} \end{array}$$

otherwise for $a = \tau$ the two process terms would be the initial states of two g-reducible families of computations, respectively, each composed of a single tree of computations with final state $\underline{0}$ and we would have:

$$\begin{array}{rcl} \mathit{pbtm}_{\mathrm{cf}}(<\tau,\mu>.<\tau,\gamma>.\underline{0} + <a,\lambda>.\underline{0}, \{\underline{0}\}) & = & \{\!|\, \frac{\mu}{\mu+\lambda} \cdot \left(\frac{1}{\mu+\lambda} + \frac{1}{\gamma}\right), \frac{\lambda}{\mu+\lambda} \cdot \frac{1}{\mu+\lambda} \,|\!\} \\ \mathit{pbtm}_{\mathrm{cf}}(<\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.\underline{0} + <a,\lambda>.\underline{0}, \{\underline{0}\}) & = & \{\!|\, \frac{1}{\frac{\mu\cdot\gamma}{\mu+\gamma}+\lambda} \,|\!\} \end{array}$$

The congruence violation with respect to the alternative composition operator can be prevented by adopting a construction analogous to the one used in [13] for weak bisimilarity over nondeterministic process terms and adapted in [3] to $\approx_{\mathrm{MB}}$. Therefore, we have to apply the exit rate equality check for τ-actions also to process terms that are initial states of g-reducible families of computations, with the equivalence classes to consider being the ones with respect to $\approx_{\mathrm{MB,g}}$.

Definition 4.10 Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. We say that P_1 is g-weakly Markovian bisimulation congruent to P_2, written $P_1 \simeq_{\mathrm{MB,g}} P_2$, iff for all action names $a \in \mathit{Name}$ and equivalence classes $D \in \mathbb{P}_{\mathrm{M}}/\approx_{\mathrm{MB,g}}$:

$$\mathit{rate}(P_1, a, D) = \mathit{rate}(P_2, a, D)$$ ■

Proposition 4.11 $\sim_{\rm MB} \subset \simeq_{\rm MB,g} \subset \approx_{\rm MB,g}$, with $\simeq_{\rm MB,g} = \approx_{\rm MB,g}$ over the set of process terms of $\mathbb{P}_{\rm M}$ that are not initial states of any g-reducible family of computations. ■

Proposition 4.12 Let $P_1, P_2 \in \mathbb{P}_{\rm M}$ and $<a,\lambda> \in Act_{\rm M}$. Then:

$$<a,\lambda>.P_1 \simeq_{\rm MB,g} <a,\lambda>.P_2 \iff P_1 \approx_{\rm MB,g} P_2$$ ■

The relation $\simeq_{\rm MB,g}$ turns out to be the coarsest congruence – with respect to all the operators of MPC as well as recursion – contained in $\approx_{\rm MB,g}$, as shown below.

Theorem 4.13 Let $P_1, P_2 \in \mathbb{P}_{\rm M}$. Whenever $P_1 \simeq_{\rm MB,g} P_2$, then:

1. $<a,\lambda>.P_1 \simeq_{\rm MB,g} <a,\lambda>.P_2$ for all $<a,\lambda> \in Act_{\rm M}$.
2. $P_1 + P \simeq_{\rm MB,g} P_2 + P$ and $P + P_1 \simeq_{\rm MB,g} P + P_2$ for all $P \in \mathbb{P}_{\rm M}$.
3. $P_1 / H \simeq_{\rm MB,g} P_2 / H$ for all $H \subseteq Name_{\rm v}$.
4. $P_1 \,\|_S\, P \simeq_{\rm MB,g} P_2 \,\|_S\, P$ and $P \,\|_S\, P_1 \simeq_{\rm MB,g} P \,\|_S\, P_2$ for all $S \subseteq Name_{\rm v}$ and $P \in \mathbb{P}_{\rm M}$. ■

Theorem 4.14 Let $P_1, P_2 \in \mathbb{P}_{\rm M}$. Then $P_1 \simeq_{\rm MB,g} P_2$ iff $P_1 + P \approx_{\rm MB,g} P_2 + P$ for all $P \in \mathbb{P}_{\rm M}$. ■

With regard to recursion, we need to extend $\simeq_{\rm MB,g}$ to open process terms in the usual way. Similar to other congruence proofs for bisimulation equivalence with respect to recursion, here we rely on a notion of g-weak Markovian bisimulation up to $\approx_{\rm MB,g}$ inspired by the notion of Markovian bisimulation up to $\sim_{\rm MB}$ of [5]. This notion differs from its nondeterministic counterpart used in [13] due to the necessity of working with equivalence classes in this Markovian setting.

