ELSEVIER

Theoretical Computer Science 202 (1998) 1–54

Theoretical Computer Science

Tutorial

A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time

Marco Bernardo *, Roberto Gorrieri

Università di Bologna, Dipartimento di Scienze dell'Informazione, Mura Anteo Zamboni 7, 40127 Bologna, Italy

Received December 1996; revised January 1997
Communicated by M. Nivat

Abstract

In this tutorial we give an overview of the process algebra EMPA, a calculus devised in order to model and analyze features of real-world concurrent systems such as nondeterminism, priorities, probabilities and time, with a particular emphasis on performance evaluation. The purpose of this tutorial is to explain the design choices behind the development of EMPA and how the four features above interact, and to show that a reasonable trade off between the expressive power of the calculus and the complexity of its underlying theory has been achieved. © 1998 — Elsevier Science B.V. All rights reserved

Keywords: Process algebra; Priority; Probability; Markov chain; Bisimulation equivalence

Contents

* Corresponding author. E-mail: bernardo@cs.unibo.it.

PII S0304-3975(97)00127-8

1. Introduction

Several process algebras have been proposed in the literature in order to model and analyze concurrent systems. *Classical process algebras* such as CCS [40], CSP [31], ACP [6] and LOTOS [10] were concerned only with functional aspects of concurrent systems. This means that actions composing algebraic terms were only given a name, and nothing was said about e.g. their duration. As a consequence, only functional properties (e.g. absence of deadlock) of concurrent systems could be investigated.

Subsequently, the expressiveness of classical process algebras was enriched by allowing for the modeling of real-world features such as priorities, probabilities and durations, thereby resulting in *prioritized process algebras* (see e.g. [4, 13, 14, 57, 17, 15]), *probabilistic process algebras* (see e.g. [49, 32, 58, 39, 36, 5, 56]), *deterministically timed process algebras* (see e.g. [51, 3, 43, 50, 41, 21, 60, 24, 38, 20]), and *stochastically timed process algebras* (see e.g. [44, 25, 29, 12, 2, 22, 27, 52, 11, 47, 26, 33, 48]). The enhanced expressive power achieved by these classes of process algebras has allowed to model and analyze a greater number of characteristics with respect to classical process algebras such as interrupt mechanisms, resources where tasks having different priorities may arrive, loss probability of communication channels, probability to reach a deadlocked state, satisfaction of real-time constraints, throughput and utilization of resources.

Now, if we examine the process algebras mentioned above, we realize that *nondeterminism*, *priorities*, *probabilities* and *time* are usually considered in isolation. In other words, to the best of our knowledge there is no process algebra where *all* the four features are taken into account. The point is that it would be nice to develop one single calculus of concurrent processes accounting for nondeterminism, priorities, probabilities and time in order to exploit all the advantages afforded by each class of process algebras, provided that the theory underlying such a calculus is not too complex. This means that a reasonable trade off between the modeling power and the availability of analysis tools should be attained.

In order to achieve this objective, we have developed a process algebra called *extended Markovian process algebra* (EMPA). The development of EMPA has been strongly influenced by the stochastically timed process algebras MTIPP [25] and PEPA

[29], and by the formalism of generalized stochastic Petri nets (GSPNs) [1]. This is witnessed by the fact that in EMPA there are three different kinds of actions: exponentially timed actions (taken from MTIPP and PEPA), prioritized weighted immediate actions (analogous to prioritized weighted immediate transitions of GSPNs), and passive actions (similar to passive actions of PEPA). Exponentially timed actions describe activities that are relevant from the performance point of view. Prioritized weighted immediate actions model logical events as well as activities that are either irrelevant from the performance point of view or unboundedly faster than the others, and are useful to express both prioritized choices and probabilistic choices. Finally, passive actions model activities waiting for the synchronization with exponentially timed or immediate activities, and are useful to express nondeterministic choices. The purpose of this tutorial is to present EMPA by showing how nondeterminism, priorities, probabilities and time have been combined together by means of the different kinds of actions just mentioned, so that a considerable expressive power has been achieved without burdening the underlying theory exceedingly.

The tutorial is organized as follows. In Section 2 we present the syntax of EMPA terms and we informally explain the semantics of EMPA operators. In Section 3 we define the integrated interleaving semantics of EMPA terms. In Section 4 we show that the coexistence of the different kinds of actions can be thought of as the coexistence of a nondeterministic kernel, a prioritized kernel, a probabilistic kernel and an exponentially timed kernel. The division of EMPA into four kernels should provide a better insight in the structure of EMPA, and allows us to make some comparisons with process algebras appeared in the literature. In Section 5 we define a notion of equivalence over EMPA terms which is built by considering the various kernels singled out in the previous section (proofs of results are included in the appendix). Finally, in Section 6 we report some concluding remarks on further enhancements of the expressiveness of EMPA and the related consequences on the complexity of the underlying theory.

This tutorial is based mainly on [9, 7] and constitutes a revised version of both of them. With respect to [9], we emphasize the coexistence of several kernels and the resulting expressive power like in [7]; unlike [7], passive actions are part of each of the four kernels in order to stress their gluing role. Two new technical results, the axiomatization of the notion of equivalence and an algorithm to check two terms for equivalence, are presented at the end of Section 5.

2. Syntax and informal semantics for EMPA

2.1. Actions: types and rates

The building blocks of EMPA are *actions*. Each action is a pair $\langle a, \tilde{\lambda} \rangle$ consisting of the *type* of the action and the *rate* of the action. The type denotes the kind of the action (e.g. transmission of a message), while the rate indicates the speed at which the action occurs from the point of view of an external observer: rates are used as a concise way to denote the random variables specifying the duration of the actions (see below). Depending on the type, like in classical process algebras, actions are divided into *external* and *internal*: as usual, we denote by τ the only internal action type we use. Moreover, we have the following classification according to the rates:

- *Active actions* are actions whose rate is specified. An active action can be either exponentially timed or immediate:
  - *Exponentially timed actions* are actions whose rate is a positive real number. Such a number is interpreted as the parameter of the exponentially distributed random variable specifying the duration of the action. We recall that an exponentially distributed random variable X has probability distribution function $F_X(t) = \Pr[X \leqslant t] = 1 - e^{-\lambda \cdot t}$ for any $t \in \mathbb{R}_+$, expected value $1/\lambda$ and variance $1/\lambda^2$, thus it is uniquely identified by its parameter $\lambda \in \mathbb{R}_+$.
  - *Immediate actions* are actions whose rate, denoted by $\infty_{l,w}$, is infinite. Such actions have duration zero, and each of them is given a *priority level* $l \in \mathbb{N}_+$ and a *weight* $w \in \mathbb{R}_+$.
- *Passive actions* are actions whose rate, denoted by $*$, is undefined. The duration of a passive action is fixed only by synchronizing it with an active action of the same type.

The classification of actions based on their rates implies that: (i) exponentially timed actions model activities that are relevant from the performance point of view, (ii) immediate actions model logical events as well as activities that are either irrelevant from the performance point of view or unboundedly faster than the others, (iii) passive actions model activities waiting for the synchronization with timed activities. The motivations behind the restriction of timed action durations to be exponentially distributed or zero are related to the possibility of defining the semantics for EMPA in the classical interleaving style, as we shall see in Section 2.3, and of obtaining performance models in the form of Markov chains, as we shall see in Section 4. The apparently reduced expressive power stemming from this choice will be examined in Section 4.5.

We denote the set of actions by $Act = AType \times ARate$ where $AType$ is the set of types and $ARate = \mathbb{R}_+ \cup Inf \cup \{*\}$, with $Inf = \{\infty_{l,w} \mid l \in \mathbb{N}_+ \wedge w \in \mathbb{R}_+\}$, is the set of rates. We use $a, b, c, \ldots$ as metavariables for $AType$, $\tilde{\lambda}, \tilde{\mu}, \tilde{\gamma}, \ldots$ for $ARate$, and $\lambda, \mu, \gamma, \ldots$ for $\mathbb{R}_+$. Finally, we denote by $APLev = \{-1\} \cup \mathbb{N}$ the set of *action priority levels*, and we assume that $* < \lambda < \infty_{l,w}$ for all $\lambda \in \mathbb{R}_+$ and $\infty_{l,w} \in Inf$.

2.2. Syntax of terms and informal semantics of operators

Let $Const$ be a set of *constants*, ranged over by $A, B, \ldots$, and let $ARFun = \{\varphi : AType \to AType \mid \varphi(\tau) = \tau \wedge \varphi(AType - \{\tau\}) \subseteq AType - \{\tau\}\}$ be a set of *action relabeling functions.*

Definition 2.1. The set $\mathscr{L}$ of *process terms* of EMPA is generated by the following syntax

$$E ::= \underline{0} \mid \langle a, \tilde{\lambda} \rangle .E \mid E/L \mid E[\varphi] \mid E + E \mid E \parallel_S E \mid A$$

where $L, S \subseteq AType - \{\tau\}$. The set $\mathscr{L}$ will be ranged over by $E, F, G, \ldots$.

The *null term* "$\underline{0}$" represents a termination or deadlocked state.

The *prefix operator* "$\langle a, \tilde{\lambda} \rangle .\_$" denotes the sequential composition of an action and a term: term $\langle a, \tilde{\lambda} \rangle .E$ can execute action $\langle a, \tilde{\lambda} \rangle$ and then behaves as term E.

The *functional abstraction operator* "$\_/L$" abstracts from the type of the actions: term E/L behaves as term E except that the type of each executed action is turned into τ whenever it is in L. The meaning of this operator is the same as that of the hiding operator of CSP [31], thereby providing a means whereby encapsulating or ignoring functional information.

The *functional relabeling operator* "$\_[\varphi]$" changes the type of the actions: term $E[\varphi]$ behaves as term E except that the type of each executed action is modified according to φ. The meaning of this operator is the same as that of the relabeling operator of CCS [40], thus providing a means whereby obtaining more compact algebraic descriptions.

The *alternative composition operator* "$\_ + \_$" expresses a choice between two terms: term $E_1 + E_2$ behaves as either term E_1 or term E_2 depending on whether an action of E_1 or an action of E_2 is executed first. As we shall see in Section 2.3, the way in which the choice is resolved depends on the kind of the actions involved in the choice itself.

The *parallel composition operator* "$\_ \|_S \_$" expresses the concurrent execution of two terms according to two synchronization disciplines. The synchronization discipline on action types is the same as that of CSP [31], hence two actions can synchronize only if they have the same type, and this coincides with the resulting type. The synchronization discipline on action rates states that action $\langle a, \tilde{\lambda} \rangle$ can be synchronized with action $\langle a, \tilde{\mu} \rangle$ only if $\min(\tilde{\lambda}, \tilde{\mu}) = *$, and the resulting rate is given by $\max(\tilde{\lambda}, \tilde{\mu})$ up to normalization. In other words, in a synchronization at most one active action can be involved and its rate determines the rate of the synchronization itself, up to normalization. The main reason behind the adoption of such a synchronization discipline on action rates is its simplicity, both from the modeling point of view and from the semantic treatment point of view. In Section 4.6 we shall investigate the expressive power resulting from this apparently restrictive rule, while the need for normalization is explained in Section 3.

Finally, let partial function $Def : Const \twoheadrightarrow \mathscr{L}$ be a set of *defining equations* of the form $A \triangleq E$. In order to guarantee the correctness of recursive definitions given by means of constants, we restrict ourselves to terms that are closed and guarded.

Definition 2.2. Let $E \in \mathscr{L}$ and $A \triangleq E' \in Def$, and let us denote by $\equiv$ the syntactical equivalence between terms. The term $E\langle A := E' \rangle$ obtained from E by replacing each occurrence of A with E' is defined by induction on the syntactical structure of E as follows:

- $\underline{0}\langle A := E' \rangle \equiv \underline{0}$
- $(\langle a, \tilde{\lambda} \rangle .E)\langle A := E' \rangle \equiv \langle a, \tilde{\lambda} \rangle .(E\langle A := E' \rangle)$
- $(E/L)\langle A := E' \rangle \equiv (E\langle A := E' \rangle)/L$
- $(E[\varphi])\langle A := E' \rangle \equiv (E\langle A := E' \rangle)[\varphi]$
- $(E_1 + E_2)\langle A := E' \rangle \equiv (E_1\langle A := E' \rangle) + (E_2\langle A := E' \rangle)$

- $(E_1 \|_S E_2)\langle A := E' \rangle \equiv (E_1\langle A := E' \rangle) \|_S (E_2\langle A := E' \rangle)$
- $B\langle A := E' \rangle \equiv \begin{cases} E' & \text{if } B \equiv A \\ B & \text{if } B \not\equiv A \end{cases}$

Definition 2.3. Let $E \in \mathscr{L}$, and let us denote by st the relation subterm-of. The set of terms obtained from E by repeatedly replacing constants by the right-hand side terms of their defining equations in Def is defined by

$$Subst_{Def}(E) = \bigcup_{n \in \mathbb{N}} Subst^n_{Def}(E)$$

where

$$Subst^n_{Def}(E) = \begin{cases} \{E\} & \text{if } n = 0, \\ \{F \in \mathscr{L} \mid F \equiv G\langle A := E' \rangle \wedge G & \\ \qquad \in Subst^{n-1}_{Def}(E) \wedge A \; st \; G \wedge A \triangleq E' \in Def\} & \text{if } n > 0. \end{cases}$$

Definition 2.4. The set of constants occurring in $E \in \mathscr{L}$ w.r.t. Def is defined by

$$Const_{Def}(E) = \{A \in Const \mid \exists F \in Subst_{Def}(E).\, A \; st \; F\}.$$

Definition 2.5. A term $E \in \mathscr{L}$ is *guardedly closed w.r.t. Def* if and only if for each constant $A \in Const_{Def}(E)$

- A is equipped in Def with defining equation $A \triangleq E'$, and
- there exists $F \in Subst_{Def}(E')$ such that, whenever an instance of a constant B satisfies $B \; st \; F$, then the same instance satisfies $B \; st \; \langle a, \tilde{\lambda} \rangle . G \; st \; F$.

We denote by $\mathscr{G}$ the set of terms in $\mathscr{L}$ that are guardedly closed w.r.t Def.

From now on, Def will not be explicitly mentioned as it will be clear from the context.

2.3. *Semantic model: race policy, preselection policy and interleaving*

In this section we address the problem of representing the semantic model of an EMPA term by examining three cases: sequence, choice, parallelism.

Sequence. Consider term $\langle a, \lambda \rangle . \underline{0}$. This term represents a system that can execute an action having type a whose duration is exponentially distributed with rate λ. Its semantic model can be represented as a *rooted labeled transition system* (LTS), i.e. a graph whose nodes describe the states of the system and whose transitions describe state changes. In this case, the LTS has two states that correspond to terms $\langle a, \lambda \rangle . \underline{0}$ and $\underline{0}$, respectively. At first sight, in this case we should have infinitely many transitions labeled with a from the first state to the second state, i.e. one transition for every possible duration of the action. Fortunately, this infinitely branching structure can be symbolically replaced by means of one single transition labeled with a, λ: the rate of the exponential distribution describing the duration of the action at hand contains all the information we need from the point of view of the underlying performance model, which is a Markov chain as we shall see in Section 4.

Choice. Consider term $\langle a, \lambda \rangle.\underline{0} + \langle b, \mu \rangle.\underline{0}$. This term represents a system that can execute two alternative exponentially timed actions. Like in [1, 25, 29], as a mechanism for choosing the action to execute we adopt the *race policy*: the action sampling the least duration succeeds. The adoption of the race policy implies that (i) the random variable describing the *sojourn time* in the state corresponding to the term above is the minimum of the exponentially distributed random variables describing the durations of the two actions, and (ii) the *execution probability* of the two actions is determined as well by the exponentially distributed random variables describing their durations. In order to compute the two quantities above, we exploit the property that the minimum of n independent exponentially distributed random variables is an exponentially distributed random variable whose rate is the sum of the n original rates [34]. As a consequence, for the term above we have that the sojourn time of the corresponding state is exponentially distributed with rate $\lambda + \mu$ (hence the mean sojourn time is $1/(\lambda+\mu)$) and the execution probabilities of the two actions are $\lambda/(\lambda+\mu)$ and $\mu/(\lambda+\mu)$, respectively. Since these two probabilities are nonzero, the semantic model of the term at hand is a LTS comprising two states that correspond to terms $\langle a, \lambda \rangle.\underline{0} + \langle b, \mu \rangle.\underline{0}$ and $\underline{0}$, respectively, as well as two transitions from the first state to the second state labeled with a, λ and b, μ, respectively.

Another important consequence of the adoption of the race policy is the fact that *immediate actions take precedence over exponentially timed actions*. If we consider term $\langle a, \lambda \rangle.\underline{0} + \langle b, \infty_{l,w} \rangle.\underline{0}$, then the underlying semantic model is a LTS with only one transition labeled with $b, \infty_{l,w}$ because action $\langle b, \infty_{l,w} \rangle$ has duration zero whereas action $\langle a, \lambda \rangle$ cannot sample duration zero from the associated exponential distribution.

Consider now term $\langle a, \infty_{l,w} \rangle.\underline{0} + \langle b, \infty_{l',w'} \rangle.\underline{0}$. This term represents a system that can execute two alternative immediate actions. Since both actions have the same duration hence the race policy does not apply, the action to execute is chosen according to the *preselection policy*: only the actions having the highest priority level are executable, and each of them is given a probability execution proportional to its own weight. The semantic model of the term above is a LTS with two states that correspond to terms $\langle a, \infty_{l,w} \rangle.\underline{0} + \langle b, \infty_{l',w'} \rangle.\underline{0}$ and $\underline{0}$, respectively, and the sojourn time in the first state is zero. If $l > l'$ ($l' > l$), then there is only one transition from the first state to the second state which is labeled with $a, \infty_{l,w}$ ($b, \infty_{l',w'}$). If $l = l'$, then there are two transitions from the first state to the second state which are labeled with $a, \infty_{l,w}$ and $b, \infty_{l',w'}$, respectively: the execution probability of the first transition is $w/(w + w')$ while the execution probability of the second transition is $w'/(w + w')$.

Finally, consider term $\langle a, * \rangle.\underline{0} + \langle b, * \rangle.\underline{0}$. This term represents a system that can execute two alternative passive actions. Since the duration of passive actions is undefined, and they are assigned neither priority levels nor weights, they can be undergone to neither the race policy nor the preselection policy. This means that passive actions can be viewed as actions of classical process algebras, hence the term above expresses a purely nondeterministic choice, where nondeterminism refers to the absence of a mechanism that specifies how the choice is resolved.

In conclusion, we observe that the alternative composition operator is *parametric in the nature of the choice*, because in its simpler form it describes a choice between two actions which is:

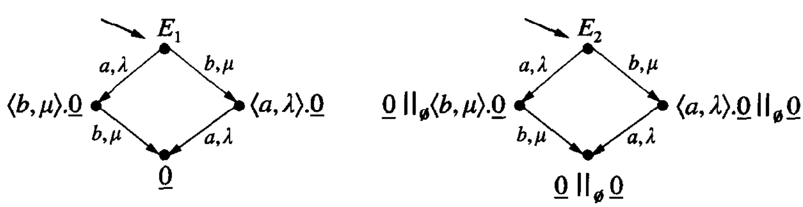
- *Prioritized* if the two actions are active and have different priority levels. The choice is solved *implicitly* if it concerns an exponentially timed action and an immediate action (because the choice is implicitly determined by the race policy), *explicitly* if it concerns two immediate actions having different priority levels (because the priority levels explicitly determine the choice).
- *Probabilistic* if the two actions are active and have the same priority level. The choice is solved *implicitly* if it concerns two exponentially timed actions (because their execution probabilities are implicitly determined by their durations due to the race policy), *explicitly* if it concerns two immediate actions having the same priority level (because their execution probabilities are explicitly determined by their weights).
- *Nondeterministic* if the two actions are passive, because in such a case neither the race policy nor the preselection policy applies.