Definition 4.15 Let $P_1, P_2 \in \mathscr{PL}_{\rm M}$ be process terms containing free occurrences of $k \in \mathbb{N}$ process variables $X_1, \dots, X_k \in Var$ at most. We define $P_1 \simeq_{\rm MB,g} P_2$ iff $P_1\{Q_i \hookrightarrow X_i \mid 1 \le i \le k\} \simeq_{\rm MB,g} P_2\{Q_i \hookrightarrow X_i \mid 1 \le i \le k\}$ for all $Q_1, \dots, Q_k \in \mathscr{PL}_{\rm M}$ containing no free occurrences of process variables. ■

Definition 4.16 Let $^+$ denote the operation of transitive closure for relations. A binary relation $\mathscr{B}$ over $\mathbb{P}_{\rm M}$ is a g-weak Markovian bisimulation up to $\approx_{\rm MB,g}$ iff, whenever $(P_1, P_2) \in \mathscr{B}$, then:

- For all visible action names $a \in Name_{\rm v}$ and equivalence classes $D \in \mathbb{P}_{\rm M} / (\mathscr{B} \cup \mathscr{B}^{-1} \cup \approx_{\rm MB,g})^+$:
$$rate(P_1, a, D) \;=\; rate(P_2, a, D)$$
- If P_1 is not an initial state of any g-reducible family of computations, then P_2 is not an initial state of any g-reducible family of computations either, and for all equivalence classes $D \in \mathbb{P}_{\rm M} / (\mathscr{B} \cup \mathscr{B}^{-1} \cup \approx_{\rm MB,g})^+$:
$$rate(P_1, \tau, D) \;=\; rate(P_2, \tau, D)$$
- If P_1 is an initial state of some g-reducible family of computations, then P_2 is an initial state of some g-reducible family of computations too, and for all g-reducible families of computations $\mathscr{C}_1^\tau$ with $P_1 \in initial(\mathscr{C}_1^\tau)$ there exists a g-reducible family of computations $\mathscr{C}_2^\tau$ with $P_2 \in initial(\mathscr{C}_2^\tau)$ such that for all equivalence classes $D \in \mathbb{P}_{\rm M} / (\mathscr{B} \cup \mathscr{B}^{-1} \cup \approx_{\rm MB,g})^+$:
$$pbtm_{\rm cf}(P_1, D \cap final(\mathscr{C}_1^\tau)) \;=\; pbtm_{\rm cf}(P_2, D \cap final(\mathscr{C}_2^\tau))$$ ■

Proposition 4.17 Let $\mathscr{B}$ be a relation over $\mathbb{P}_{\rm M}$. If $\mathscr{B}$ is a g-weak Markovian bisimulation up to $\approx_{\rm MB,g}$, then $(P_1, P_2) \in \mathscr{B}$ implies $P_1 \approx_{\rm MB,g} P_2$ for all $P_1, P_2 \in \mathbb{P}_{\rm M}$. Moreover $(\mathscr{B} \cup \mathscr{B}^{-1} \cup \approx_{\rm MB,g})^+ = \approx_{\rm MB,g}$. ■

Theorem 4.18 Let $P_1, P_2 \in \mathscr{PL}_{\rm M}$ be process terms containing free occurrences of $k \in \mathbb{N}$ process variables $X_1, \dots, X_k \in Var$ at most. Whenever $P_1 \simeq_{\rm MB,g} P_2$, then:

$$\mathrm{rec}\, X_1 : \dots : \mathrm{rec}\, X_k : P_1 \;\simeq_{\rm MB,g}\; \mathrm{rec}\, X_1 : \dots : \mathrm{rec}\, X_k : P_2$$ ■

4.3 Exactness at Steady State

We conclude by examining the exactness of the CTMC-level aggregation induced by $\approx_{\mathrm{MB,g}}$ and $\simeq_{\mathrm{MB,g}}$. In general, a CTMC aggregation is said to be exact at steady state (resp. transient state) iff the steady-state (resp. transient) probability of being in a macrostate of an aggregated CTMC is the sum of the steady-state (resp. transient) probabilities of being in each of the constituent microstates of the original CTMC from which the aggregated one has been obtained. This property implies the preservation of steady-state (resp. transient) reward-based performance measures across CTMC models.