Parallelism. Consider terms

$$E_1 \equiv \langle a, \lambda \rangle.\langle b, \mu \rangle.\underline{0} + \langle b, \mu \rangle.\langle a, \lambda \rangle.\underline{0}$$

$$E_2 \equiv \langle a, \lambda \rangle.\underline{0} \parallel_{\emptyset} \langle b, \mu \rangle.\underline{0}$$

Term E_1 represents a system that can execute either $\langle a, \lambda \rangle$ followed by $\langle b, \mu \rangle$ or $\langle b, \mu \rangle$ followed by $\langle a, \lambda \rangle$, while term E_2 represents a system that can execute $\langle a, \lambda \rangle$ in parallel with $\langle b, \mu \rangle$. Following the *interleaving* style of classical process algebras, we propose the following two LTSs as semantic models for E_1 and E_2, respectively:



The isomorphism between the two LTSs is correct from the functional point of view by definition of interleaving, and also from the performance point of view: due to the *memoryless property of exponential distributions* [34], if we assume that E_2 completes a before b, then the residual time to the completion of b is still exponentially distributed with rate μ, so the rate labeling the transition from state $\underline{0} \parallel_{\emptyset} \langle b, \mu \rangle.\underline{0}$ to state $\underline{0} \parallel_{\emptyset} \underline{0}$ is μ itself instead of μ conditional on λ. In other words, this means that the semantics for EMPA can be defined in the interleaving style as in the case of classical process algebras.

Furthermore, it is worth noting that in the right-hand LTS there is no transition from state E_2 to state $\underline{0}\,\|_{\emptyset}\,\underline{0}$ recording the possible simultaneous completion of a and b. The reason is that the probability of such a simultaneous completion is zero because the durations of a and b are described by means of *continuous* probability distribution functions.

3. Integrated interleaving semantics of EMPA terms

The main problem to tackle when defining the semantics for EMPA is that the actions executable by a given term may have different priority levels, and only those having the highest priority level are actually executable. Let us call *potential move* of a given term a pair composed of (i) an action executable by the term, and (ii) a derivative term obtained by executing that action. To solve the problem above, we compute inductively the multiset[1] of the potential moves of a given term regardless of priority levels, and then we select those having the highest priority level. This is motivated in our framework by the fact that the actual executability as well as the execution probability of an action depend upon *all* the actions that are executable at the same time when it is executable: only if we know all the potential moves of a given term, we can correctly determine the transitions of the corresponding state and their rates. We denote by $PMove = Act \times \mathcal{G}$ the set of all the potential moves.

The formal definition of the integrated interleaving semantics for EMPA is based on the transition relation $\longrightarrow$, which is the least subset of $\mathcal{G} \times Act \times \mathcal{G}$ satisfying the inference rule reported in the first part of Table 1. This rule selects the potential moves that have the highest priority level (or are passive), and then merges together those having the same action type, the same priority level and the same derivative term in order to produce *standard LTSs*. The first operation is carried out through functions $Select : \mathcal{M}u_{fin}(PMove) \longrightarrow \mathcal{M}u_{fin}(PMove)$ and $PL : Act \longrightarrow APLev$, which are defined in the third part of Table 1. The second operation is carried out through function $Melt : \mathcal{M}u_{fin}(PMove) \longrightarrow \mathcal{P}_{fin}(PMove)$ and partial function $Min : (ARate \times ARate) \multimap ARate$, which are defined in the fourth part of Table 1. The name *Min* should recall the adoption of the race policy: the minimum of a set of random variables has to be computed. We regard *Min* as an associative and commutative operation, thus we take the liberty to apply it to multisets of rates.

[1] We use "$\{\!|$" and "$|\!\}$" as brackets for multisets, "$\_ \oplus \_$" to denote multiset union, $\mathcal{M}u_{fin}(S)$ ($\mathcal{P}_{fin}(S)$) to denote the collection of finite multisets (sets) over set S, $M(s)$ to denote the multiplicity of element s in multiset M, and $\pi_i(M)$ to denote the multiset obtained by projecting the tuples in multiset M on their ith component. Thus, e.g., $(\pi_1(PM_2))(\langle a, *\rangle)$ in the fifth part of Table 1 denotes the multiplicity of tuples of PM_2 whose first component is $\langle a, *\rangle$.

Table 1
Inductive rules for EMPA integrated interleaving semantics

$$\frac{(\langle a,\tilde{\lambda}\rangle,E')\in Melt(Select(PM(E)))}{E\xrightarrow{a,\tilde{\lambda}}E'}$$

$$PM(\underline{0})=\emptyset$$

$$PM(\langle a,\tilde{\lambda}\rangle.E)=\{\!|(\langle a,\tilde{\lambda}\rangle,E)|\!\}$$

$$\begin{aligned}PM(E/L)=\ &\{\!|(\langle a,\tilde{\lambda}\rangle,E'/L)\mid(\langle a,\tilde{\lambda}\rangle,E')\in PM(E)\wedge a\notin L|\!\}\oplus\\&\{\!|(\langle\tau,\tilde{\lambda}\rangle,E'/L)\mid(\langle a,\tilde{\lambda}\rangle,E')\in PM(E)\wedge a\in L|\!\}\end{aligned}$$

$$PM(E[\varphi])=\{\!|(\langle\varphi(a),\tilde{\lambda}\rangle,E'[\varphi])\mid(\langle a,\tilde{\lambda}\rangle,E')\in PM(E)|\!\}$$

$$PM(E_1+E_2)=PM(E_1)\oplus PM(E_2)$$

$$\begin{aligned}PM(E_1\parallel_S E_2)=\ &\{\!|(\langle a,\tilde{\lambda}\rangle,E_1'\parallel_S E_2)\mid a\notin S\wedge(\langle a,\tilde{\lambda}\rangle,E_1')\in PM(E_1)|\!\}\oplus\\&\{\!|(\langle a,\tilde{\lambda}\rangle,E_1\parallel_S E_2')\mid a\notin S\wedge(\langle a,\tilde{\lambda}\rangle,E_2')\in PM(E_2)|\!\}\oplus\\&\{\!|(\langle a,\tilde{\gamma}\rangle,E_1'\parallel_S E_2')\mid a\in S\wedge\\&\qquad(\langle a,\tilde{\lambda}_1\rangle,E_1')\in PM(E_1)\wedge\\&\qquad(\langle a,\tilde{\lambda}_2\rangle,E_2')\in PM(E_2)\wedge\\&\qquad\tilde{\gamma}=Norm(a,\tilde{\lambda}_1,\tilde{\lambda}_2,PM(E_1),PM(E_2))|\!\}\end{aligned}$$

$$PM(A)=PM(E)\qquad\text{if } A\triangleq E$$

$$\begin{aligned}Select(PM)=\{\!|(\langle a,\tilde{\lambda}\rangle,E)\in PM\mid\forall(\langle b,\tilde{\mu}\rangle,E')\in PM.\,&PL(\langle a,\tilde{\lambda}\rangle)\geqslant PL(\langle b,\tilde{\mu}\rangle)\vee\\&PL(\langle a,\tilde{\lambda}\rangle)=-1|\!\}\end{aligned}$$

$$PL(\langle a,*\rangle)=-1\qquad PL(\langle a,\lambda\rangle)=0\qquad PL(\langle a,\infty_{l,w}\rangle)=l$$

$$\begin{aligned}Melt(PM)=\{(\langle a,\tilde{\lambda}\rangle,E)\mid(\langle a,\tilde{\mu}\rangle,E)\in PM\wedge\\\tilde{\lambda}=Min\{\!|\tilde{\gamma}\mid(\langle a,\tilde{\gamma}\rangle,E)\in PM\wedge PL(\langle a,\tilde{\gamma}\rangle)=PL(\langle a,\tilde{\mu}\rangle)|\!\}\}\end{aligned}$$

$$*\,Min\,*=*\qquad\lambda_1\,Min\,\lambda_2=\lambda_1+\lambda_2\qquad\infty_{l,w_1}\,Min\,\infty_{l,w_2}=\infty_{l,w_1+w_2}$$

$$Norm(a,\tilde{\lambda}_1,\tilde{\lambda}_2,PM_1,PM_2)=\begin{cases}Split(\tilde{\lambda}_1,1/(\pi_1(PM_2))(\langle a,*\rangle)) & \text{if } \tilde{\lambda}_2=*\\Split(\tilde{\lambda}_2,1/(\pi_1(PM_1))(\langle a,*\rangle)) & \text{if } \tilde{\lambda}_1=*\end{cases}$$

$$Split(*,p)=*\qquad Split(\lambda,p)=\lambda\cdot p\qquad Split(\infty_{l,w},p)=\infty_{l,w\cdot p}$$

Example 3.1. If we consider the term

$$E\equiv\langle a,\lambda\rangle.F+\langle a,\lambda\rangle.F$$

then we have two identical potential moves $(\langle a,\lambda\rangle,F)$ which are merged into $(\langle a,2\cdot\lambda\rangle,F)$ by means of *Melt* and *Min*. We would like to point out that producing a single transition $E\xrightarrow{a,\lambda}F$ from the two identical potential moves above is wrong because the average sojourn time in the state corresponding to term E would be altered. Thus, *Melt* and *Min* correctly manage this situation, and do not require to decorate transitions with *auxiliary labels* like in [25] nor to take into account the *multiplicity of transitions* like in [29], thereby allowing for the generation of standard LTSs as integrated semantic models.

The multiset $PM(E) \in \mathcal{M}u_{fin}(PMove)$ of potential moves of $E \in \mathcal{G}$ is defined by structural induction in the second part of Table 1 according to the intuitive meaning of operators explained in Section 2.2. In order to enforce the *bounded capacity assumption* [30], which establishes that the rate at which an activity is carried out cannot be increased by synchronizing it with other activities, in the rule for the parallel composition operator a *normalization* is required which suitably computes the rates of potential moves resulting from the *synchronization of the same active action with several independent or alternative passive actions*. The normalization operates in such a way that applying *Min* to the rates of the synchronizations involving the active action gives as a result the rate of the active action itself, and that each synchronization is assigned the same execution probability. This normalization is carried out through partial function $Norm : (AType \times ARate \times ARate \times \mathcal{M}u_{fin}(PMove) \times \mathcal{M}u_{fin}(PMove)) \multimap ARate$ and function $Split : (ARate \times \mathbb{R}_{]0,1]}) \rightarrow ARate$, which are defined in the fifth part of Table 1. Note that $Norm(a, \tilde{\lambda}_1, \tilde{\lambda}_2, PM_1, PM_2)$ is defined if and only if $\min(\tilde{\lambda}, \tilde{\mu}) = *$, which is the condition on action rates we have required in Section 2.2 in order for a synchronization to be permitted.

Example 3.2. Consider the terms

$$E_1 \equiv \langle a, \lambda \rangle . \underline{0} \parallel_{\{a\}} (\langle a, * \rangle . \underline{0} \parallel_{\emptyset} \langle a, * \rangle . \underline{0})$$
$$E_2 \equiv \langle a, \lambda \rangle . \underline{0} \parallel_{\{a\}} (\langle a, * \rangle . \underline{0} + \langle a, * \rangle . \underline{0})$$

In both cases, the left-hand side operand of " $\parallel_{\{a\}}$ " has one potential move $(\langle a, \lambda \rangle, \underline{0})$ and the right-hand side operand has two potential moves whose action is $\langle a, * \rangle$, hence the whole term has two potential moves whose type is a. Since both terms consist of a single active action which is exponentially timed with rate λ, the rate of each of the two potential moves cannot be λ otherwise the mean sojourn time of the states corresponding to E_1 and E_2 would be $1/(2 \cdot \lambda)$ instead of $1/\lambda$: a normalization must take place so that the sum of the rates of the two potential moves turns out to be λ. Assuming that independent or alternative passive actions have the same execution probability when they are involved in a synchronization, *Norm* computes the rate of each of the potential moves above by dividing λ by the number of independent or alternative passive actions with which the synchronization can take place. As a consequence, the rate of each of the two potential moves is $\lambda/2$.

Definition 3.3. The *integrated interleaving semantics* of $E \in \mathcal{G}$ is the LTS $\mathcal{I}[\![E]\!] = (\uparrow E, Act, \rightarrow_E, E)$ where $\uparrow E$ is the set of states reachable from E, and $\rightarrow_E$ is $\rightarrow$ restricted to $\uparrow E \times Act \times \uparrow E$.

Definition 3.4. $E \in \mathscr{G}$ is *performance closed* if and only if $\mathscr{I}[\![E]\!]$ does not contain passive transitions. We denote by $\mathscr{E}$ the set of performance closed terms of $\mathscr{G}$.

Given a term $E \in \mathscr{G}$, its *integrated* interleaving semantics $\mathscr{I}[\![E]\!]$ fully represents the behavior of E because transitions are decorated by both the action type and the action rate. One can think of obtaining two *projected* semantic models, i.e. the *functional semantics* $\mathscr{F}[\![E]\!]$ and the *performance semantics* $\mathscr{P}[\![E]\!]$, from $\mathscr{I}[\![E]\!]$ by simply dropping action rates and action types, respectively. As a matter of fact, this is the case for the functional semantics.

Definition 3.5. The *functional semantics* of $E \in \mathscr{G}$ is the LTS $\mathscr{F}[\![E]\!] = (\uparrow E, AType, \longrightarrow_{E,\mathscr{F}}, E)$ where $\longrightarrow_{E,\mathscr{F}}$ is $\longrightarrow_E$ restricted to $\uparrow E \times AType \times \uparrow E$.

The definition of the performance semantics requires instead a more careful treatment which is deferred to the next section, and is given *only for performance closed terms* since these are completely specified from the performance standpoint.

4. EMPA kernels

Due to the coexistence of exponentially timed actions, prioritized weighted immediate actions, and passive actions, EMPA can be viewed as being made out of four kernels (see Fig. 1): a *nondeterministic kernel*, a *prioritized kernel*, a *probabilistic kernel*, and an *exponentially timed kernel*.

In Sections 4.1–4.4 we examine each of these kernels separately by presenting how the functional semantics (in the case of the nondeterministic kernel and the prioritized kernel) or the integrated interleaving semantics (in the case of the probabilistic kernel and the exponentially timed kernel) specializes to the kernel at hand, by defining the performance semantics in the case of the probabilistic kernel and the exponentially timed kernel, and by making some comparison with related process algebras appeared in the literature. In Section 4.5 we consider the interplay between the probabilistic kernel and the exponentially timed kernel both from the point of view of the performance semantics and from the point of view of the probability distribution functions that can be modeled. In Section 4.6 we emphasize the role played by passive actions and we report some remarks on the synchronization discipline on action rates adopted in EMPA. Finally, in Section 4.7 we show an example that should highlight the considerable expressive power of EMPA resulting from the coexistence of the four kernels.

4.1. Nondeterministic kernel

The *nondeterministic kernel* $EMPA_{nd}$ is the sublanguage of EMPA obtained by considering only passive actions. Since the duration of passive actions is completely unspecified, $EMPA_{nd}$ is a classical process algebra and allows for pure nondeterminism. We define below the syntax and the functional semantics of $EMPA_{nd}$ terms.

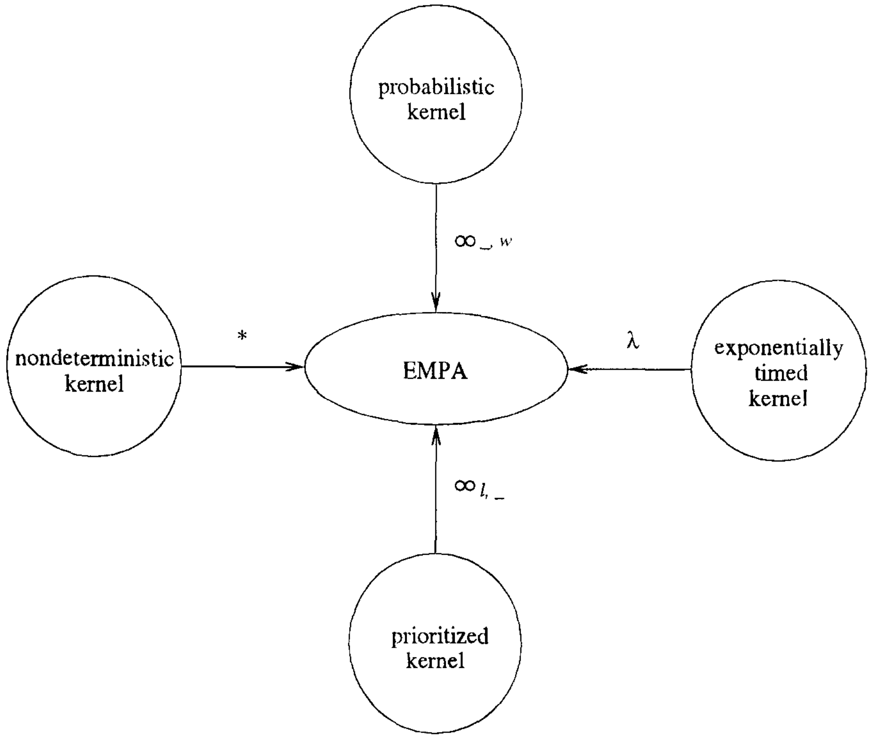


Fig. 1. EMPA kernels.

Definition 4.1. The set $\mathscr{L}_{nd}$ of process terms of EMPA$_{nd}$ is generated by the following syntax:

$$E ::= \underline{0} \mid \langle a, * \rangle.E \mid E/L \mid E[\varphi] \mid E + E \mid E \|_S E \mid A$$

where $L, S \subseteq AType - \{\tau\}$. We denote by $\mathscr{G}_{nd}$ the set of guardedly closed terms of $\mathscr{L}_{nd}$.

Definition 4.2. The functional semantics of $E \in \mathscr{G}_{nd}$ is the LTS $\mathscr{F}_{nd}[\![E]\!] = (\uparrow E, AType, \longrightarrow_{E,\mathscr{F}_{nd}}, E)$ where $\longrightarrow_{E,\mathscr{F}_{nd}}$ is the restriction of $\longrightarrow_{\mathscr{F}_{nd}}$ (defined in Table 2) to $\uparrow E \times AType \times \uparrow E$.

Proposition 4.3. *For any $E \in \mathscr{G}_{nd}$, $\mathscr{F}_{nd}[\![E]\!]$ is isomorphic to $\mathscr{F}[\![E]\!]$.*

Proof. It follows immediately from the fact that, for $E \in \mathscr{G}_{nd}$, function *Norm* always evaluates to "$*$", function *Select* boils down to the identity function, and the application of function *Melt* is irrelevant because the effect of $* \, Min * = *$ would stem from the fact that $\longrightarrow$ is a transition relation instead of a multitransition relation. $\square$

Table 2
Inductive rules for EMPA_{nd} functional semantics

$$\langle a,*\rangle.E \xrightarrow{a}_{\mathscr{F}_{nd}} E$$

$$\frac{E \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{E/L \xrightarrow{a}_{\mathscr{F}_{nd}} E'/L} \text{ if } a \notin L \qquad \frac{E \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{E/L \xrightarrow{\tau}_{\mathscr{F}_{nd}} E'/L} \text{ if } a \in L$$

$$\frac{E \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{E[\varphi] \xrightarrow{\varphi(a)}_{\mathscr{F}_{nd}} E'[\varphi]}$$

$$\frac{E_1 \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{E_1 + E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E'} \qquad \frac{E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{E_1 + E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E'}$$

$$\frac{E_1 \xrightarrow{a}_{\mathscr{F}_{nd}} E_1'}{E_1 \parallel_S E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E_1' \parallel_S E_2} \text{ if } a \notin S \qquad \frac{E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E_2'}{E_1 \parallel_S E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E_1 \parallel_S E_2'} \text{ if } a \notin S$$

$$\frac{E_1 \xrightarrow{a}_{\mathscr{F}_{nd}} E_1' \quad E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E_2'}{E_1 \parallel_S E_2 \xrightarrow{a}_{\mathscr{F}_{nd}} E_1' \parallel_S E_2'} \text{ if } a \in S$$

$$\frac{E \xrightarrow{a}_{\mathscr{F}_{nd}} E'}{A \xrightarrow{a}_{\mathscr{F}_{nd}} E'} \text{ if } A \triangleq E$$

We conclude by observing that the operators of EMPA_{nd} are a mix of the operators of CCS [40] and CSP [31]: the functional abstraction operator coincides with the hiding operator of CSP, the functional relabeling operator coincides with the relabeling operator of CCS, and the parallel composition operator reduces to the parallel composition operator of CSP since the constraint on action rates is always satisfied in EMPA_{nd}. Therefore, all the results and analysis techniques developed for classical process algebras can be applied to EMPA_{nd}.