The aggregation to examine – which we call GW-lumpability – shares with the one induced by $\approx_{\mathrm{MB}}$ and $\simeq_{\mathrm{MB}}$ – called W-lumpability in [3] – the characteristic of viewing certain sequences of exponentially timed τ-actions to be equivalent to individual exponentially timed τ-actions having the same average duration and the same execution probability as the corresponding sequences when the latter are considered locally to the processes originating them.[1] On the other hand, due to the idea of context embodied in the notion of g-reducible family of computations and the consequent capability of distinguishing between action disabling and action interruption, a notable difference between GW-lumpability and W-lumpability is that the former may aggregate states also in the case of concurrent processes, while the latter cannot.

Reducing a computation formed by at least two exponentially timed τ-transitions to a single exponentially timed τ-transition with the same average duration amounts to approximating a hypoexponentially (or Erlang) distributed random variable with an exponentially distributed random variable having the same expected value. This implies that, in general, GW-lumpability cannot preserve transient performance measures, as was the case with W-lumpability [3]. However, while W-lumpability at least preserves transient properties expressed in terms of the mean time to certain events, this is no longer the case with GW-lumpability as we have seen at the beginning of Sect. 4.1.

What turns out for GW-lumpability is that, similar to W-lumpability, it preserves steady-state performance measures, provided that the states traversed by any replica of a reducible computation have the same rewards and the transitions – belonging to the replica or to the context – departing from any two traversed states have pairwise identical rewards. However, unlike W-lumpability, we have to confine ourselves to processes in which synchronizations (if any) do not take place right before the beginning of computations that are reducible according to the “or” option of Def. 4.2. This constraint comes from the insensitivity conditions for generalized semi-Markov processes mentioned in [12, 8, 11].

Theorem 4.19 GW-lumpability is exact at steady state over every process term $P \in \mathbb{P}_{\mathrm{M}}$ such that, for all g-reducible families of computations $\mathscr{C}^{\tau}$ in $[\![P]\!]_{\mathrm{M}}$ with size $m \geq 2$, or size $m = 1$ and all the traversed states being not fully unstable, no state in *initial*$(\mathscr{C}^{\tau})$ is the target state of a transition in $[\![P]\!]_{\mathrm{M}}$ arising from the synchronization of two or more actions. ■

Example 4.20 In order to illustrate the need for the constraint on synchronizations in Thm. 4.19, consider the following two process terms:

$$\begin{array}{lll} P_1 & \equiv & \mathrm{rec}\,X : <\tau,\mu>.<\tau,\gamma>.<b,\delta>.X \parallel_{\{b\}} \mathrm{rec}\,Y : <a,\lambda>.<b,\delta>.Y \\ P_2 & \equiv & \mathrm{rec}\,X : <\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.<b,\delta>.X \parallel_{\{b\}} \mathrm{rec}\,Y : <a,\lambda>.<b,\delta>.Y \end{array}$$

Observe that $P_1 \approx_{\mathrm{MB,g}} P_2$ and that $[\![P_1]\!]_{\mathrm{M}}$ and $[\![P_2]\!]_{\mathrm{M}}$ are given by the two labeled multitransition systems depicted at the beginning of Sect. 4.1, respectively, with an additional transition labeled with $<b,\delta>$ from the final state to the initial one. In the case that $\mu = \gamma = \lambda = \delta = 1$ and $\delta \otimes \delta = \delta$, it turns out that the steady-state probability distribution for $[\![P_1]\!]_{\mathrm{M}}$ is as follows:

[1] To be precise, since the Markov property of the original CTMC is not preserved but the aggregated stochastic process is still assumed to be a CTMC, it would be more appropriate to call those aggregations pseudo-aggregations [14].

$$\begin{array}{llllll} \pi[s_{1,1}] = \frac{2}{13} & \pi[s_{1,2}] = \frac{1}{13} & \pi[s_{1,3}] = \frac{1}{13} \\ \pi[s_{1,4}] = \frac{2}{13} & \pi[s_{1,5}] = \frac{3}{13} & \pi[s_{1,6}] = \frac{4}{13} \end{array}$$

whereas the steady-state probability distribution for $[\![P_2]\!]_{\rm M}$ is as follows:

$$\begin{array}{ll} \pi[s_{2,1}] = \frac{2}{10} & \pi[s_{2,2}] = \frac{1}{10} \\ \pi[s_{2,3}] = \frac{4}{10} & \pi[s_{2,4}] = \frac{3}{10} \end{array}$$

Thus, the CTMC underlying $[\![P_2]\!]_{\rm M}$ is not an exact aggregation of the CTMC underlying $[\![P_1]\!]_{\rm M}$ because:

$$\begin{array}{ll} \pi[s_{1,1}] + \pi[s_{1,2}] \neq \pi[s_{2,1}] & \pi[s_{1,3}] \neq \pi[s_{2,2}] \\ \pi[s_{1,4}] + \pi[s_{1,5}] \neq \pi[s_{2,3}] & \pi[s_{1,6}] \neq \pi[s_{2,4}] \end{array}$$

As can be noted, the transition in $[\![P_1]\!]_{\rm M}$ labeled with $<b,\delta>$ arises from the synchronization of two b-actions and its target state is the initial state of a computation belonging to a g-reducible family with size $m = 2$; hence, Thm. 4.19 does not apply.

In contrast, if we consider a synchronization-free variant of the two process terms above like for instance:

$$\begin{array}{lll} P_3 & \equiv & \mathrm{rec}\,X : <\tau,\mu>.<\tau,\gamma>.<b_1,\delta_1>.X \parallel_{\emptyset} \mathrm{rec}\,Y : <a,\lambda>.<b_2,\delta_2>.Y \\ P_4 & \equiv & \mathrm{rec}\,X : <\tau,\frac{\mu\cdot\gamma}{\mu+\gamma}>.<b_1,\delta_1>.X \parallel_{\emptyset} \mathrm{rec}\,Y : <a,\lambda>.<b_2,\delta_2>.Y \end{array}$$

we have that for $\mu = \gamma = \lambda = \delta_1 = \delta_2 = 1$ the steady-state probability distribution for $[\![P_3]\!]_{\rm M}$ is:

$$\begin{array}{lll} \pi[s_{3,1}] = \frac{1}{6} & \pi[s_{3,2}] = \frac{1}{6} & \pi[s_{3,3}] = \frac{1}{6} \\ \pi[s_{3,4}] = \frac{1}{6} & \pi[s_{3,5}] = \frac{1}{6} & \pi[s_{3,6}] = \frac{1}{6} \end{array}$$

and the steady-state probability distribution for $[\![P_4]\!]_{\rm M}$ is:

$$\begin{array}{ll} \pi[s_{4,1}] = \frac{2}{6} & \pi[s_{4,2}] = \frac{1}{6} \\ \pi[s_{4,3}] = \frac{2}{6} & \pi[s_{4,4}] = \frac{1}{6} \end{array}$$

hence the CTMC underlying $[\![P_4]\!]_{\rm M}$ is an exact aggregation of the CTMC underlying $[\![P_3]\!]_{\rm M}$ because:

$$\begin{array}{ll} \pi[s_{3,1}] + \pi[s_{3,2}] = \pi[s_{4,1}] & \pi[s_{3,3}] = \pi[s_{4,2}] \\ \pi[s_{3,4}] + \pi[s_{3,5}] = \pi[s_{4,3}] & \pi[s_{3,6}] = \pi[s_{4,4}] \end{array} \quad \blacksquare$$

5 Conclusion

In this paper, we have introduced $\approx_{\rm MB,g}$ and $\simeq_{\rm MB,g}$ as variants of the weak Markovian bisimulation equivalences $\approx_{\rm MB}$ and $\simeq_{\rm MB}$ proposed in [3], which suffer from a limited usefulness for state space reduction purposes as they are not congruences with respect to the parallel composition operator. The motivation behind $\approx_{\rm MB,g}$ and $\simeq_{\rm MB,g}$ is thus that of retrieving full compositionality. Taking inspiration from the idea of preserving the context of [11], this has been achieved by enhancing the abstraction capability – with respect to $\approx_{\rm MB}$ and $\simeq_{\rm MB}$ – when dealing with concurrent computations. The price to pay for the resulting compositional abstraction capability is that the exactness at steady state of the induced CTMC-level aggregation does not hold for all the considered processes – as it was for $\approx_{\rm MB}$ and $\simeq_{\rm MB}$ – but only for sequential processes with abstraction and concurrent processes whose synchronizations do not take place right before the beginning of computations to be reduced. Additionally, not even transient properties expressed in terms of the mean time to certain events are preserved in general.