4.2. Prioritized kernel

The *prioritized kernel* $\mathrm{EMPA}_{pt,w}$ is the sublanguage of EMPA obtained by considering only immediate actions having the same weight w and passive actions. Since each immediate action is given a priority level, $\mathrm{EMPA}_{pt,w}$ is a prioritized process algebra: in this framework, the priority level of a passive action is considered to be unspecified, and the weight and the duration of an immediate action are ignored. We define below the syntax and the functional semantics of $\mathrm{EMPA}_{pt,w}$ terms.

Definition 4.4. The set $\mathscr{L}_{pt,w}$ of process terms of $\mathrm{EMPA}_{pt,w}$ is generated by the following syntax

$$E ::= \underline{0} \mid \langle a,\tilde{\lambda}\rangle.E \mid E/L \mid E[\varphi] \mid E+E \mid E \parallel_S E \mid A$$

Table 3
Inductive rules for $\mathrm{EMPA}_{pt,w}$ functional semantics

$$\frac{(\langle a,\tilde{\lambda}\rangle,E') \in Select(PM(E))}{E \xrightarrow{a}_{\mathscr{F}_{pt,w}} E'}$$

$$PM(\underline{0}) = \emptyset$$

$$PM(\langle a,\tilde{\lambda}\rangle.E) = \{(\langle a,\tilde{\lambda}\rangle,E)\}$$

$$\begin{aligned} PM(E/L) = {} & \{(\langle a,\tilde{\lambda}\rangle,E'/L) \mid (\langle a,\tilde{\lambda}\rangle,E') \in PM(E) \wedge a \notin L\} \\ & \cup \{(\langle \tau,\tilde{\lambda}\rangle,E'/L) \mid (\langle a,\tilde{\lambda}\rangle,E') \in PM(E) \wedge a \in L\} \end{aligned}$$

$$PM(E[\varphi]) = \{(\langle \varphi(a),\tilde{\lambda}\rangle,E'[\varphi]) \mid (\langle a,\tilde{\lambda}\rangle,E') \in PM(E)\}$$

$$PM(E_1 + E_2) = PM(E_1) \cup PM(E_2)$$

$$\begin{aligned} PM(E_1 \parallel_S E_2) = {} & \{(\langle a,\tilde{\lambda}\rangle,E_1' \parallel_S E_2) \mid a \notin S \wedge (\langle a,\tilde{\lambda}\rangle,E_1') \in PM(E_1)\} \\ & \cup \{(\langle a,\tilde{\lambda}\rangle,E_1 \parallel_S E_2') \mid a \notin S \wedge (\langle a,\tilde{\lambda}\rangle,E_2') \in PM(E_2)\} \\ & \cup \{(\langle a,\tilde{\gamma}\rangle,E_1' \parallel_S E_2') \mid a \in S \wedge \\ & \qquad (\langle a,\tilde{\lambda}_1\rangle,E_1') \in PM(E_1) \\ & \qquad \wedge (\langle a,\tilde{\lambda}_2\rangle,E_2') \in PM(E_2) \\ & \qquad \wedge ((\tilde{\lambda}_1 = * \wedge \tilde{\gamma} = \tilde{\lambda}_2) \vee (\tilde{\lambda}_2 = * \wedge \tilde{\gamma} = \tilde{\lambda}_1))\} \end{aligned}$$

$$PM(A) = PM(E) \quad \text{if } A \triangleq E$$

where $\tilde{\lambda} \in \{\infty_{l,w} \mid l \in \mathbb{N}_+\} \cup \{*\}$ and $L,S \subseteq AType - \{\tau\}$. We denote by $\mathscr{G}_{pt,w}$ the set of guardedly closed terms of $\mathscr{L}_{pt,w}$.

Definition 4.5. The functional semantics of $E \in \mathscr{G}_{pt,w}$ is the LTS $\mathscr{F}_{pt,w}[\![E]\!] = (\uparrow E, AType, \longrightarrow_{E,\mathscr{F}_{pt,w}}, E)$ where $\longrightarrow_{E,\mathscr{F}_{pt,w}}$ is the restriction of $\longrightarrow_{\mathscr{F}_{pt,w}}$ (defined in Table 3) to $\uparrow E \times AType \times \uparrow E$.

Proposition 4.6. *For any $E \in \mathscr{G}_{pt,w}$, $\mathscr{F}_{pt,w}[\![E]\!]$ is isomorphic to $\mathscr{F}[\![E]\!]$.*

Proof. It follows immediately from the fact that, for $E \in \mathscr{G}_{pt,w}$, function *Norm* has the same effect of side condition $(\tilde{\lambda}_1 = * \wedge \tilde{\gamma} = \tilde{\lambda}_2) \vee (\tilde{\lambda}_2 = * \wedge \tilde{\gamma} = \tilde{\lambda}_1)$, and the application of function *Melt* is irrelevant, because we are interested in action types only. □

Since every nonpassive action is given an explicit priority level, from the *syntactical* point of view $\mathrm{EMPA}_{pt,w}$ is similar to the proposal of [14] where for each action type both a *prioritized* version and an *unprioritized* version are provided. The difference is that in [14] unprioritized actions are preempted only by *internal* prioritized actions in order to achieve compositionality, while the semantic treatment of priorities in $\mathrm{EMPA}_{pt,w}$ is quite different because the actual priority level of an action is independent of its visibility. As we shall see in Section 5, in order to achieve compositionality in the presence of actions having different priority levels, we extend EMPA with a *priority operator* in the style of [4] such that the priority structure is enforced only within its scope, and we show that every EMPA term can be thought of as having a single occurrence of the priority operator on top of it. We claim that our approach is convenient from the *modeling* point of view because the priority level of each action is exactly that specified by the designer: to enforce it, there is no need to introduce artificial prioritized τ loops in the algebraic description of the system like in [14], nor to burden the algebraic description with occurrences of the priority operator like in [4].

Furthermore, we observe that $\mathrm{EMPA}_{pt,w}$ is different from the proposal of [13], where a *prioritized choice operator* is explicitly defined, from the proposal of [57], where priority is expressed as *extremal probability* and the computation proceeds in *locksteps*, from CCSR [17], where priority is used to arbitrate between simultaneous *resource requests* and *lockstep parallelism* is considered, and from CCS^{prio} [15], where actions are allowed to preempt others only at the same site so as to capture a notion of *localized precedence*.

Finally, if we consider the features a prioritized process algebra should possess according to [57], we have that the priority relation of $\mathrm{EMPA}_{pt,w}$ is *globally dynamic*, i.e. it may be the case that an action with type a has priority over an action with type b in one state, and the converse in some other state. On the other hand, the priority relation of $\mathrm{EMPA}_{pt,w}$ cannot define *arbitrary partial orders* because the leveled priority structure causes incomparable actions to have the same priority. As an example, it is not possible in $\mathrm{EMPA}_{pt,w}$ to express the fact that, in a given state, action type a takes precedence over action types b and c while action type d takes precedence only over action type c.

4.3. Probabilistic kernel

The *probabilistic kernel* $\mathrm{EMPA}_{pb,l}$ is the sublanguage of EMPA obtained by considering only immediate actions having the same priority level l and passive actions. Since each immediate action is given a weight, $\mathrm{EMPA}_{pb,l}$ is a probabilistic process algebra: in this framework, the weight of a passive action is considered to be unspecified, and the priority level and the duration of an immediate action are ignored. We define below the syntax, the integrated semantics, the functional semantics and the performance semantics of $\mathrm{EMPA}_{pb,l}$ terms.

Definition 4.7. The set $\mathscr{L}_{pb,l}$ of process terms of $\mathrm{EMPA}_{pb,l}$ is generated by the following syntax

$$E ::= \underline{0} \mid \langle a, \tilde{\lambda} \rangle .E \mid E/L \mid E[\varphi] \mid E + E \mid E \parallel_S E \mid A$$

where $\tilde{\lambda} \in \{\infty_{l,w} \mid w \in \mathbb{R}_+\} \cup \{*\}$ and $L, S \subseteq AType - \{\tau\}$. We denote by $\mathscr{G}_{pb,l}$ the set of guardedly closed terms of $\mathscr{L}_{pb,l}$, and by $\mathscr{E}_{pb,l}$ the set of performance closed terms of $\mathscr{G}_{pb,l}$.

Table 4
Rule for $\mathrm{EMPA}_{pb,l}$ integrated semantics

$$\frac{(\langle a,\tilde{\lambda}\rangle, E') \in \mathit{Melt}(PM(E))}{E \xrightarrow{a,\tilde{\lambda}}_{\mathscr{I}_{pb,l}} E'}$$

Definition 4.8. The integrated semantics of $E \in \mathscr{G}_{pb,l}$ is the LTS $\mathscr{I}_{pb,l}[\![E]\!] = (\uparrow E, Act, \longrightarrow_{E,\mathscr{I}_{pb,l}}, E)$ where $\longrightarrow_{E,\mathscr{I}_{pb,l}}$ is the restriction of $\longrightarrow_{\mathscr{I}_{pb,l}}$ (defined in Table 4) to $\uparrow E \times Act \times \uparrow E$.

Definition 4.9. The functional semantics of $E \in \mathscr{G}_{pb,l}$ is the LTS $\mathscr{F}_{pb,l}[\![E]\!] = (\uparrow E, AType, \longrightarrow_{E,\mathscr{F}_{pb,l}}, E)$ where $\longrightarrow_{E,\mathscr{F}_{pb,l}}$ is the restriction of $\longrightarrow_{E,\mathscr{I}_{pb,l}}$ to $\uparrow E \times AType \times \uparrow E$.

Proposition 4.10. *For any $E \in \mathscr{G}_{pb,l}$, $\mathscr{I}_{pb,l}[\![E]\!]$ is isomorphic to $\mathscr{I}[\![E]\!]$.*

Proof. It is a straightforward consequence of the fact that, for $E \in \mathscr{G}_{pb,l}$, function *Select* boils down to the identity function. $\square$

Corollary 4.11. *For any $E \in \mathscr{G}_{pb,l}$, $\mathscr{F}_{pb,l}[\![E]\!]$ is isomorphic to $\mathscr{F}[\![E]\!]$.*

The performance semantics of terms in $\mathscr{E}_{pb,l}$ is defined by considering the execution probability of each immediate transition exiting from a given state: since such a probability is proportional to the weight of the transition and depends only on the current state, the underlying performance model is a *homogeneous discrete-time Markov chain* (HDTMC) [34]. Every HDTMC can be formalized by means of a *probabilistically rooted labeled transition system* (PLTS), i.e. a LTS where the initial state is replaced by a probability mass function specifying for each state the probability that it is the initial one. The labels of the transitions of such a PLTS are the execution probabilities of the transitions themselves: they are computed according to the weights of the corresponding actions because of the adoption of the preselection policy.

Definition 4.12. The *Markovian semantics* of $E \in \mathscr{E}_{pb,l}$ is the PLTS $\mathscr{M}_{pb,l}[\![E]\!] = (\uparrow E, \mathbb{R}_{]0,1]}, \longrightarrow_{E,\mathscr{M}_{pb,l}}, P_{E,\mathscr{M}_{pb,l}})$ where

- $\longrightarrow_{E,\mathscr{M}_{pb,l}}$ is the least subset of $\uparrow E \times \mathbb{R}_{]0,1]} \times \uparrow E$ such that $F \xrightarrow{p}_{E,\mathscr{M}_{pb,l}} F'$ whenever

$$p = \sum \{\!| w \mid F \xrightarrow{a,\infty_{l,w}}_{E,\mathscr{I}_{pb,l}} F' \wedge a \in AType |\!\} \Big/ \sum \{\!| w \mid F \xrightarrow{a,\infty_{l,w}}_{E,\mathscr{I}_{pb,l}} F'' \wedge a \in AType \wedge F'' \in \uparrow E |\!\}$$

- $P_{E,\mathscr{M}_{pb,l}} : \uparrow E \longrightarrow \mathbb{R}_{[0,1]}$, $P_{E,\mathscr{M}_{pb,l}}(F) = \begin{cases} 1 & \text{if } F \equiv E \\ 0 & \text{if } F \not\equiv E \end{cases}.$

Unlike probabilistic process algebras which appeared in the literature, $\mathrm{EMPA}_{pb,l}$ does not rely on an explicit probabilistic alternative composition operator since the probabilistic information, i.e. weights, is encoded within actions. Besides, in $\mathrm{EMPA}_{pb,l}$ probabilistic and nondeterministic aspects coexist due to the presence of passive actions, thereby causing $\mathrm{EMPA}_{pb,l}$ to be viewed as a possible syntactical counterpart of formal models for randomized distributed computations such as those defined in [55]. By performing more accurate comparisons, we see that $\mathrm{EMPA}_{pb,l}$ differs from PCCS [32], WSCCS [58] and CPP [36] due to the two reasons above (i.e. absence of a *probabilistic alternative composition operator* and presence of *pure nondeterminism*, though a restricted form of nondeterminism is allowed in CPP) as well as the fact that in these calculi the computation proceeds in *locksteps*: it is however worth noting that the idea of using *weights instead of probabilities* proposed for WSCCS is retained in $\mathrm{EMPA}_{pb,l}$ for operational convenience. If we consider instead probabilistic calculi where the computation does not proceed in locksteps, we realize that $\mathrm{EMPA}_{pb,l}$ differs from prACP [5] and PCSP [56] because of the two usual reasons: furthermore, in prACP a *probabilistic parallel composition operator* is introduced while the corresponding operator of $\mathrm{EMPA}_{pb,l}$ is not probabilistic. Finally, $\mathrm{EMPA}_{pb,l}$ differs from the probabilistic calculus proposed in [49] because there probabilities can be assigned only to internal actions, and from LOTOS-P [39] since there a probabilistic alternative composition operator is introduced.

Following the classification of models of probabilistic processes proposed in [19], it is easily seen that *performance closed* terms of $\mathrm{EMPA}_{pb,l}$ do not represent reactive models because the choice among enabled actions having different types is probabilistic instead of nondeterministic. As a consequence, $\mathscr{E}_{pb,l}$ produces *generative* models because the relative frequency of performing actions having different types is explicitly described. More accurately, $\mathscr{E}_{pb,l}$ is even finer as it can produce *stratified* models where the intended relative frequencies of actions are preserved in a levelwise fashion in the presence of a restriction.

Example 4.13. Consider an operating system having three processes to be multiprogrammed: the garbage collector gc, user process up_1, and user process up_2. Suppose that each process is given 1/3 of the CPU cycles, and that this frequency must be preserved for the garbage collector whenever one of the user processes is denied further access to the machine due to a restriction context. The corresponding $\mathrm{EMPA}_{pb,l}$ term is

$$Sched \triangleq \langle gc, \infty_{l,1}\rangle.Sched + \langle \tau, \infty_{l,2}\rangle.(\langle up_1, \infty_{l,1}\rangle.Sched + \langle up_2, \infty_{l,1}\rangle.Sched).$$

If we consider $Sched \parallel_{\{up_2\}} \underline{0}$, then the execution probability of the action having type gc is still $\frac{1}{3}$, as required.

It is however possible to describe also *reactive* models in $\mathrm{EMPA}_{pb,l}$ by means of the interplay of weighted immediate actions and passive actions, i.e. by means of terms that are not performance closed.

Table 5
Rule for EMPA_{et} integrated semantics

$$\frac{(\langle a,\tilde{\lambda}\rangle, E') \in Melt(PM(E))}{E \xrightarrow{a,\tilde{\lambda}}_{\mathscr{I}_{et}} E'}$$

Example 4.14. Suppose that two people want to flip a coin in order to make a decision: only one of them actually flips the coin, and the outcome of the coin toss is a head with probability $\frac{1}{2}$, a tail with probability $\frac{1}{2}$. This scenario can be modeled as follows:

$$GetCoin \triangleq \langle person_1, *\rangle.FlipCoin + \langle person_2, *\rangle.FlipCoin$$

$$FlipCoin \triangleq \langle flip, \infty_{l,1/2}\rangle.Head + \langle flip, \infty_{l,1/2}\rangle.Tail$$

The underlying model is reactive because the relative frequency with which the two people get the coin is not specified, i.e. it is left to the environment, while the probability of the outcome of the coin toss is governed by the system.

The capability of describing reactive, generative and stratified probabilistic processes by means of $\mathrm{EMPA}_{pb,l}$ terms is one of the most notable consequences of the parametricity (see Section 2.3) of the alternative composition operator of EMPA.

4.4. Exponentially timed kernel

The *exponentially timed kernel* EMPA_{et} is the sublanguage of EMPA obtained by considering only exponentially timed actions and passive actions. Since each active action is given a duration through a probability distribution function, EMPA_{et} is a stochastically timed process algebra. We define below the syntax, the integrated semantics, the functional semantics and the performance semantics of EMPA_{et} terms.

Definition 4.15. The set $\mathscr{L}_{et}$ of process terms of EMPA_{et} is generated by the following syntax

$$E ::= \underline{0} \mid \langle a,\tilde{\lambda}\rangle.E \mid E/L \mid E[\varphi] \mid E+E \mid E\|_S E \mid A$$

where $\tilde{\lambda} \in \mathbb{R}_+ \cup \{*\}$ and $L,S \subseteq AType - \{\tau\}$. We denote by $\mathscr{G}_{et}$ the set of guardedly closed terms of $\mathscr{L}_{et}$, and by $\mathscr{E}_{et}$ the set of performance closed terms of $\mathscr{G}_{et}$.

Definition 4.16. The integrated semantics of $E \in \mathscr{G}_{et}$ is the LTS $\mathscr{I}_{et}[\![E]\!] = (\uparrow E, Act, \longrightarrow_{E,\mathscr{I}_{et}}, E)$ where $\longrightarrow_{E,\mathscr{I}_{et}}$ is the restriction of $\longrightarrow_{\mathscr{I}_{et}}$ (defined in Table 5) to $\uparrow E \times Act \times \uparrow E$.

Definition 4.17. The functional semantics of $E \in \mathscr{G}_{et}$ is the LTS $\mathscr{F}_{et}[\![E]\!] = (\uparrow E, AType, \longrightarrow_{E,\mathscr{F}_{et}}, E)$ where $\longrightarrow_{E,\mathscr{F}_{et}}$ is the restriction of $\longrightarrow_{E,\mathscr{I}_{et}}$ to $\uparrow E \times AType \times \uparrow E$.

Proposition 4.18. *For any $E \in \mathcal{G}_{et}$, $\mathcal{I}_{et}[\![E]\!]$ is isomorphic to $\mathcal{I}[\![E]\!]$.*

Proof. It is a straightforward consequence of the fact that, for $E \in \mathcal{G}_{et}$, function *Select* boils down to the identity function. □

Corollary 4.19. *For any $E \in \mathcal{G}_{et}$, $\mathcal{F}_{et}[\![E]\!]$ is isomorphic to $\mathcal{F}[\![E]\!]$.*

The performance semantics of terms in $\mathcal{E}_{et}$ is defined by considering the rate of each exponentially timed transition exiting from a given state: since the execution probability of each of these transitions depends only on the current state, and the sojourn time of every state is exponentially distributed, the underlying performance model is a *homogeneous continuous-time Markov chain* (HCTMC) [34] which can be formalized by means of a PLTS. The labels of the transitions of such a PLTS are the rates of the corresponding exponentially timed actions.