With regard to [11], where weak isomorphism has been studied, our equivalences $\approx_{\rm MB,g}$ and $\simeq_{\rm MB,g}$ have been developed in the more liberal bisimulation framework. A more important novelty with respect to weak isomorphism is that we have considered not only individual sequences of exponentially timed τ-actions. In fact, we have addressed trees of exponentially timed τ-actions and we have established the conditions under which such trees can be reduced – also in the presence of parallel composition – by locally preserving both the average duration and the execution probability of their branches.

Another approach to abstracting from τ-actions in an exponentially timed setting comes from [4], where a variant of Markovian bisimilarity was defined that checks for exit rate equality with respect to all equivalence classes apart from the one including the processes under examination. Congruence and axiomatization results were provided for the proposed equivalence, and a logical characterization based on CSL was illustrated in [2]. However, unlike $\approx_{\mathrm{MB,g}}$ and $\simeq_{\mathrm{MB,g}}$, nothing was said about exactness.

As far as future work is concerned, we would like to investigate equational and logical characterizations of $\simeq_{\mathrm{MB,g}}$ as well as conduct case studies for assessing its usefulness in practice (especially with respect to the constraint on synchronizations that guarantees steady-state exactness). With regard to verification issues, since $\simeq_{\mathrm{MB}} \subset \simeq_{\mathrm{MB,g}}$ for non-divergent process terms, we have that the equivalence checking algorithm developed for $\simeq_{\mathrm{MB}}$ in [3] can be exploited for compositional state space reduction with respect to $\simeq_{\mathrm{MB,g}}$, by applying it to each of the sequential processes composed in parallel.

Acknowledgment: This work has been funded by MIUR-PRIN project *PaCo – Performability-Aware Computing: Logics, Models, and Languages*.

References

[1] A. Aldini, M. Bernardo, and F. Corradini, *"A Process Algebraic Approach to Software Architecture Design"*, Springer, 2010, doi:10.1007/978-1-84800-223-4.

[2] C. Baier, J.-P. Katoen, H. Hermanns, and V. Wolf, *"Comparative Branching-Time Semantics for Markov Chains"*, in Information and Computation 200:149–214, 2005, doi:10.1016/j.ic.2005.03.001.

[3] M. Bernardo, *"Weak Markovian Bisimulation Congruences and Exact CTMC-Level Aggregations for Sequential Processes"*, in Proc. of TGC 2011, LNCS 7173:89–103.

[4] M. Bravetti, *"Revisiting Interactive Markov Chains"*, in Proc. of MTCS 2002, ENTCS 68(5):1–20, doi:10.1016/S1571-0661(04)80520-6.

[5] M. Bravetti, M. Bernardo, and R. Gorrieri, *"A Note on the Congruence Proof for Recursion in Markovian Bisimulation Equivalence"*, in Proc. of PAPM 1998, pp. 153–164.

[6] P. Buchholz, *"Exact and Ordinary Lumpability in Finite Markov Chains"*, in Journal of Applied Probability 31:59–75, 1994, doi:10.2307/3215235.

[7] S. Derisavi, H. Hermanns, and W.H. Sanders, *"Optimal State-Space Lumping in Markov Chains"*, in Information Processing Letters 87:309–315, 2003, doi:10.1016/S0020-0190(03)00343-0.

[8] W. Henderson and D. Lucic, *"Aggregation and Disaggregation Through Insensitivity in Stochastic Petri Nets"*, in Performance Evaluation 17:91–114, 1993, doi:10.1016/0166-5316(93)90002-C.

[9] H. Hermanns, *"Interactive Markov Chains"*, LNCS 2428, 2002, doi:10.1007/3-540-45804-2.

[10] H. Hermanns and M. Rettelbach, *"Syntax, Semantics, Equivalences, and Axioms for MTIPP"*, in Proc. of PAPM 1994, pp. 71–87.

[11] J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996, doi:10.1017/CBO9780511569951.

[12] K. Matthes, *"Zur Theorie der Bedienungsprozesse"*, in Proc. of the *3rd Prague Conf. on Information Theory, Statistical Decision Functions and Random Processes*, pp. 513–528, 1962.

[13] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989.

[14] G. Rubino and B. Sericola, *"Sojourn Times in Finite Markov Processes"*, in Journal of Applied Probability 27:744–756, 1989, doi:10.2307/3214379.

[15] W.J. Stewart, *"Introduction to the Numerical Solution of Markov Chains"*, Princeton University Press, 1994.