Definition 4.20. The *Markovian semantics* of $E \in \mathcal{E}_{et}$ is the PLTS $\mathcal{M}_{et}[\![E]\!] = (\uparrow E, \mathbb{R}_+, \longrightarrow_{E,\mathcal{M}_{et}}, P_{E,\mathcal{M}_{et}})$ where:

- $\longrightarrow_{E,\mathcal{M}_{et}}$ is the least subset of $\uparrow E \times \mathbb{R}_+ \times \uparrow E$ such that $F \xrightarrow{\lambda}_{E,\mathcal{M}_{et}} F'$ whenever

$$\lambda = \sum \{\!|\mu \mid F \xrightarrow{a,\mu}_{E,\mathcal{I}_{et}} F' \wedge a \in AType|\!\}$$

- $P_{E,\mathcal{M}_{et}} : \uparrow E \longrightarrow \mathbb{R}_{[0,1]}$, $P_{E,\mathcal{M}_{et}}(F) = \begin{cases} 1 & \text{if } F \equiv E \\ 0 & \text{if } F \not\equiv E \end{cases}$.

We recall that the reason why we consider only exponential distributions in order to specify durations is twofold. On the one hand, the underlying performance models turn out to be HCTMCs, so we can exploit the related theory in order to derive performance measures. On the other hand, the memoryless property of exponential distributions allows us to define the integrated semantics for EMPA through the interleaving approach, as we have noted in Section 2.3.

We would like to point out that EMPA$_{et}$ closely resembles the stochastically timed process algebras MTIPP [25] and PEPA [29]. As we shall see in Section 4.6, the main difference among them is the synchronization discipline on action rates.

4.5. Joining the probabilistic and the exponentially timed kernels

When we consider the whole process algebra, we have to cope with the coexistence of immediate and exponentially timed actions. From the performance standpoint, this means that the sojourn time of some states is exponentially distributed (*tangible states*), while the sojourn time of other states is zero (*vanishing states*). In order to define the performance semantics of terms in $\mathcal{E}$, in [8] we have devised an algorithm that eliminates immediate transitions together with the related vanishing states, and produces HCTMCs.

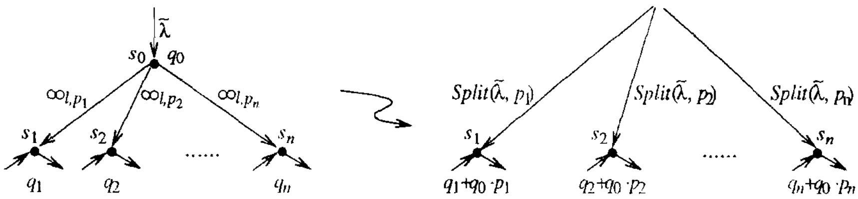


Fig. 2. Graph reduction rule.

Given $E \in \mathscr{E}$, the algorithm comprises several steps. The first step consists of dropping action types, removing selfloops composed of an immediate transition (hereafter called immediate selfloops for short), changing the weight of each immediate transition into the corresponding execution probability, and determining the initial state probability function. Formally, from the LTS $\mathscr{I}[\![E]\!] = (\uparrow E, Act, \longrightarrow_E, E)$ we obtain the PLTS $\mathscr{P}_1[\![E]\!] = (S_{E,1}, \mathbb{R}_+ \cup Inf, \longrightarrow_{E,1}, P_{E,1})$ where[2]

- $S_{E,1} = \uparrow E$.
- Let $PM_1(s) = Melt(\{\!|(\tilde{\lambda}, s') \mid s \xrightarrow{a,\tilde{\lambda}}_E s' \wedge a \in AType|\!\})$ for any $s \in S_{E,1}$. Then $\longrightarrow_{E,1}$ is the least subset of $S_{E,1} \times (\mathbb{R}_+ \cup Inf) \times S_{E,1}$ such that:
  - If s is tangible and $(\lambda, s') \in PM_1(s)$, then $s \xrightarrow{\lambda}_{E,1} s'$.
  - If s is vanishing and in $PM_1(s)$ there are exactly $m \geqslant 1$ potential moves (∞_{l,w_j}, s_j), $1 \leqslant j \leqslant m$, such that $s_j \not\equiv s$, then there are m transitions $s \xrightarrow{\infty_{l,w_j/w}}_{E,1} s_j$, $1 \leqslant j \leqslant m$, where $w = \sum_{j=1}^{m} w_j$.
- $P_{E,1} : S_{E,1} \longrightarrow \mathbb{R}_{[0,1]}$, $P_{E,1}(s) = \begin{cases} 1 & \text{if } s \equiv E \\ 0 & \text{if } s \not\equiv E \end{cases}$.

The kth step, $k \geqslant 2$, handles a vanishing state by eliminating the state itself as well as its outgoing immediate transitions, splitting the transitions entering the vanishing state, removing immediate selfloops created by splitting immediate transitions entering the vanishing state and exiting from states reached by the eliminated immediate transitions, and distributing the initial state probability associated with the vanishing state among the states reached by the eliminated immediate transitions. Formally, if we assume that the vanishing state considered at the kth step is the one shown in Fig. 2, we build PLTS $\mathscr{P}_k[\![E]\!] = (S_{E,k}, \mathbb{R}_+ \cup Inf, \longrightarrow_{E,k}, P_{E,k})$ where

- $S_{E,k} = S_{E,k-1} - \{s_0\}$.
- Let $PM_k(s) = Melt(\{\!|(\tilde{\lambda}, s') \mid s \xrightarrow{\tilde{\lambda}}_{E,k-1} s' \wedge s' \not\equiv s_0|\!\} \oplus \{\!|(Split(\tilde{\lambda}, p_i), s_i) \mid s \xrightarrow{\tilde{\lambda}}_{E,k-1} s_0 \wedge 1 \leqslant i \leqslant n|\!\})$ for any $s \in S_{E,k}$. Then $\longrightarrow_{E,k}$ is the least subset of $S_{E,k} \times (\mathbb{R}_+ \cup Inf) \times S_{E,k}$ such that
  - If s is tangible, or vanishing but $s \notin \{s_i \mid 1 \leqslant i \leqslant n\}$, and $(\tilde{\lambda}, s') \in PM_k(s)$, then $s \xrightarrow{\tilde{\lambda}}_{E,k} s'$.

[2] With abuse of notation, we apply function $Melt$ to multisets of pairs whose first components are rates instead of actions.

  – If s is vanishing, $s \equiv s_i$ and in $PM_k(s)$ there are exactly $m \geqslant 1$ potential moves (∞_{l,p_j}, s_j), $1 \leqslant j \leqslant m$, such that $s_j \not\equiv s$, then there are m transitions $s \xrightarrow{\infty_{l,p_j/p}}_{E,k} s_j$, $1 \leqslant j \leqslant m$, where $p = \sum_{j=1}^{m} p_j$.

- $P_{E,k} : S_{E,k} \rightarrow \mathbb{R}_{[0,1]}$, $P_{E,k}(s) = \begin{cases} P_{E,k-1}(s) & \text{if } s \notin \{s_i \mid 1 \leqslant i \leqslant n\} \\ P_{E,k-1}(s) + P_{E,k-1}(s_0) \cdot p_i & \text{if } s \equiv s_i \end{cases}$.

Definition 4.21. The *Markovian semantics* of $E \in \mathscr{E}$ is the PLTS $\mathscr{M}[\![E]\!] = (S_{E,\mathscr{M}}, \mathbb{R}_+, \rightarrow_{E,\mathscr{M}}, P_{E,\mathscr{M}})$ obtained by applying the algorithm above.

Theorem 4.22. *Let $E \in \mathscr{E}$. If $\mathscr{I}[\![E]\!]$ has finitely many states, then the algorithm terminates and $\mathscr{M}[\![E]\!]$ has no immediate transitions, has finitely many states, and is unique.*

Proof. See [8]. □

We conclude by observing that the coexistence in EMPA of the probabilistic and the exponentially timed kernels allows *phase-type distributions* [42] to be modeled. This makes the limitation to exponential distributions less restrictive, as it becomes possible to describe or approximate distributions frequently occurring in practice. For more details, the reader is referred to [8].

4.6. The gluing role of passive actions: synchronization

The four kernels of EMPA share a common feature: the presence of passive actions. This is due to the synchronization discipline on action rates adopted in EMPA, which causes passive actions to act as a *glue* for the various kernels forming EMPA.

The main consequence of the synchronization discipline on action rates is that only *client-server communications* are directly expressible: the rate of the action resulting from a synchronization is determined by the rate of the only possible active action involved in the synchronization itself. This choice has been made due to its *simplicity*, since it avoids the need to define the rate of the action deriving from the synchronization of two active actions. Also, this choice has been made due to its *modularity*. When modeling an n-way synchronization, only the designer of the active component must know the rate of the synchronization, while the other $n-1$ designers can get rid of it by leaving it unspecified through passive actions. Furthermore, possible subsequent changes of the rate affect only one component.

As observed in Section 3, to compute correctly the rate of a synchronization according to the *bounded capacity assumption* [30], which states that the rate at which a term carries out an action cannot be increased in case of synchronization, a normalization is required that takes into account the number of alternative or independent passive actions that can be synchronized with the active action at hand.

Example 4.23. Consider a queueing system $M/M/2/2$ [34], i.e. a service center with two independent servers where the customer interarrival time is exponentially distributed with rate λ and the service time of each server is exponentially distributed with rate μ. This queueing system can be represented as follows:

$$QS_{M/M/2/2} \triangleq Arrivals \parallel_{\{a\}} Servers_2$$

$$Arrivals \triangleq \langle a, \lambda \rangle . Arrivals$$

$$Servers_2 \triangleq S \parallel_{\emptyset} S$$

$$S \triangleq \langle a, * \rangle . \langle s, \mu \rangle . S.$$

The normalization operates in such a way that in $\mathscr{I}[\![QS_{M/M/2/2}]\!]$ the two transitions leaving the initial state have rate $\lambda/2$, so the rate λ of the involved active component *Arrivals* is preserved. Such a normalization is completely *transparent to the designer* in EMPA and stochastically timed process algebras like PEPA [29] since it is embodied in the semantic rules. On the contrary, in the case of stochastically timed process algebras like MTIPP [25], where passive actions are simulated through exponentially timed actions with rate 1 and the rate of the synchronization of two actions is given by the product of their rates, no normalization is carried out because the bounded capacity assumption is not made. As a consequence, it is *responsibility of the designer* to define the rates of actions with type a in both terms S so that their sum is 1, otherwise the expected underlying HCTMC would not be obtained. The problem of the context-dependent meaning of the rates of MTIPP actions has been partially alleviated in IMTIPP [27] and PMTIPP [52] by means of the introduction of *unprioritized unweighted immediate actions* and a *probabilistic choice operator*, respectively, and completely solved in MLOTOS [26] by keeping *action execution separated from time passing*.

Despite the fact that client-server communications frequently occur in computing systems, it would be useful to be able to describe other kinds of communication, as recognized in [30]. Some of them can be described in other stochastically timed process algebras: for example, *flexible client-server communications* in MTIPP [25], where the service requirement is expressed by means of an action whose rate describes a scaling factor instead of a passive action, and *patient communications* in PEPA [29], where the two terms involved in a synchronization work together at the rate of the slowest one. Unlike patient communications, flexible client-server communications can be modeled *indirectly* in EMPA.

Example 4.24. Consider a queueing system $M/M/1/q$ [34] with scalable service rate, i.e. a service center with one server and a FIFO queue with $q - 1$ seats where the customer interarrival time is exponentially distributed with rate λ and the service time depends on the number of customers in the queue. This queueing system can be represented as follows:

$$SSRQS_{M/M/1/q} \triangleq Arrivals \parallel_{\{a\}} (Queue_0 \parallel_{\{d_h \mid 1 \leqslant h \leqslant q-1\}} Server)$$

$$Arrivals \triangleq \langle a, \lambda \rangle . Arrivals$$

$$Queue_0 \triangleq \langle a, * \rangle . Queue_1$$

$$Queue_h \triangleq \langle a, * \rangle . Queue_{h+1} + \langle d_h, * \rangle . Queue_{h-1}, \quad 0 < h < q-1$$

$$Queue_{q-1} \triangleq \langle d_{q-1}, * \rangle . Queue_{q-2}$$

$$Server \triangleq \langle d_1, \infty_{1,1} \rangle . Server_1 + \cdots + \langle d_{q-1}, \infty_{1,1} \rangle . Server_{q-1}$$

$$Server_h \triangleq \langle s, sf(h) \cdot \mu \rangle . Server, \quad 1 \leqslant h \leqslant q-1$$

where μ is the basic service rate and $sf : \mathbb{N}_+ \rightarrow \mathbb{R}_+$ describes the scaling factor.

Other kinds of communications are listed in [30]: *polite communications*, *impolite communications* and *timed synchronizations*. In [7] we have shown that all of them can be described *indirectly* with EMPA obtaining the expected underlying HCTMC. This means that the limitation to client-server synchronizations, introduced for the sake of simplicity, is not so restrictive from the modeling viewpoint as it might seem.

4.7. A summarizing example

We finally report an example that should demonstrate the usefulness of the four kernels embodied in EMPA from the standpoint of the modeling and analysis of concurrent systems.

The example examined below is the Dining Philosophers problem. Suppose we are given n philosophers P_i $(0 \leqslant i \leqslant n-1)$ sitting at a round table each with a plate in front, and n chopsticks C_i $(0 \leqslant i \leqslant n-1)$ each shared by two neighbor philosophers and used to get the rice at the center of the table. Let us denote by "$\_ +_n \_$" the sum modulo n, and let $think_i$ be the action type "P_i is thinking", pu_i $(pu_{i+_n 1})$ be "P_i picks up C_i $(C_{i+_n 1})$", eat_i be "P_i is eating", and pd_i $(pd_{i+_n 1})$ be "P_i puts down C_i $(C_{i+_n 1})$". The scenario can be described as follows:

$$DP_n \equiv (P_0 \parallel_\emptyset \ldots \parallel_\emptyset P_{n-1}) \parallel_{\{pu_i, pd_i \mid 0 \leqslant i \leqslant n-1\}} (C_0 \parallel_\emptyset \ldots \parallel_\emptyset C_{n-1})$$

$$P_i \triangleq \langle think_i, * \rangle . (\langle pu_i, * \rangle . \langle pu_{i+_n 1}, * \rangle . P_i' + \langle pu_{i+_n 1}, * \rangle . \langle pu_i, * \rangle . P_i')$$

$$P_i' \triangleq \langle eat_i, * \rangle . \langle pd_i, * \rangle . \langle pd_{i+_n 1}, * \rangle . P_i$$

$$C_i \triangleq \langle pu_i, * \rangle . \langle pd_i, * \rangle . C_i .$$

Since all the actions are passive, the system is purely nondeterministic: this is exactly the same description we would obtain with classical process algebras.

As a naive solution to break the symmetry that may cause deadlock, we could introduce a precedence relation among philosophers by means of the priority levels of immediate actions, thus modifying the specification of P_i as follows:

$$P_i \triangleq \langle think_i, *\rangle.\langle pu_i, \infty_{i+1,1}\rangle.\langle pu_{i+_n 1}, \infty_{i+1,1}\rangle.\langle eat_i, *\rangle.\langle pd_i, *\rangle.\langle pd_{i+_n 1}, *\rangle.P_i.$$

To solve the problem in a more elegant and fair manner, we could use the randomized distributed algorithm of [37]: P_i flips a fair coin to choose between C_i and C_{i+1}, gets the chosen chopstick as soon as it becomes free, and gets the other chopstick if it is free, otherwise releases the chosen chopstick and flips the coin again. This algorithm can be easily described in EMPA through the weights of immediate actions by modifying the specification of P_i as follows:

$$P_i \triangleq \langle think_i, *\rangle.P_i'$$

$$P_i' \triangleq \langle \tau, \infty_{1,1/2}\rangle.\langle pu_i, *\rangle.(\langle pu_{i+_n 1}, *\rangle.P_i'' + \langle pd_i, *\rangle.P_i')$$
$$+ \langle \tau, \infty_{1,1/2}\rangle.\langle pu_{i+_n 1}, *\rangle.(\langle pu_i, *\rangle.P_i'' + \langle pd_{i+_n 1}, *\rangle.P_i')$$

$$P_i'' \triangleq \langle eat_i, *\rangle.\langle pd_i, *\rangle.\langle pd_{i+_n 1}, *\rangle.P_i$$

Finally, by performance closing the system, with EMPA we can even assess some performance indices like e.g. the average time during which there is at least one philosopher eating, i.e. the chopstick utilization. The specification of P_i has to be modified as follows:

$$P_i \triangleq \langle think_i, \lambda_i\rangle.P_i'$$

$$P_i' \triangleq \langle \tau, \infty_{1,1/2}\rangle.\langle pu_i, \infty_{1,1}\rangle.(\langle pu_{i+_n 1}, \infty_{2,1}\rangle.P_i'' + \langle pd_i, \infty_{2,1}\rangle.P_i')$$
$$+ \langle \tau, \infty_{1,1/2}\rangle.\langle pu_{i+_n 1}, \infty_{1,1}\rangle.(\langle pu_i, \infty_{2,1}\rangle.P_i'' + \langle pd_{i+_n 1}, \infty_{2,1}\rangle.P_i')$$

$$P_i'' \triangleq \langle eat_i, \mu_i\rangle.\langle pd_i, \infty_{1,1}\rangle.\langle pd_{i+_n 1}, \infty_{1,1}\rangle.P_i$$

Observe that actions pu_i and pd_i have been modeled as immediate, because they are irrelevant from the performance evaluation point of view. Thus immediate actions provide a mechanism for *performance abstraction* in the same way as action type τ provides a mechanism for *functional abstraction*. Moreover, it is worth noting that priority levels of actions pu_i and pd_i have been fixed in such a way that, whenever the chopstick not initially chosen by a philosopher is free, the philosopher does pick up that chopstick instead of releasing the other one.

Other examples that highlight the expressive power of EMPA can be found in [8]: they are concerned with different kinds of queueing systems as well as the alternating bit protocol.

5. A notion of equivalence for EMPA

A notion of equivalence for EMPA should relate terms describing concurrent systems that are indistinguishable from the point of view of an external observer, i.e. having the *same functional and performance properties*. The purpose of this section is to develop such a notion of integrated equivalence as well as to make sure that it is a *congruence* in order to allow for compositional reasoning. The notion of integrated equivalence will be defined according to the *bisimulation* style [46, 40, 35]. The main motivation for resorting to this branching time semantics is that it will be possible to establish a clear connection between the equivalence itself and the notion of *ordinary lumping* [54] which is frequently used for aggregation purposes in performance evaluation. It is worth noting that the integrated equivalence allows for a *qualitative analysis*, namely by means of it we can investigate whether two terms represent two concurrent systems possessing the same functional and performance chracteristics regardless of their actual values. In order to carry out a *quantitative analysis*, i.e. to know whether a functional property holds, or to assess the value of a performance measure, we have to study the projected semantic models of (the simplest) one of the two terms.

The section is organized as follows. In Section 5.1 we introduce a notion of equivalence denoted $\sim_{FP}$ which is defined on the projected semantic models, and we show that it is not appropriate because it is not a congruence. In Section 5.2 we present a notion of equivalence denoted $\sim_{EMB}$ which is defined on the integrated semantic model by refining the idea of probabilistic bisimulation [35] according to the various kernels of EMPA singled out in Section 4. In Section 5.3 we prove that $\sim_{EMB}$ is a congruence, and in Section 5.4 we demonstrate that $\sim_{EMB}$ is the coarsest congruence contained in $\sim_{FP}$ as far as terms that cannot execute internal immediate actions are concerned. In Section 5.5 we give a sound and complete axiomatization of $\sim_{EMB}$ for the set of nonrecursive terms of EMPA. Finally, in Section 5.6 we develop an algorithm in the style of [16] that can be used both to check two EMPA terms for $\sim_{EMB}$ and to minimize the integrated semantic model of an EMPA term, and we show the relationship between $\sim_{EMB}$ and ordinary lumping. Proofs of results can be found in the appendix.

5.1. A deceptively integrated equivalence: $\sim_{FP}$

It is straightforward to define two projected equivalences on the two projected semantic models. For the functional semantic model we use *classical bisimilarity* [46, 40], whereas for the performance semantic model we use a variant of *probabilistic bisimilarity* [35] which takes into account initial state probabilities.

Definition 5.1. Let $E_1, E_2 \in \mathscr{G}$. We say that E_1 is *functionally equivalent* to E_2, written $E_1 \sim_F E_2$, if and only if $\mathscr{F}[\![E_1]\!] = (\uparrow E_1, AType, \longrightarrow_{E_1,\mathscr{F}}, E_1)$ is *bisimilar* to $\mathscr{F}[\![E_2]\!] = (\uparrow E_2, AType, \longrightarrow_{E_2,\mathscr{F}}, E_2)$, i.e. there exists a relation $\mathscr{B} \subseteq \uparrow E_1 \times \uparrow E_2$ such that:

- $(E_1, E_2) \in \mathscr{B}$;

- for each $(F_1, F_2) \in \mathscr{B}$ and for each $a \in AType$:
  - whenever $F_1 \xrightarrow{a}_{E_1,\mathscr{F}} F_1'$, then $F_2 \xrightarrow{a}_{E_2,\mathscr{F}} F_2'$ and $(F_1', F_2') \in \mathscr{B}$;
  - whenever $F_2 \xrightarrow{a}_{E_2,\mathscr{F}} F_2'$, then $F_1 \xrightarrow{a}_{E_1,\mathscr{F}} F_1'$ and $(F_1', F_2') \in \mathscr{B}$.

Definition 5.2. Let $E_1, E_2 \in \mathscr{E}$. We say that E_1 is *performance equivalent* to E_2, written $E_1 \sim_P E_2$, if and only if $\mathscr{M}[\![E_1]\!] = (S_{E_1}, \mathbb{R}_+, \longrightarrow_{E_1,\mathscr{M}}, P_{E_1})$ is *p-bisimilar* to $\mathscr{M}[\![E_2]\!] = (S_{E_2}, \mathbb{R}_+, \longrightarrow_{E_2,\mathscr{M}}, P_{E_2})$, i.e. there exists an equivalence relation $\mathscr{B} \subseteq (S_{E_1} \cup S_{E_2}) \times (S_{E_1} \cup S_{E_2})$ such that:
- for each $C \in (S_{E_1} \cup S_{E_2})/\mathscr{B}$, $\sum_{s \in C \cap S_{E_1}} P_{E_1}(s) = \sum_{s \in C \cap S_{E_2}} P_{E_2}(s)$;
- whenever $(s_1, s_2) \in \mathscr{B} \cap (S_{E_1} \times S_{E_2})$, then for each $C \in (S_{E_1} \cup S_{E_2})/\mathscr{B}$

$$\sum \{\!| \lambda \mid s_1 \xrightarrow{\lambda}_{E_1,\mathscr{M}} s_1' \wedge s_1' \in C \cap S_{E_1} |\!\} = \sum \{\!| \lambda \mid s_2 \xrightarrow{\lambda}_{E_2,\mathscr{M}} s_2' \wedge s_2' \in C \cap S_{E_2} |\!\}$$

A natural candidate notion of equivalence may be $\sim_{FP} = \sim_F \cap \sim_P$. However, the examples below show that $\sim_{FP}$ is not useful as it is not a congruence.

Example 5.3. Consider terms

$$E_1 \equiv \langle a, \lambda \rangle.\underline{0} + \langle b, \mu \rangle.\underline{0}$$
$$E_2 \equiv \langle a, \mu \rangle.\underline{0} + \langle b, \lambda \rangle.\underline{0}$$

where $\lambda \neq \mu$. It turns out that $E_1 \sim_{FP} E_2$ but $E_1 \parallel_{\{b\}} \underline{0} \not\sim_P E_2 \parallel_{\{b\}} \underline{0}$ because the left-hand side term can execute only one action with rate λ while the right-hand side term can execute only one action with rate μ. Note that the action with type a of E_1 has execution probability $\lambda/(\lambda + \mu)$, while the action with type a of E_2 has execution probability $\mu/(\lambda + \mu)$, and this is not detected by $\sim_{FP}$.

Example 5.4. Consider terms

$$E_1 \equiv \langle a, \infty_{1,1} \rangle.\underline{0}$$
$$E_2 \equiv \langle a, \infty_{2,1} \rangle.\underline{0}$$

It turns out that $E_1 \sim_{FP} E_2$ but $E_1 + \langle b, \infty_{1,1} \rangle.\underline{0} \not\sim_F E_2 + \langle b, \infty_{1,1} \rangle.\underline{0}$ because the left-hand side term can execute an action with type b while the right-hand side term cannot.

Example 5.5. Consider terms

$$E_1 \equiv \langle a, \infty_{1,1} \rangle.\underline{0}$$
$$E_2 \equiv \langle a, \infty_{1,2} \rangle.\underline{0}$$

It turns out that $E_1 \sim_{FP} E_2$ but $E_1 + \langle b, \infty_{1,1} \rangle.\langle b, \lambda \rangle.\underline{0} \not\sim_P E_2 + \langle b, \infty_{1,1} \rangle.\langle b, \lambda \rangle.\underline{0}$ because state $\langle b, \lambda \rangle.\underline{0}$ has initial state probability $\frac{1}{2}$ in the Markovian semantics of the left-hand side term, $\frac{1}{3}$ in the Markovian semantics of the right-hand side term.

The examples above show that $\sim_{FP}$ is unable to keep track of the link between the functional part and the performance part of the actions. This means that to achieve semantic compositionality, it is *necessary* to define an equivalence based on the integrated semantic model, and this will be stressed by Theorem 5.26. Incidentally, this is even *convenient* with respect to $\sim_{FP}$, since it avoids the need to build the two projected semantic models and checking them for bisimilarity and p-bisimilarity, respectively.

5.2. A really integrated equivalence: $\sim_{EMB}$

In order to define a really integrated equivalence in the bisimulation style, we have to consider the various kernels of EMPA:

- The exponentially timed kernel and the probabilistic kernel should be treated by following the notion of *probabilistic bisimulation* proposed in [35], which consists of requiring a bisimulation to be an equivalence relation such that two bisimilar terms have the *same aggregated probability* to reach the same equivalence class by executing actions of the same type and priority level.
  - In the case of the exponentially timed kernel, the notion of probabilistic bisimulation must be refined by requiring additionally that two bisimilar terms have *identically distributed sojourn times*. For example, if we consider terms $E_1 \equiv \langle a, \lambda\rangle.F + \langle a, \mu\rangle.G$ and $E_2 \equiv \langle a, 2\cdot\lambda\rangle.F + \langle a, 2\cdot\mu\rangle.G$ then both transitions labeled with a, λ and $a, 2\cdot\lambda$ have execution probability $\lambda/(\lambda+\mu)$, and both transitions labeled with a, μ and $a, 2\cdot\mu$ have execution probability $\mu/(\lambda+\mu)$, but the average sojourn time of E_1 is twice the average sojourn time of E_2. Due to the race policy, requiring that two bisimilar terms have identically distributed sojourn times and the same aggregated probability to reach the same equivalence class by executing exponentially timed actions of the same type, amounts to requiring that two bisimilar terms have the *same aggregated rate* to reach the same equivalence class by executing exponentially timed actions of the same type. For example, it must hold that

    $$\langle a, \lambda\rangle.F + \langle a, \mu\rangle.F \sim \langle a, \lambda+\mu\rangle.F$$

    This coincides with the notion of *Markovian bisimulation* proposed in [25, 29, 12].
  - In the case of the probabilistic kernel, the notion of probabilistic bisimulation must be restated in terms of weights. As a consequence, two bisimilar terms are required to have the *same aggregated weight* to reach the same equivalence class by executing immediate actions of the same type and priority level. For example, it must hold that

    $$\langle a, \infty_{l,w_1}\rangle.F + \langle a, \infty_{l,w_2}\rangle.F \sim \langle a, \infty_{l,w_1+w_2}\rangle.F$$

    This coincides with the notion of *direct bisimulation* proposed in [58].
- The nondeterministic kernel should be treated by following the classical notion of bisimulation [40]. Thus, bisimilar terms are required to have the same passive actions reaching the same equivalence class, *regardless of the actual number* of these passive actions. For example, it must hold that

  $$\langle a, *\rangle.F + \langle a, *\rangle.F \sim \langle a, *\rangle.F$$

- Concerning the prioritized kernel, it might seem useful to be able to write equations like

$$\langle c, \infty_{l,w}\rangle.E + \langle d, \infty_{l',w'}\rangle.F \sim \langle d, \infty_{l',w'}\rangle.F \qquad \text{if } l' > l$$
$$\langle a, \lambda\rangle.E + \langle b, \infty_{l,w}\rangle.F \sim \langle b, \infty_{l,w}\rangle.F$$

The problem is that the applicability of such equations depends on the context: e.g., terms $E_1 \equiv (\langle a, \lambda\rangle.E + \langle b, \infty_{l,w}\rangle.F) \parallel_{\{b\}} \underline{0}$ and $E_2 \equiv (\langle b, \infty_{l,w}\rangle.F) \parallel_{\{b\}} \underline{0}$ are not equivalent because E_1 can execute one action while E_2 cannot execute actions at all. To solve the problem, we follow the proposal of [4] by introducing a *priority operator* "$\Theta(\_)$": priority levels are taken to be potential, and they become effective only within the scope of the priority operator. We thus consider the language $\mathscr{L}_\Theta$ generated by the following syntax

$$E ::= \underline{0} \mid \langle a, \tilde{\lambda}\rangle.E \mid E/L \mid E[\varphi] \mid \Theta(E) \mid E + E \mid E \parallel_S E \mid A$$

whose semantic rules are those in Table 1 except that the rule in the first part is replaced by

$$\frac{(\langle a, \tilde{\lambda}\rangle, E') \in Melt(PM(E))}{E \xrightarrow{a, \tilde{\lambda}} E'}$$

and the following rule for the priority operator is introduced in the second part

$$PM(\Theta(E)) = Select(PM(E))$$

It is easily seen that EMPA coincides with the set of terms $\{\Theta(E) \mid E \in \mathscr{L}\}$.

All the conditions above that should be met in order for two terms to be considered equivalent can be subsumed by means of the following function expressing the *aggregated rate* with which a term can reach a class of terms by executing actions of a given type and priority level.

Definition 5.6. We define partial function *Rate*: $(\mathscr{G}_\Theta \times AType \times APLev \times \mathscr{P}(\mathscr{G}_\Theta)) \multimap ARate$ by[3]

$$Rate(E, a, l, C) = Min\{\!|\, \tilde{\lambda} \mid E \xrightarrow{a, \tilde{\lambda}} E' \wedge PL(\langle a, \tilde{\lambda}\rangle) = l \wedge E' \in C \,|\!\}$$

Definition 5.7. An equivalence relation $\mathscr{B} \subseteq \mathscr{G}_\Theta \times \mathscr{G}_\Theta$ is a *strong extended Markovian bisimulation* (*strong EMB*) if and only if, whenever $(E_1, E_2) \in \mathscr{B}$, then for all $a \in AType$, $l \in APLev$ and $C \in \mathscr{G}_\Theta/\mathscr{B}$

$$Rate(E_1, a, l, C) = Rate(E_2, a, l, C)$$

In this case we say that E_1 and E_2 are *strongly extended Markovian bisimilar* (*strongly EMB*).

[3] We let $Min\,\emptyset = \bot$, $\tilde{\lambda}\,Min\,\bot = \tilde{\lambda}$, $Split(\bot, p) = \bot$.

As an example, the identity relation $Id_{\mathscr{G}_\Theta}$ over $\mathscr{G}_\Theta$ is a strong EMB, and it is contained in any strong EMB due to reflexive property. We now prove that the largest strong EMB is the union of all the strong EMBs, and we define the integrated equivalence as the largest strong EMB.

Lemma 5.8. *Let $\{\mathscr{B}_i \mid i \in I\}$ be a family of strong EMBs. Then $\mathscr{B} = (\bigcup_{i \in I} \mathscr{B}_i)^+$ is a strong EMB.*

Proposition 5.9. *Let $\sim_{EMB}$ be the union of all the strong EMBs. Then $\sim_{EMB}$ is the largest strong EMB.*

Definition 5.10. We call $\sim_{EMB}$ the *strong extended Markovian bisimulation equivalence* (strong EMBE), and we say that $E_1, E_2 \in \mathscr{G}_\Theta$ are *strongly extended Markovian bisimulation equivalent* (strongly EMBE) if and only if $E_1 \sim_{EMB} E_2$.

In other words, two terms $E_1, E_2 \in \mathscr{G}_\Theta$ are strongly EMBE if and only if they are strongly EMB. It is worth noting that, despite the presence of several different kernels, we have been able to come up with a *compact and elegant* notion of equivalence in the style of [35].

We conclude the section by exhibiting two necessary conditions and one sufficient condition in order for two terms to be strongly EMBE. The necessary conditions below are based on aggregated rates independent of equivalence classes, so they are easily checkable. The first necessary condition guarantees that the states associated with strongly EMBE terms have identically distributed sojourn times, if tangible, or identical total weights, if vanishing. The second necessary condition is finer since it guarantees that strongly EMBE terms carry out actions of the same type and priority level at exactly the same aggregated rate: this will be used in Section 5.4 and recalled in Section 5.6.

Proposition 5.11. *Let $E_1, E_2 \in \mathscr{G}_\Theta$ such that $E_1 \sim_{EMB} E_2$.*

(i) *For all $l \in APLev$*

$$Min\{\!|Rate(E_1, a, l, \mathscr{G}_\Theta) \mid a \in AType|\!\} = Min\{\!|Rate(E_2, a, l, \mathscr{G}_\Theta) \mid a \in AType|\!\}$$

(ii) *For all $a \in AType$ and $l \in APLev$*

$$Rate(E_1, a, l, \mathscr{G}_\Theta) = Rate(E_2, a, l, \mathscr{G}_\Theta)$$

The sufficient condition below is based on the notion of strong EMB up to $\sim_{EMB}$, which is helpful to avoid redundancy in strong EMBs: for example, if $\mathscr{B}$ is a strong EMB and $(E_1 \parallel_S E_2, E_3) \in \mathscr{B}$, then also $(E_2 \parallel_S E_1, E_3) \in \mathscr{B}$ although it may be retrieved from the fact that $E_2 \parallel_S E_1 \sim_{EMB} E_1 \parallel_S E_2$. The sufficient condition states that in order for two terms $E_1, E_2 \in \mathscr{G}_\Theta$ to be strongly EMBE, it suffices to find out a strong EMB up to $\sim_{EMB}$ containing the pair (E_1, E_2). The notion of strong EMB up to $\sim_{EMB}$ will be used in Section 5.3.

Definition 5.12. An equivalence relation $\mathcal{B} \subseteq \mathcal{G}_\Theta \times \mathcal{G}_\Theta$ is a *strong EMB up to* $\sim_{EMB}$ if and only if, whenever $(E_1, E_2) \in \mathcal{B}$, then for all $a \in AType$, $l \in APLev$ and $C \in \mathcal{G}_\Theta / (\sim_{EMB} \mathcal{B} \sim_{EMB})$

$$Rate(E_1, a, l, C) = Rate(E_2, a, l, C)$$

Proposition 5.13. *If* $\mathcal{B} \subseteq \mathcal{G}_\Theta \times \mathcal{G}_\Theta$ *is a strong EMB up to* $\sim_{EMB}$ *and* $(E_1, E_2) \in \mathcal{B}$, *then* $E_1 \sim_{EMB} E_2$.

5.3. $\sim_{EMB}$ is a congruence

In this section we show that $\sim_{EMB}$ is preserved by all the operators of EMPA.

Theorem 5.14. *Let* $E_1, E_2 \in \mathcal{G}_\Theta$. *If* $E_1 \sim_{EMB} E_2$ *then*:
(i) *For every* $\langle a, \tilde{\lambda} \rangle \in Act$, $\langle a, \tilde{\lambda} \rangle . E_1 \sim_{EMB} \langle a, \tilde{\lambda} \rangle . E_2$.
(ii) *For every* $L \subseteq AType - \{\tau\}$, $E_1/L \sim_{EMB} E_2/L$.
(iii) *For every* $\varphi \in ARFun$, $E_1[\varphi] \sim_{EMB} E_2[\varphi]$.
(iv) $\Theta(E_1) \sim_{EMB} \Theta(E_2)$.
(v) *For every* $F \in \mathcal{G}_\Theta$, $E_1 + F \sim_{EMB} E_2 + F$ *and* $F + E_1 \sim_{EMB} F + E_2$.
(vi) *For every* $F \in \mathcal{G}_\Theta$ *and* $S \subseteq AType - \{\tau\}$, $E_1 \parallel_S F \sim_{EMB} E_2 \parallel_S F$ *and* $F \parallel_S E_1 \sim_{EMB} F \parallel_S E_2$.

In order to prove that $\sim_{EMB}$ is preserved by recursive definitions as well, we extend its definition to terms that are guardedly closed up to constants devoid of defining equation. Note that such constants act as variables.

Definition 5.15. A constant $A \in Const$ is *free w.r.t.* Def_Θ if and only if, for any $E \in \mathcal{L}_\Theta$, $A \triangleq E \notin Def_\Theta$.

Definition 5.16. A term $E \in \mathcal{L}_\Theta$ is *partially guardedly closed* (*pgc*) *w.r.t.* Def_Θ if and only if for each constant $A \in Const_{Def_\Theta}(E)$ either A is free w.r.t. Def_Θ or
- A is equipped in Def_Θ with defining equation $A \triangleq E'$, and
- there exists $F \in Subst_{Def_\Theta}(E')$ such that, whenever an instance of a constant B nonfree w.r.t. Def_Θ satisfies $B \, st \, F$, then the same instance satisfies $B \, st \, \langle a, \tilde{\lambda} \rangle . G \, st \, F$.

Definition 5.17. Let $E \in \mathcal{L}_\Theta$, $A \in Const$ free w.r.t. Def_Θ, and $B \in \mathcal{G}_\Theta$. The term $E\langle\langle A := B \rangle\rangle$ obtained from E by replacing each occurrence of A with B is defined by induction on the syntactical structure of E as follows:
- $\underline{0}\langle\langle A := B \rangle\rangle \equiv \underline{0}$
- $(\langle a, \tilde{\lambda} \rangle . E)\langle\langle A := B \rangle\rangle \equiv \langle a, \tilde{\lambda} \rangle . (E\langle\langle A := B \rangle\rangle)$
- $(E/L)\langle\langle A := B \rangle\rangle \equiv (E\langle\langle A := B \rangle\rangle)/L$
- $(E[\varphi])\langle\langle A := B \rangle\rangle \equiv (E\langle\langle A := B \rangle\rangle)[\varphi]$
- $\Theta(E)\langle\langle A := B \rangle\rangle \equiv \Theta(E\langle\langle A := B \rangle\rangle)$
- $(E_1 + E_2)\langle\langle A := B \rangle\rangle \equiv (E_1\langle\langle A := B \rangle\rangle) + (E_2\langle\langle A := B \rangle\rangle)$

- $(E_1 \parallel_S E_2)\langle\langle A := B\rangle\rangle \equiv (E_1\langle\langle A := B\rangle\rangle) \parallel_S (E_2\langle\langle A := B\rangle\rangle)$
- $A'\langle\langle A := B\rangle\rangle \equiv \begin{cases} B & \text{if } A' \equiv A \\ A' & \text{if } A' \not\equiv A \wedge A' \text{ free w.r.t. } \mathit{Def}_\Theta \\ A'' & \text{if } A' \not\equiv A \wedge A' \triangleq E \in \mathit{Def}_\Theta \wedge A'' \triangleq E\langle\langle A := B\rangle\rangle \in \mathit{Def}_\Theta \end{cases}$

Definition 5.18. Let $E_1, E_2 \in \mathscr{L}_\Theta$ be pgc, and suppose that $\mathit{Const}_{\mathit{Def}_\Theta}(E_1) \cup \mathit{Const}_{\mathit{Def}_\Theta}(E_2)$ contains $\{A_i \in \mathit{Const} \mid i \in I\}$ as free constants. We say that E_1 and E_2 are strongly EMBE if and only if, for all sets $\{B_i \in \mathscr{G}_\Theta \mid i \in I\}$ such that $E_1\langle\langle A_i := B_i\rangle\rangle_{i\in I}, E_2\langle\langle A_i := B_i\rangle\rangle_{i\in I} \in \mathscr{G}_\Theta$, it turns out that

$$E_1\langle\langle A_i := B_i\rangle\rangle_{i\in I} \sim_{EMB} E_2\langle\langle A_i := B_i\rangle\rangle_{i\in I}$$

Theorem 5.19. *Let $E_1, E_2 \in \mathscr{L}_\Theta$ be pgc, and suppose that $\mathit{Const}_{\mathit{Def}_\Theta}(E_1) \cup \mathit{Const}_{\mathit{Def}_\Theta}(E_2)$ contains only $A \in \mathit{Const}$ as a free constant. Let $A_1 \triangleq E_1\langle\langle A := A_1\rangle\rangle$, $A_2 \triangleq E_2\langle\langle A := A_2\rangle\rangle \in \mathit{Def}_\Theta$ with $A_1, A_2 \in \mathscr{G}_\Theta$. If $E_1 \sim_{EMB} E_2$, then $A_1 \sim_{EMB} A_2$.*

We conclude by reporting an example in which we exploit the congruence property.

Example 5.20. Consider a queueing system $M/M/n/n$ with arrival rate λ and service rate μ [34]. Such a queueing system represents a service center composed of n independent servers, such that the customer interarrival time is exponentially distributed with rate λ and the service time of each server is exponentially distributed with rate μ. The queueing system at hand can be given two different descriptions with EMPA: a *state-oriented description* where the focus is on the state of the set of servers (intended as the number of servers that are currently busy), and a *resource-oriented description* where the servers are modeled separately [59]. The state-oriented description is given by

$$\begin{aligned}
&\mathit{System}^{so}_{M/M/n/n} \triangleq \mathit{Arrivals} \parallel_{\{a\}} \mathit{Servers}_0 \\
&\mathit{Arrivals} \triangleq \langle a, \lambda\rangle.\mathit{Arrivals} \\
&\mathit{Servers}_0 \triangleq \langle a, *\rangle.\mathit{Servers}_1 \\
&\mathit{Servers}_h \triangleq \langle a, *\rangle.\mathit{Servers}_{h+1} + \langle s, h \cdot \mu\rangle.\mathit{Servers}_{h-1}, \quad 1 \leqslant h \leqslant n-1 \\
&\mathit{Servers}_n \triangleq \langle s, n \cdot \mu\rangle.\mathit{Servers}_{n-1}
\end{aligned}$$

whereas the resource-oriented description is given by

$$\begin{aligned}
&\mathit{System}^{ro}_{M/M/n/n} \triangleq \mathit{Arrivals} \parallel_{\{a\}} \mathit{Servers} \\
&\mathit{Arrivals} \triangleq \langle a, \lambda\rangle.\mathit{Arrivals} \\
&\mathit{Servers} \triangleq \underbrace{S \parallel_\emptyset S \parallel_\emptyset \ldots \parallel_\emptyset S}_{n} \\
&S \triangleq \langle a, *\rangle.\langle s, \mu\rangle.S
\end{aligned}$$

Since in these representations immediate actions do not occur, we have that $\Theta(\mathit{System}^{so}_{M/M/n/n}) \sim_{EMB} \mathit{System}^{so}_{M/M/n/n}$ and $\Theta(\mathit{System}^{ro}_{M/M/n/n}) \sim_{EMB} \mathit{System}^{ro}_{M/M/n/n}$. We now take advantage of the fact that $\sim_{EMB}$ is a congruence: to prove $System^{so}_{M/M/n/n} \sim_{EMB} System^{ro}_{M/M/n/n}$, it suffices to prove $Servers_0 \sim_{EMB} Servers$. This is the case because of the strong EMB up to $\sim_{EMB}$ given by the reflexive, symmetric and transitive closure of the relation made out of the following pairs of terms:

$$\begin{array}{ll} Servers_0, & S \,\|_{\emptyset}\, S \,\|_{\emptyset} \ldots \|_{\emptyset}\, S \\ Servers_1, & \langle s,\mu\rangle.S \,\|_{\emptyset}\, S \,\|_{\emptyset} \ldots \|_{\emptyset}\, S \\ Servers_2, & \langle s,\mu\rangle.S \,\|_{\emptyset}\, \langle s,\mu\rangle.S \,\|_{\emptyset} \ldots \|_{\emptyset}\, S \\ \ldots, & \ldots \\ Servers_n, & \langle s,\mu\rangle.S \,\|_{\emptyset}\, \langle s,\mu\rangle.S \,\|_{\emptyset} \ldots \|_{\emptyset}\, \langle s,\mu\rangle.S \end{array}$$

5.4. *Relationship between $\sim_{EMB}$ and $\sim_{FP}$*

In this section we investigate the relationship between $\sim_{EMB}$ and $\sim_{FP}$. Since $\sim_{EMB}$ is defined over EMPA$_{\Theta}$ terms whereas $\sim_{FP}$ is defined over EMPA terms, in this section we have to carefully introduce priority operators whenever necessary. The first result we prove is that the inclusion $\sim_{EMB} \subseteq \sim_{FP}$ holds in $\mathscr{E} \times \mathscr{E}$.

Theorem 5.21. *Let $E_1, E_2 \in \mathscr{G}$. If $E_1 \sim_{EMB} E_2$ then $E_1 \sim_F E_2$.*

Theorem 5.22. *Let $E_1, E_2 \in \mathscr{E}$. If $E_1 \sim_{EMB} E_2$ then $E_1 \sim_P E_2$.*

The inclusion $\sim_{EMB} \subsetneq \sim_{FP}$ in $\mathscr{E} \times \mathscr{E}$ is strict, as one can see by considering the examples below. Additionally, such examples show that $\sim_{EMB}$ cannot abstract from priority levels nor weights of immediate actions; otherwise, the congruence property would no longer hold.

Example 5.23. Consider terms E_1 and E_2 of Example 5.3. Then $E_1 \sim_{FP} E_2$ but $E_1 \not\sim_{EMB} E_2$ because $Rate(E_1, a, 0, \mathscr{G}_{\Theta}) = \lambda \neq \mu = Rate(E_2, a, 0, \mathscr{G}_{\Theta})$ thereby violating the necessary condition in Proposition 5.11(ii).

Example 5.24. Consider terms E_1 and E_2 of Example 5.4. Then $E_1 \sim_{FP} E_2$ but $E_1 \not\sim_{EMB} E_2$ because $Rate(E_1, a, 1, \mathscr{G}_{\Theta}) = \infty_{1,1} \neq \perp = Rate(E_2, a, 1, \mathscr{G}_{\Theta})$ thereby violating the necessary condition in Proposition 5.11(ii). Let $\sim_{EMB'}$ be the equivalence defined by relaxing Definition 5.7 to abstract from the priority level of immediate actions. Then $E_1 \sim_{EMB'} E_2$ but $\sim_{EMB'}$ would not be a congruence. For example, $\Theta(E_1 + \langle b, \infty_{1,1}\rangle.\underline{0}) \not\sim_{EMB'} \Theta(E_2 + \langle b, \infty_{1,1}\rangle.\underline{0})$ and $\Theta(E_1 \,\|_{\emptyset}\, \langle b, \infty_{1,1}\rangle.\underline{0}) \not\sim_{EMB'} \Theta(E_2 \,\|_{\emptyset}\, \langle b, \infty_{1,1}\rangle.\underline{0})$ because the left-hand side terms can execute an action with type b while the right-hand side terms cannot.

Example 5.25. Consider terms E_1 and E_2 of Example 5.5. Then $E_1 \sim_{FP} E_2$ but $E_1 \not\sim_{EMB} E_2$ because $Rate(E_1, a, 1, \mathscr{G}_{\Theta}) = \infty_{1,1} \neq \infty_{1,2} = Rate(E_2, a, 1, \mathscr{G}_{\Theta})$ thereby violating the necessary condition in Proposition 5.11(ii). Let $\sim_{EMB'}$ be the equivalence defined by relaxing Definition 5.7 to consider execution probabilities instead of weights for immediate actions (see the notion of *relative bisimulation* proposed in [58]). Then $E_1 \sim_{EMB'} E_2$ but $\sim_{EMB'}$ would not be a congruence. For example, $E_1 + \langle b, \infty_{1,1}\rangle.\underline{0} \not\sim_{EMB'} E_2 + \langle b, \infty_{1,1}\rangle.\underline{0}$ and $E_1 \parallel_{\emptyset} \langle b, \infty_{1,1}\rangle.\underline{0} \not\sim_{EMB'} E_2 \parallel_{\emptyset} \langle b, \infty_{1,1}\rangle.\underline{0}$ because the left-hand side terms can execute actions having type a with probability 1/2 while the right-hand side terms can execute actions having type a with probability $\frac{2}{3}$.

As a matter of fact, the second result we prove is that $\sim_{EMB}$, restricted to the set $\mathscr{E}_{-\tau\infty}$ of terms in $\mathscr{E}$ whose integrated semantic model does not contain internal immediate transitions, is the coarsest congruence contained in $\sim_{FP}$.

Theorem 5.26. *Let $E_1, E_2 \in \mathscr{E}_{-\tau\infty}$. Then $E_1 \sim_{EMB} E_2$ if and only if, for all $F \in \mathscr{G}$ and $S \subseteq AType - \{\tau\}$ such that $E_1 + F$, $E_2 + F$, $E_1 \parallel_S F$, $E_2 \parallel_S F \in \mathscr{E}_{-\tau\infty}$, it turns out that $E_1 + F \sim_{FP} E_2 + F$ and $E_1 \parallel_S F \sim_{FP} E_2 \parallel_S F$.*

In the following example we show the problems that arise when internal immediate transitions come into play.

Example 5.27. Consider the terms

$$E_1 \equiv \langle a, \infty_{1,1}\rangle.A, \qquad E_2 \equiv \langle a, \infty_{1,1}\rangle.B,$$

where

$$A \triangleq \langle \tau, \infty_{1,1}\rangle.A, \qquad B \triangleq \langle \tau, \infty_{1,2}\rangle.B.$$

It turns out that $E_1 \not\sim_{EMB} E_2$ because A and B violate the necessary condition expressed by Proposition 5.11(ii), and that E_1 and E_2 cannot be distinguished with respect to $\sim_{FP}$ by means of a context based on the alternative composition operator or the parallel composition operator.

In fact, the alternative composition operator does not allow us to introduce a choice at the level of A and B. The parallel composition operator in principle allows us to introduce a choice at the level of A and B. However, we also have to introduce an exponentially timed action, in such a way that the state having a transition labeled with this action has different initial state probabilities in the Markovian semantics of the two resulting terms (see Proof of Theorem 5.26). The problem is that such an action cannot be executed at all because A and B always have a higher priority level action ready to be executed, and this action cannot be blocked by means of an appropriate synchronization set as it is internal.

5.5. *Axiomatization of $\sim_{EMB}$*

Since we have proved that $\sim_{EMB}$ is a congruence, we now develop an equational theory for nonrecursive EMPA terms according to $\sim_{EMB}$. Such a theory is based on the set $\mathscr{A}$ of axioms in Table 6, and we denote by $Ded(\mathscr{A})$ the corresponding deductive

Table 6
Axioms for $\sim_{EMB}$

$$
\begin{array}{ll}
(\mathscr{A}_1) & (E_1 + E_2) + E_3 = E_1 + (E_2 + E_3) \\
(\mathscr{A}_2) & E_1 + E_2 = E_2 + E_1 \\
(\mathscr{A}_3) & E + \underline{0} = E \\
(\mathscr{A}_4) & \langle a, \tilde{\lambda}_1\rangle.E + \langle a, \tilde{\lambda}_2\rangle.E = \langle a, \tilde{\lambda}_1 \, Min \, \tilde{\lambda}_2\rangle.E \quad \text{if } PL(\langle a, \tilde{\lambda}_1\rangle) = PL(\langle a, \tilde{\lambda}_2\rangle) \\
(\mathscr{A}_5) & \underline{0}/L = \underline{0} \\
(\mathscr{A}_6) & (\langle a, \tilde{\lambda}\rangle.E)/L = \begin{cases} \langle a, \tilde{\lambda}\rangle.(E/L) & \text{if } a \notin L \\ \langle \tau, \tilde{\lambda}\rangle.(E/L) & \text{if } a \in L \end{cases} \\
(\mathscr{A}_7) & (E_1 + E_2)/L = E_1/L + E_2/L \\
(\mathscr{A}_8) & \underline{0}[\varphi] = \underline{0} \\
(\mathscr{A}_9) & (\langle a, \tilde{\lambda}\rangle.E)[\varphi] = \langle \varphi(a), \tilde{\lambda}\rangle.(E[\varphi]) \\
(\mathscr{A}_{10}) & (E_1 + E_2)[\varphi] = E_1[\varphi] + E_2[\varphi] \\
(\mathscr{A}_{11}) & \Theta(\underline{0}) = \underline{0} \\
(\mathscr{A}_{12}) & \Theta\left(\sum_{i\in I} \langle a_i, \tilde{\lambda}_i\rangle.E_i\right) = \sum_{j\in J} \langle a_j, \tilde{\lambda}_j\rangle.\Theta(E_j) \\
& \qquad \text{where } J = \{ i \in I \mid \tilde{\lambda}_i = * \vee \forall h \in I.\, PL(\langle a_i, \tilde{\lambda}_i\rangle) \geqslant PL(\langle a_h, \tilde{\lambda}_h\rangle)\} \\
(\mathscr{A}_{13}) & \left(\sum_{i\in I_1} \langle a_i, \tilde{\lambda}_i\rangle.E_i\right) \|_S \left(\sum_{i\in I_2} \langle a_i, \tilde{\lambda}_i\rangle.E_i\right) = \sum_{j\in J_1} \langle a_j, \tilde{\lambda}_j\rangle.(E_j \|_S \sum_{i\in I_2} \langle a_i, \tilde{\lambda}_i\rangle.E_i) \\
& \qquad + \sum_{j\in J_2} \langle a_j, \tilde{\lambda}_j\rangle.\left(\sum_{i\in I_1} \langle a_i, \tilde{\lambda}_i\rangle.E_i \|_S E_j\right) \\
& \qquad + \sum_{k\in K_1} \sum_{h\in H_k} \langle a_k, Split(\tilde{\lambda}_k, 1/n_k)\rangle.(E_k \|_S E_h) \\
& \qquad + \sum_{k\in K_2} \sum_{h\in H_k} \langle a_k, Split(\tilde{\lambda}_k, 1/n_k)\rangle.(E_h \|_S E_k) \\
& \qquad \text{where } J_1 = \{ i \in I_1 \mid a_i \notin S\} \\
& \qquad \phantom{\text{where }} J_2 = \{ i \in I_2 \mid a_i \notin S\} \\
& \qquad \phantom{\text{where }} K_1 = \{ k \in I_1 \mid \exists h \in I_2.\, a_h = a_k \in S \wedge \tilde{\lambda}_h = *\} \\
& \qquad \phantom{\text{where }} K_2 = \{ k \in I_2 \mid \exists h \in I_1.\, a_h = a_k \in S \wedge \tilde{\lambda}_h = *\} \\
& \qquad \phantom{\text{where }} H_k = \begin{cases} \{ h \in I_2 \mid a_h = a_k \wedge \tilde{\lambda}_h = *\} & \text{if } k \in K_1 \\ \{ h \in I_1 \mid a_h = a_k \wedge \tilde{\lambda}_h = *\} & \text{if } k \in K_2 \end{cases} \\
& \qquad \phantom{\text{where }} n_k = |H_k|
\end{array}
$$

system.[4] As it can be noted, the main difference with respect to the axiomatization of classical process algebras according to bisimulation equivalence lies in axiom $\mathscr{A}_4$: the idempotency property of the alternative composition operator holds only in the case of passive actions, and the fact that rates of exponentially timed actions are summed up is a consequence of the adoption of the race policy.

Now we prove that $Ded(\mathscr{A})$ is a *sound and complete* deductive system with respect to $\sim_{EMB}$ for the set $\mathscr{G}_{\Theta,nrec}$ of nonrecursive terms in $\mathscr{G}_\Theta$. To accomplish this, we introduce as usual the definition of normal form for a term, and then we prove that every term can be transformed into a term in normal form via $Ded(\mathscr{A})$.

[4] The reader is referred to [23] for notions and results concerning deductive systems for algebraic theories of concurrent processes.

Definition 5.28. $F \in \mathscr{G}_{\Theta,nrec}$ is in *sum normal form* (*snf*) if and only if $F \equiv \sum_{i \in I} \langle a_i, \tilde{\lambda}_i \rangle . F_i$ where every F_i is itself in snf, and we assume $F \equiv \underline{0}$ whenever $I = \emptyset$.

Definition 5.29. We define function *size* : $\mathscr{G}_{\Theta,nrec} \longrightarrow \mathbb{N}_+$ by structural induction as follows:
- $size(\underline{0}) = 1$;
- $size(\langle a, \tilde{\lambda} \rangle . E) = 1 + size(E)$;
- $size(E/L) = size(E[\varphi]) = size(\Theta(E)) = size(E)$;
- $size(E_1 + E_2) = \max(size(E_1), size(E_2))$;
- $size(E_1 \parallel_S E_2) = size(E_1) + size(E_2)$.

Lemma 5.30. *For any* $E \in \mathscr{G}_{\Theta,nrec}$, $size(E) \geqslant 1$.

Lemma 5.31. *For any axiom* $E_1 = E_2$ *in Table* 6, $size(E_1) \geqslant size(E_2)$.

Lemma 5.32. *For any* $E \in \mathscr{G}_{\Theta,nrec}$ *there exists* $F \in \mathscr{G}_{\Theta,nrec}$ *in snf such that* $\mathscr{A} \vdash E = F$.

Theorem 5.33. *The deductive system* $Ded(\mathscr{A})$ *is sound and complete with respect to* $\sim_{EMB}$ *for the set* $\mathscr{G}_{\Theta,nrec}$ *of nonrecursive terms.*

5.6. An algorithm to check for $\sim_{EMB}$

We conclude our study of a notion of equivalence for EMPA by presenting an algorithm that can be used to check whether two finite-state terms of $\mathscr{G}$ are strongly EMBE or not, and to minimize the integrated semantic model of a finite-state term with respect to $\sim_{EMB}$.

The algorithm, which is shown in Table 7, is an adaptation to our framework of the algorithm described in [16] (which could be applied to the functional semantic models of finite-state terms of $\mathscr{G}$), which is in turn a variant of the algorithm proposed in [45] to solve the relational coarsest partition problem. Given a labeled transition system with state space S representing the union of the integrated semantic models of two finite-state terms of $\mathscr{G}$ to be checked for $\sim_{EMB}$, or the integrated semantic model of a finite-state term of $\mathscr{G}$ to be minimized with respect to $\sim_{EMB}$, the idea of the algorithm is to repeatedly refine the current partition until this is a strong EMB. As we can see, the initial partition contains only one class which is the entire state space, and the partition resulting after the first execution of the repeat-until cycle is the coarsest partition satisfying the necessary condition for $\sim_{EMB}$ of Proposition 5.11(ii). By following the proposal of [45], this algorithm can be implemented in $O(m \log n)$ time and $O(m + n)$ space where n is the number of states and m is the number of transitions.

It is worth noting that a variant of the algorithm in Table 7 can be used to compute the coarsest ordinary lumping [54] of the Markovian semantics of a given term, hence allowing for the determination of performance measures by solving a smaller Markov chain which is equivalent to the original one.

Table 7
Algorithm to check for $\sim_{EMB}$

```
begin
  Partition := {S};
  Splitters := {S};
  repeat
    OldPartition := Partition;
    choose C' in Splitters;
    Splitters := Splitters − {C'};
    for each a ∈ AType do
      for each l ∈ APLev do
        for each C ∈ OldPartition do begin
          ⟨let Partition_C be the coarsest partition of C such that for every E1, E2 ∈ C'' ∈ Partition_C
                              Rate(E1, a, l, C') = Rate(E2, a, l, C')⟩;
          if Partition_C ≠ {C} then begin
            Partition := Partition − {C} ∪ Partition_C;
            Splitters := Splitters − {C} ∪ Partition_C
          end
        end
  until Splitters = ∅
end
```

Definition 5.34. Let $M_i = (S_i, \mathbb{R}_+, \longrightarrow_i, P_i)$, $i \in \{1,2\}$, be two PLTSs representing two HCTMCs. We say that M_2 is an *ordinary lumping* of M_1 if and only if S_2 is a partition of S_1 such that $C \xrightarrow{\lambda}_2 C'$ whenever for every $s \in C$ it holds $\lambda = \sum \{\!|\mu \mid s \xrightarrow{\mu}_1 s' \wedge s' \in C'|\!\}$.

Proposition 5.35. *Let $M_i = (S_i, \mathbb{R}_+, \longrightarrow_i, P_i)$, $i \in \{1,2\}$, be two PLTSs representing two HCTMCs. If M_2 is an ordinary lumping of M_1, then M_2 is p-bisimilar to M_1.*

We conclude by giving the definition of lumped Markovian semantics and by showing the relation among $\sim_{EMB}$, p-bisimilarity and ordinary lumpability.

Definition 5.36. Let $\mathcal{M}[\![E]\!] = (S_{E,\mathcal{M}}, \mathbb{R}_+, \longrightarrow_{E,\mathcal{M}}, P_{E,\mathcal{M}})$ be the Markovian semantics of $E \in \mathscr{E}$. The *lumped Markovian semantics* of E is the PLTS $\mathcal{M}_l[\![E]\!] = (S_{E,\mathcal{M}_l}, \mathbb{R}_+, \longrightarrow_{E,\mathcal{M}_l}, P_{E,\mathcal{M}_l})$ where:

- $S_{E,\mathcal{M}_l} = \mathit{Partition}$ where *Partition* is the result of the algorithm in Table 8;
- $\longrightarrow_{E,\mathcal{M}_l}$ is the least subset of $S_{E,\mathcal{M}_l} \times \mathbb{R}_+ \times S_{E,\mathcal{M}_l}$ such that $C \xrightarrow{\lambda}_{E,\mathcal{M}_l} C'$ whenever

  $$\lambda = \sum \{\!|\mu \mid s \xrightarrow{\mu}_{E,\mathcal{M}} s' \wedge s' \in C'|\!\}$$

  with s fixed in C;
- $P_{E,\mathcal{M}_l} : S_{E,\mathcal{M}_l} \longrightarrow \mathbb{R}_{[0,1]}$, $P_{E,\mathcal{M}_l}(C) = \sum_{s \in C} P_{E,\mathcal{M}}(s)$.

Theorem 5.37. *Let $E_1, E_2 \in \mathscr{E}$. If $E_1 \sim_{EMB} E_2$ then $\mathcal{M}_l[\![E_1]\!]$ is p-bisimilar to $\mathcal{M}_l[\![E_2]\!]$.*

Corollary 5.38. *Let $E_1, E_2 \in \mathscr{E}$. If $E_1 \sim_{EMB} E_2$ then $\mathcal{M}_l[\![E_1]\!]$ is p-isomorphic to $\mathcal{M}_l[\![E_2]\!]$, i.e. there exists a bijection $\beta : S_{E_1,\mathcal{M}_l} \longrightarrow S_{E_2,\mathcal{M}_l}$ such that*

- $P_{E_1,\mathcal{M}_l}(s) = P_{E_2,\mathcal{M}_l}(\beta(s))$ *for any* $s \in S_{E_1,\mathcal{M}_l}$;
- $s \xrightarrow{\lambda}_{E_1,\mathcal{M}_l} s' \Longleftrightarrow \beta(s) \xrightarrow{\lambda}_{E_2,\mathcal{M}_l} \beta(s')$ *for any* $s \in S_{E_1,\mathcal{M}_l}$.

Table 8
Algorithm to compute the coarsest ordinary lumping

```
begin
  Partition := {S_{E,M}};
  Splitters := {S_{E,M}};
  repeat
    OldPartition := Partition;
    choose C' in Splitters;
    Splitters := Splitters − {C'};
    for each C ∈ OldPartition do begin
      ⟨let Partition_C be the coarsest partition of C such that for every s1,s2 ∈ C'' ∈ Partition_C
              Σ{|λ | s1 -λ->_{E,M} s ∧ s ∈ C'|} = Σ{|λ | s2 -λ->_{E,M} s ∧ s ∈ C'|}⟩;
      if Partition_C ≠ {C} then begin
        Partition := Partition − {C} ∪ Partition_C;
        Splitters := Splitters − {C} ∪ Partition_C
      end
    end
  until Splitters = ∅
end
```

The corollary above reveals the adequacy of $\sim_{EMB}$ from the performance standpoint: if $\Theta(E_1) \sim_{EMB} \Theta(E_2)$, then their underlying lumped Markovian models have the *same transient and steady-state (if any) probability distributions*, i.e. they describe two concurrent systems having the same performance characteristics.

6. Conclusions

In this tutorial we have shown that it is possible to develop a process algebra allowing for nondeterminism, priorities, probabilities and time without burdening the underlying theory exceedingly. In particular, we have shown that the expressiveness of EMPA is considerable, because it can be viewed as the sum of the expressiveness of a classical process algebra, a prioritized process algebra, a probabilistic process algebra and an exponentially timed process algebra, while the underlying theory is relatively simple, since the idea of potential move used in the definition of the integrated interleaving semantics is intuitive, and the notion of integrated equivalence is compact as well as elegant.

The problem we are currently investigating is how to further enhance the expressive power of EMPA. This means extending the timed kernel in order to *directly* cope with *durations following arbitrary distributions*, hence avoiding the need to resort to approximations obtained by means of the interplay of exponentially timed and immediate actions. To achieve this, we have to understand which information must be added to states and transitions of the integrated semantics, how to define the performance semantics, and how to extend the notion of integrated equivalence. We are afraid that the enhanced expressiveness obtained by means of general distributions cannot be traded with the complexity of the underlying theory. However, this is extremely challenging because it should permit to bridge the *gap between deterministically timed process algebras and stochastically timed process algebras.*

Acknowledgements

We are grateful to Pedro D'Argenio for his remarks about the modeling of Dining Philosophers.

This research has been partially funded by MURST, CNR and Esprit Project LOMAPS n. 8130.

Appendix A. Proofs of properties of $\sim_{EMB}$

Proof of Lemma 5.8. Once observed that $\mathcal{B}$ is an equivalence relation because it is the transitive closure of the union of equivalence relations, assume that $(E_1,E_2)\in\mathcal{B}$. Since $\mathcal{B}=\bigcup_{n\in\mathbb{N}_+}\mathcal{B}^{(n)}$ where $\mathcal{B}^{(n)}=(\bigcup_{i\in I}\mathcal{B}_i)^n$, we have $(E_1,E_2)\in\mathcal{B}^{(n)}$ for some $n\in\mathbb{N}_+$. The result follows by proving by induction on $n\in\mathbb{N}_+$ that, whenever $(E_1,E_2)\in\mathcal{B}^{(n)}$, then $Rate(E_1,a,l,C)=Rate(E_2,a,l,C)$ for all $a\in AType$, $l\in APLev$, $C\in\mathcal{G}_\Theta/\mathcal{B}$.

- If $n=1$, then $(E_1,E_2)\in\mathcal{B}_i$ for some $i\in I$. Let $\mathcal{G}_\Theta/\mathcal{B}_i=\{C_{i,j}\mid j\in J_i\}$. Since $(E_1,E_2)\in\mathcal{B}_i$ implies $(E_1,E_2)\in\mathcal{B}$, we have that for each $C_{i,j}\in\mathcal{G}_\Theta/\mathcal{B}_i$ there exists $C\in\mathcal{G}_\Theta/\mathcal{B}$ such that $C_{i,j}\subseteq C$, so each equivalence class of $\mathcal{B}$ can be written as the union of a set of equivalence classes of $\mathcal{B}_i$. As a consequence, for all $a\in AType$, $l\in APLev$ and $C\in\mathcal{G}_\Theta/\mathcal{B}$, if $C=\bigcup_{j\in J_i'}C_{i,j}$ where $J_i'\subseteq J_i$, then $Rate(E_1,a,l,C)=Min\{\!|Rate(E_1,a,l,C_{i,j})\mid j\in J_i'|\!\}=Min\{\!|Rate(E_2,a,l,C_{i,j})\mid j\in J_i'|\!\}=Rate(E_2,a,l,C)$ because $\mathcal{B}_i$ is a strong EMB.
- Let $n>1$. From $(E_1,E_2)\in\mathcal{B}^{(n)}$ we derive that there exists $F\in\mathcal{G}_\Theta$ such that $(E_1,F)\in\mathcal{B}^{(n-1)}$ and $(F,E_2)\in\mathcal{B}_i$ for some $i\in I$. Thus for all $a\in AType$, $l\in APLev$ and $C\in\mathcal{G}_\Theta/\mathcal{B}$, it turns out $Rate(E_1,a,l,C)=Rate(F,a,l,C)$ by the induction hypothesis, and $Rate(F,a,l,C)=Rate(E_2,a,l,C)$ by applying the same argument as the previous point. □

Proof of Proposition 5.9. By definition, $\sim_{EMB}$ contains the largest strong EMB. If we prove that $\sim_{EMB}$ is a strong EMB, then we are done. Since $\sim_{EMB}\subseteq\sim_{EMB}^+$ trivially holds, and $\sim_{EMB}^+\subseteq\sim_{EMB}$ is due to the fact that $\sim_{EMB}^+$ is a strong EMB by virtue of Lemma 5.8 and that $\sim_{EMB}$ contains all the strong EMBs by definition, we have $\sim_{EMB}=\sim_{EMB}^+$. Again $\sim_{EMB}^+$ is a strong EMB because of Lemma 5.8, hence so is $\sim_{EMB}$. □

Proof of Proposition 5.13. Given $\mathcal{B}\subseteq\mathcal{G}_\Theta\times\mathcal{G}_\Theta$ strong EMB up to $\sim_{EMB}$, we first prove that $\sim_{EMB}\mathcal{B}\sim_{EMB}$ is a strong EMB. Let $(E_1,E_2)\in\sim_{EMB}\mathcal{B}\sim_{EMB}$, i.e. $E_1\sim_{EMB}E_1'\,\mathcal{B}\,E_2'\sim_{EMB}E_2$, and $a\in AType$, $l\in APLev$ and $C\in\mathcal{G}_\Theta/(\sim_{EMB}\mathcal{B}\sim_{EMB})$. Since $\sim_{EMB}\subseteq\sim_{EMB}\mathcal{B}\sim_{EMB}$ in that $F_1\sim_{EMB}F_2$ implies $F_1\sim_{EMB}F_2\,\mathcal{B}\,F_2\sim_{EMB}F_2$ for any F_1, $F_2\in\mathcal{G}_\Theta$, C is the union of some equivalence classes with respect to

$\sim_{EMB}$. As a consequence, for $j \in \{1,2\}$ we have $Rate(E_j, a, l, C) = Rate(E'_j, a, l, C)$ because $E_j \sim_{EMB} E'_j$. Since $Rate(E'_1, a, l, C) = Rate(E'_2, a, l, C)$ in that $(E'_1, E'_2) \in \mathscr{B}$, the result follows. To complete the proof, we observe that $\mathscr{B} \subseteq \sim_{EMB} \mathscr{B} \sim_{EMB}$ because $Id_{\mathscr{G}_\Theta} \subseteq \sim_{EMB}$, and $\sim_{EMB} \mathscr{B} \sim_{EMB} \subseteq \sim_{EMB}$ because $\sim_{EMB} \mathscr{B} \sim_{EMB}$ is a strong EMB, hence $\mathscr{B} \subseteq \sim_{EMB}$ by transitivity.

Proof of Theorem 5.14. Let $E_1, E_2 \in \mathscr{G}_\Theta$ be such that $E_1 \sim_{EMB} E_2$.

(i) Let $\mathscr{B} \subseteq \mathscr{G}_\Theta \times \mathscr{G}_\Theta$ be a strong EMB such that $(E_1, E_2) \in \mathscr{B}$. Given $\langle a, \tilde{\lambda} \rangle \in Act$, we prove that

$$\mathscr{B}' = (\mathscr{B} \cup \{(\langle a, \tilde{\lambda} \rangle.E_1, \langle a, \tilde{\lambda} \rangle.E_2), (\langle a, \tilde{\lambda} \rangle.E_2, \langle a, \tilde{\lambda} \rangle.E_1)\})^+$$

is a strong EMB. Observed that $\mathscr{B}'$ is an equivalence relation, we have two cases.

– If $(\langle a, \tilde{\lambda} \rangle.E_1, \langle a, \tilde{\lambda} \rangle.E_2) \in \mathscr{B}$, then $\mathscr{B}' = \mathscr{B}$ and the result trivially follows.

– Assume that $(\langle a, \tilde{\lambda} \rangle.E_1, \langle a, \tilde{\lambda} \rangle.E_2) \notin \mathscr{B}$. Observed that

$$\begin{aligned}\mathscr{G}_\Theta/\mathscr{B}' = {} & (\mathscr{G}_\Theta/\mathscr{B} - \{[\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}}, [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}}\}) \\ & \cup \{[\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}} \cup [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}}\}\end{aligned}$$

let $(F_1, F_2) \in \mathscr{B}'$ and $b \in AType$, $l \in APLev$, $C \in \mathscr{G}_\Theta/\mathscr{B}'$.

- If $(F_1, F_2) \in \mathscr{B}$ and $C \in \mathscr{G}_\Theta/\mathscr{B} - \{[\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}}, [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}}\}$, then trivially $Rate(F_1, b, l, C) = Rate(F_2, b, l, C)$.
- If $(F_1, F_2) \in \mathscr{B}$ and $C = [\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}} \cup [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}}$, then for $j \in \{1,2\}$ we have

$$Rate(F_j, b, l, C) = Rate(F_j, b, l, [\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}})\, Min\, Rate(F_j, b, l, [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}})$$

so $Rate(F_1, b, l, C) = Rate(F_2, b, l, C)$.
- If $(F_1, F_2) \in \mathscr{B}' - \mathscr{B}$, i.e. $F_1 \in [\langle a, \tilde{\lambda} \rangle.E_1]_{\mathscr{B}}$ and $F_2 \in [\langle a, \tilde{\lambda} \rangle.E_2]_{\mathscr{B}}$, then for $j \in \{1,2\}$ we have

$$Rate(F_j, b, l, C) = \begin{cases} \tilde{\lambda} & \text{if } b = a \wedge l = PL(\langle a, \tilde{\lambda} \rangle) \wedge C = [E_j]_{\mathscr{B}'}, \\ \perp & \text{otherwise.} \end{cases}$$

Since $[E_1]_{\mathscr{B}'} = [E_1]_{\mathscr{B}} = [E_2]_{\mathscr{B}} = [E_2]_{\mathscr{B}'}$, it turns out that $Rate(F_1, b, l, C) = Rate(F_2, b, l, C)$.

(ii) Given $L \subseteq AType - \{\tau\}$, we prove that

$$\mathscr{B}' = \mathscr{B} \cup Id_{\mathscr{G}_\Theta}, \quad \text{where} \quad \mathscr{B} = \{(E_1/L, E_2/L) \mid E_1 \sim_{EMB} E_2\},$$

is a strong EMB. Observed that $\mathscr{B}'$ is an equivalence relation, and that either each of the terms of an equivalence class has "$\_/L$" as outermost operator or none of them has, let $(F_1, F_2) \in \mathscr{B}'$ and $a \in AType$, $l \in APLev$, $C \in \mathscr{G}_\Theta/\mathscr{B}'$.

– If $(F_1, F_2) \in Id_{\mathscr{G}_\Theta}$, then trivially $Rate(F_1, a, l, C) = Rate(F_2, a, l, C)$.

– If $(F_1, F_2) \in \mathscr{B}$, then $F_1 \equiv E_1/L$ and $F_2 \equiv E_2/L$ where $E_1 \sim_{EMB} E_2$.

- If none of the terms in C has "$\_/L$" as outermost operator, then trivially $Rate(F_1, a, l, C) = \perp = Rate(F_2, a, l, C)$.

- If each of the terms in C has "$\_/L$" as outermost operator, given $E/L \in C$ it turns out that $C = \{E'/L \mid E' \in [E]_{\sim_{EMB}}\}$. Thus for $j \in \{1,2\}$ we have

$$Rate(F_j, a, l, C) = \begin{cases} Rate(E_j, a, l, [E]_{\sim_{EMB}}) \\ Rate(E_j, \tau, l, [E]_{\sim_{EMB}})\, Min \\ \quad Min\{\!|Rate(E_j, b, l, [E]_{\sim_{EMB}}) \mid b \in L|\!\} \end{cases}$$

depending on whether $a \notin L \cup \{\tau\}$ or $a = \tau$. From $E_1 \sim_{EMB} E_2$ it follows $Rate(F_1, a, l, C) = Rate(F_2, a, l, C)$.

(iii) Given $\varphi \in ARFun$, the proof that

$$\mathscr{B}' = \mathscr{B} \cup Id_{\mathscr{G}_\Theta}, \quad \text{where} \quad \mathscr{B} = \{(E_1[\varphi], E_2[\varphi]) \mid E_1 \sim_{EMB} E_2\},$$

is a strong EMB is similar to the one developed in (ii). The main difference is that in the last subcase the result follows from the fact that for $j \in \{1,2\}$ we have

$$Rate(F_j, a, l, C) = Min\{\!|Rate(E_j, b, l, [E]_{\sim_{EMB}}) \mid \varphi(b) = a|\!\}$$

(iv) The proof that

$$\mathscr{B}' = \mathscr{B} \cup Id_{\mathscr{G}_\Theta}, \quad \text{where} \quad \mathscr{B} = \{(\Theta(E_1), \Theta(E_2)) \mid E_1 \sim_{EMB} E_2\},$$

is a strong EMB is similar to the one developed in (ii). The main difference is that in the last subcase the result follows from the fact that for $j \in \{1,2\}$ we have

$$Rate(F_j, a, l, C) = \begin{cases} Rate(E_j, a, l, [E]_{\sim_{EMB}}) & \text{if } \neg(E_j \xrightarrow{b, \tilde{\lambda}} F \wedge PL(\langle b, \tilde{\lambda}\rangle) > l) \\ & \quad \vee (l = -1) \\ \perp & \text{otherwise} \end{cases}$$

(v) Let $\mathscr{B} \subseteq \mathscr{G}_\Theta \times \mathscr{G}_\Theta$ be a strong EMB such that $(E_1, E_2) \in \mathscr{B}$. Given $F \in \mathscr{G}_\Theta$, the proof that

$$\mathscr{B}' = (\mathscr{B} \cup \{(E_1 + F, E_2 + F), (E_2 + F, E_1 + F)\})^+$$

is a strong EMB is similar to the one developed in (i). The main difference is that in the last subcase the result follows from the fact that for $j \in \{1,2\}$ we have

$$Rate(F_j, b, l, C) = Rate(E_j, b, l, C)\, Min\, Rate(F, b, l, C)$$

and from the following considerations:
- If $C \in \mathscr{G}_\Theta/\mathscr{B} - \{[E_1 + F]_{\mathscr{B}}, [E_2 + F]_{\mathscr{B}}\}$, then from $(E_1, E_2) \in \mathscr{B}$ we derive $Rate(E_1,$
  $b, l, C) = Rate(E_2, b, l, C)$ so $Rate(F_1, b, l, C) = Rate(F_2, b, l, C)$.
- If $C = [E_1 + F]_{\mathscr{B}} \cup [E_2 + F]_{\mathscr{B}}$, then for $j \in \{1,2\}$ we have

$$Rate(E_j, b, l, C) = Rate(E_j, b, l, [E_1 + F]_{\mathscr{B}})\, Min\, Rate(E_j, b, l, [E_2 + F]_{\mathscr{B}})$$

Since $(E_1, E_2) \in \mathscr{B}$, it turns out that $Rate(E_1, b, l, C) = Rate(E_2, b, l, C)$ so $Rate$ $(F_1, b, l, C) = Rate(F_2, b, l, C)$.

(vi) Given $S \subseteq AType - \{\tau\}$, the proof that

$$\mathscr{B}' = \mathscr{B} \cup Id_{\mathscr{G}_\Theta}, \quad \text{where} \quad \mathscr{B} = \{(E_1 \parallel_S F, E_2 \parallel_S F) \mid E_1 \sim_{EMB} E_2 \wedge F \in \mathscr{G}_\Theta\},$$

is a strong EMB is similar to the one developed in (ii). The main difference is that in the last subcase, where given $E \parallel_S G \in C$ it turns out that $C = \{E' \parallel_S G | E' \in [E]_{\sim_{EMB}}\}$, the result follows from the considerations below:

– If $a \notin S$, then for $j \in \{1,2\}$ we have that

$$Rate(F_j, a, l, C) = \begin{cases} Rate(E_j, a, l, [E]_{\sim_{EMB}}) \, Min \, Rate(F, a, l, \{G\}) \\ Rate(E_j, a, l, [E]_{\sim_{EMB}}) \\ Rate(F, a, l, \{G\}) \\ \perp \end{cases}$$

depending on whether $E_j \in [E]_{\sim_{EMB}} \wedge F \equiv G$, $E_j \notin [E]_{\sim_{EMB}} \wedge F \equiv G$, $E_j \in [E]_{\sim_{EMB}} \wedge F \not\equiv G$ or $E_j \notin [E]_{\sim_{EMB}} \wedge F \not\equiv G$. Since $E_1 \sim_{EMB} E_2$, it follows that $Rate(F_1, a, l, C) = Rate(F_2, a, l, C)$.

– If $a \in S$, then for $j \in \{1,2\}$ we have that

$$Rate(F_j, a, l, C) = \begin{cases} Rate(E_j, a, l, [E]_{\sim_{EMB}}) \, Min \, Rate(F, a, l, \{G\}) \\ Rate(E_j, a, l, [E]_{\sim_{EMB}}) \\ Rate(F, a, l, \{G\}) \\ \perp \end{cases}$$

depending on whether $Rate(F, a, -1, \{G\}) \neq \perp \wedge Rate(E_j, a, -1, [E]_{\sim_{EMB}}) \neq \perp$, $Rate(F, a, -1, \{G\}) \neq \perp \wedge Rate(E_j, a, -1, [E]_{\sim_{EMB}}) = \perp$, $Rate(F, a, -1, \{G\}) = \perp \wedge Rate(E_j, a, -1, [E]_{\sim_{EMB}}) \neq \perp$ or $Rate(F, a, -1, \{G\}) = \perp \wedge Rate(E_j, a, -1, [E]_{\sim_{EMB}}) = \perp$. Since $E_1 \sim_{EMB} E_2$, it follows that $Rate(F_1, a, l, C) = Rate(F_2, a, l, C)$. □

Proof of Theorem 5.19. It suffices to prove that $\mathscr{B}' = \mathscr{B} \cup \mathscr{B}^{-1}$ where

$$\mathscr{B} = \{(F_1, F_2) \mid F_1 \equiv F\langle\langle B := A_1 \rangle\rangle \wedge F_2 \equiv F\langle\langle B := A_2 \rangle\rangle \wedge F \in \mathscr{L}_\Theta \text{ pgc with at most } B \in Const_{Def_\Theta}(F) \text{ free}\}$$

is a strong EMB up to $\sim_{EMB}$: the result will follow by taking $F \equiv B$. We first observe that $\mathscr{B}' \subseteq \mathscr{G}_\Theta \times \mathscr{G}_\Theta$ is reflexive (because if F does not contain free variables then $F \in \mathscr{G}_\Theta$ and $F_1 \equiv F \equiv F_2$), symmetric (by definition), and transitive (for any $F \in \mathscr{L}_\Theta$ pgc with at most $B \in Const_{Def_\Theta}(F)$ free we have $(F_1, F_2) \in \mathscr{B}'$ and $(F_2, F_1) \in \mathscr{B}'$, and transitivity is guaranteed by $F_1, F_2 \in \mathscr{G}_\Theta$ and reflexivity). Given $(F_1, F_2) \in \mathscr{B}'$, $a \in AType$, $l \in APLev$, and $C \in \mathscr{G}_\Theta/(\sim_{EMB} \mathscr{B}' \sim_{EMB})$, we prove that $Rate(F_1, a, l, C) = Rate(F_2, a, l, C)$ by proceeding by induction on the maximum depth d of the inference of a potential move for F_1 having type a, priority level l, and derivative term in C.

- If $d=1$, then only the rule for the prefix operator has been used to deduce the potential move. Therefore $F \equiv \langle a,\tilde{\lambda}\rangle.F'$ with $PL(\langle a,\tilde{\lambda}\rangle)=l$, and for $j\in\{1,2\}$ we have $F_j \equiv \langle a,\tilde{\lambda}\rangle.(F'\langle\langle B:=A_j\rangle\rangle)$. Since $(F'\langle\langle B:=A_1\rangle\rangle, F'\langle\langle B:=A_2\rangle\rangle)\in\mathscr{B}$, it turns out that $C=[F'\langle\langle B:=A_1\rangle\rangle]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}=[F'\langle\langle B:=A_2\rangle\rangle]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}$ hence $Rate(F_1,a,l,C)=\tilde{\lambda}=Rate(F_2,a,l,C)$.
- If $d>1$, then several subcases arise depending on the syntactical structure of F.
  - If $F\equiv F'/L$, then for $j\in\{1,2\}$ we have $F_j\equiv(F'\langle\langle B:=A_j\rangle\rangle)/L$. Since F_1 has a potential move having type a (with $a\notin L$), priority level l, and derivative term in C, such that the depth of its inference is d, $F'\langle\langle B:=A_1\rangle\rangle$ has a potential move having type b (with $b=a$ if $a\neq\tau$, $b\in L\cup\{\tau\}$ if $a=\tau$), priority level l, and derivative term $G\in C'\in\mathscr{G}_\Theta/(\sim_{EMB}\mathscr{B}'\sim_{EMB})$, such that the depth of its inference is $d-1$ and $C=[G/L]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}$. For $j\in\{1,2\}$ we have

$$Rate(F_j,a,l,C)=\begin{cases} Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C') \\ Rate(F'\langle\langle B:=A_j\rangle\rangle,\tau,l,C')\,Min \\ \quad Min\{\!|Rate(F'\langle\langle B:=A_j\rangle\rangle,b,l,C')\mid b\in L|\!\} \end{cases}$$

  depending on whether $a\notin L\cup\{\tau\}$ or $a=\tau$. From the induction hypothesis, it follows that $Rate(F_1,a,l,C)=Rate(F_2,a,l,C)$.
  - If $F\equiv F'[\varphi]$, then the proof is similar to the one developed in the first subcase. The result follows by applying the induction hypothesis to the fact that for $j\in\{1,2\}$ we have

$$Rate(F_j,a,l,C)=Min\{\!|Rate(F'\langle\langle B:=A_j\rangle\rangle,b,l,C')\mid\varphi(b)=a|\!\}$$

  - If $F\equiv\Theta(F')$, then the proof is similar to the one developed in the first subcase. The result follows by applying the induction hypothesis to the fact that for $j\in\{1,2\}$ we have

$$Rate(F_j,a,l,C)=Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C')$$

  - If $F\equiv F'+F''$, then for $j\in\{1,2\}$ we have $F_j\equiv(F'\langle\langle B:=A_j\rangle\rangle)+(F''\langle\langle B:=A_j\rangle\rangle)$. Since F_1 has a potential move having type a, priority level l, and derivative term in C, such that the depth of its inference is d, $F'\langle\langle B:=A_1\rangle\rangle$ ($F''\langle\langle B:=A_1\rangle\rangle$) has the same move but the depth of its inference is $d-1$. For $j\in\{1,2\}$ we have

$$\begin{aligned} &Rate(F_j,a,l,C)\\ &\quad=Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C)\,Min\,Rate(F''\langle\langle B:=A_j\rangle\rangle,a,l,C)\end{aligned}$$

  From the induction hypothesis, it follows that $Rate(F_1,a,l,C)=Rate(F_2,a,l,C)$.
  - If $F\equiv F'\,\|_S\,F''$, then for $j\in\{1,2\}$ we have $F_j\equiv(F'\langle\langle B:=A_j\rangle\rangle)\,\|_S\,(F''\langle\langle B:=A_j\rangle\rangle)$. Suppose that F_1 has a potential move having type a, priority level l, and derivative term in C, such that the depth of its inference is d.
    - If $a\notin S$, then $F'\langle\langle B:=A_1\rangle\rangle$ ($F''\langle\langle B:=A_1\rangle\rangle$) has a potential move having type a, priority level l, and derivative term in $G\in C'\in\mathscr{G}_\Theta/(\sim_{EMB}\mathscr{B}'\sim_{EMB})$, such

that the depth of its inference is $d-1$ and $C=[G\,\|_S\,(F''\langle\langle B:=A_1\rangle\rangle)]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}$ ($C=[(F'\langle\langle B:=A_1\rangle\rangle)\,\|_S\,G]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}$). For $j\in\{1,2\}$ we have

$$Rate(F_j,a,l,C)=\begin{cases} Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C')\,Min \\ \quad Rate(F''\langle\langle B:=A_j\rangle\rangle,a,l,[F''\langle\langle B:=A_j\rangle\rangle]_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}) \\ Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C') \end{cases}$$

depending on whether $F'\langle\langle B:=A_j\rangle\rangle\in C'$ or $F'\langle\langle B:=A_j\rangle\rangle\notin C'$. From the induction hypothesis, it follows that $Rate(F_1,a,l,C)=Rate(F_2,a,l,C)$.

* If $a\in S$, then $F'\langle\langle B:=A_1\rangle\rangle$ ($F''\langle\langle B:=A_1\rangle\rangle$) has a potential move having type a, priority level l, and derivative term in $G'\in C'\in\mathscr{G}_\Theta/(\sim_{EMB}\,\mathscr{B}'\,\sim_{EMB})$, such that the depth of its inference is at most $d-1$, and $F''\langle\langle B:=A_1\rangle\rangle$ ($F'\langle\langle B:=A_1\rangle\rangle$) has a potential move having type a, priority level -1, and derivative term in $G''\in C''\in\mathscr{G}_\Theta/(\sim_{EMB}\,\mathscr{B}'\,\sim_{EMB})$, such that the depth of its inference is at most $d-1$; besides, $C=[G'\,\|_S\,G'']_{\sim_{EMB}\mathscr{B}'\sim_{EMB}}$. For $j\in\{1,2\}$ we have

$$Rate(F_j,a,l,C)$$
$$=\begin{cases} Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C')\,Min\,Rate(F''\langle\langle B:=A_j\rangle\rangle,a,l,C'') \\ Rate(F'\langle\langle B:=A_j\rangle\rangle,a,l,C') \end{cases}$$

depending on whether $Rate(F'\langle\langle B:=A_j\rangle\rangle,a,-1,C')\neq\perp$ or $Rate(F'\langle\langle B:=A_j\rangle\rangle,a,-1,C')=\perp$. From the induction hypothesis, it follows that $Rate(F_1,a,l,C)=Rate(F_2,a,l,C)$.

- If $F\equiv B'$, then for $j\in\{1,2\}$ we have $F_j\equiv B'\langle\langle B:=A_j\rangle\rangle$.
  * If $B'\equiv B$, then for $j\in\{1,2\}$ we have $F_j\equiv A_j$. Since F_1 has a potential move having type a, priority level l, and derivative term in C, such that the depth of its inference is d, then $E_1\langle\langle A:=A_1\rangle\rangle$ has the same potential move but the depth of its inference is $d-1$. For $j\in\{1,2\}$ we have $Rate(F_j,a,l,C)=Rate(E_j,a,l,C)$. From the induction hypothesis and the fact that $E_1\sim_{EMB}E_2$, it follows that $Rate(F_1,a,l,C)=Rate(F_2,a,l,C)$.
  * If $B'\not\equiv B$, then $B'\in\mathscr{G}_\Theta$ and the result trivially follows from the fact that $F_1\equiv B'\equiv F_2$. □

Proof of Theorem 5.21. It follows immediately from the fact that $\Theta(E_1)\sim_{EMB}\Theta(E_2)$ and from the definitions of $\sim_{EMB}$ and $\sim_F$. □

Proof of Theorem 5.22. Let $E_1,E_2\in\mathscr{E}$ such that $E_1\sim_{EMB}E_2$, and let $\mathscr{B}\subseteq\mathscr{G}_\Theta\times\mathscr{G}_\Theta$ be a strong EMB such that $(\Theta(E_1),\Theta(E_2))\in\mathscr{B}$. Let us group the steps of the algorithm for determining the Markovian semantics into macrosteps, where a given macrostep results in the elimination of the forks of immediate transitions whose upstream vanishing states belong to the same vanishing equivalence class of $(\uparrow E_1\cup\uparrow E_2)/\mathscr{B}$. Let us denote by $\hat{\mathscr{P}}_0[\![E_1]\!]$ and $\hat{\mathscr{P}}_0[\![E_2]\!]$ the two PLTSs produced by the execution of the first step (up to immediate selfloop removal), and by $\hat{\mathscr{P}}_h[\![E_1]\!]$ and $\hat{\mathscr{P}}_h[\![E_2]\!]$ the two PLTSs produced by the execution of macrostep $h \geqslant 1$ related to vanishing equivalence class C_h (up to removal of immediate selfloops incident on states not belonging to C_h). The result follows by proving that $\hat{\mathscr{P}}_h[\![E_1]\!]$ and $\hat{\mathscr{P}}_h[\![E_2]\!]$ are p-bisimilar by proceeding by induction on the number $h \in \mathbb{N}$ of vanishing equivalence classes of $(\uparrow E_1 \cup \uparrow E_2)/\mathscr{B}$.

- Let $h = 0$. We prove that

  $$\mathscr{B}_0 = \mathscr{B} \cap ((S_{E_1,1} \cup S_{E_2,1}) \times (S_{E_1,1} \cup S_{E_2,1}))$$

  is a p-bisimulation between $\hat{\mathscr{P}}_0[\![E_1]\!]$ and $\hat{\mathscr{P}}_0[\![E_2]\!]$. Observe that $\mathscr{B}_0$ is an equivalence relation, and that

  $$(S_{E_1,1} \cup S_{E_2,1})/\mathscr{B}_0 = \{C \mid C = C' \cap ((S_{E_1,1} \cup S_{E_2,1}) \times (S_{E_1,1} \cup S_{E_2,1})) \neq \emptyset \wedge C' \in \mathscr{G}_\Theta/\mathscr{B}\}$$

  – Let $C \in (S_{E_1,1} \cup S_{E_2,1})/\mathscr{B}_0$. For $j \in \{1,2\}$ we have

  $$\sum_{s \in C \cap S_{E_j,1}} P_{E_j,1}(s) = \begin{cases} 1 & \text{if } E_j \in C \\ 0 & \text{if } E_j \notin C \end{cases}$$

  Since $(E_1, E_2) \in \mathscr{B}$, it turns out that $E_1 \in C$ if and only if $E_2 \in C$, so

  $$\sum_{s \in C \cap S_{E_1,1}} P_{E_1,1}(s) = \sum_{s \in C \cap S_{E_2,1}} P_{E_2,1}(s)$$

  – Let $(s_1, s_2) \in \mathscr{B}_0 \cap (S_{E_1,1} \times S_{E_2,1})$. Let $C \in (S_{E_1,1} \cup S_{E_2,1})/\mathscr{B}_0$, and C' be the corresponding equivalence class in $\mathscr{G}_\Theta/\mathscr{B}$. Since $h = 0$, s_1 and s_2 are tangible. For $j \in \{1,2\}$ we have

  $$\begin{aligned} & Min\{\!|\lambda \mid s_j \xrightarrow{\lambda}_{E_j,1} s'_j \wedge s'_j \in C \cap S_{E_j,1}|\!\} \\ & = Min\{\!|\lambda \mid s_j \xrightarrow{a,\lambda}_{E_j} s'_j \wedge a \in AType \wedge s'_j \in C \cap S_{E_j,1}|\!\} \\ & = Min\{\!|Rate(s_j, a, 0, C') \mid a \in AType|\!\} \end{aligned}$$

  as $S_{E_j,1} = \uparrow E_j$. Since $(s_1, s_2) \in \mathscr{B}$ and $C' \in \mathscr{G}_\Theta/\mathscr{B}$, it turns out

  $$\begin{aligned} & Min\{\!|\lambda \mid s_1 \xrightarrow{\lambda}_{E_1,1} s'_1 \wedge s'_1 \in C \cap S_{E_1,1}|\!\} \\ & = Min\{\!|\lambda \mid s_2 \xrightarrow{\lambda}_{E_2,1} s'_2 \wedge s'_2 \in C \cap S_{E_2,1}|\!\} \end{aligned}$$

- Let $h \geqslant 1$ and suppose that $\hat{\mathscr{P}}_{h-1}[\![E_1]\!]$ and $\hat{\mathscr{P}}_{h-1}[\![E_2]\!]$ are p-bisimilar via $\mathscr{B}_{h-1} \subseteq (S_{E_1,k_1} \cup S_{E_2,k_2}) \times (S_{E_1,k_1} \cup S_{E_2,k_2})$. Let $C_h \in (S_{E_1,k_1} \cup S_{E_2,k_2})/\mathscr{B}_{h-1}$ be the vanishing equivalence class considered during macrostep h, and let S_{E_1,k'_1} and S_{E_2,k'_2} be the set of states of $\hat{\mathscr{P}}_h[\![E_1]\!]$ and $\hat{\mathscr{P}}_h[\![E_2]\!]$, respectively. We prove that

  $$\mathscr{B}_h = \mathscr{B}_{h-1} \cap ((S_{E_1,k'_1} \cup S_{E_2,k'_2}) \times (S_{E_1,k'_1} \cup S_{E_2,k'_2}))$$

is a p-bisimulation between $\hat{\mathscr{P}}_h[\![E_1]\!]$ and $\hat{\mathscr{P}}_h[\![E_2]\!]$. Observe that $\mathscr{B}_h$ is an equivalence relation, and that

$$(S_{E_1,k_1'} \cup S_{E_2,k_2'})/\mathscr{B}_h = \{C \mid C = C' \cap ((S_{E_1,k_1'} \cup S_{E_2,k_2'}) \times (S_{E_1,k_1'} \cup S_{E_2,k_2'})) \\ \neq \emptyset \wedge C' \in (S_{E_1,k_1} \cup S_{E_2,k_2})/\mathscr{B}_{h-1}\}$$

- Let $C \in (S_{E_1,k_1'} \cup S_{E_2,k_2'})/\mathscr{B}_h$, and C' be the corresponding class in $(S_{E_1,k_1} \cup S_{E_2,k_2})/\mathscr{B}_{h-1}$.
  * If $C_h \cap (S_{E_1,k_1'} \cup S_{E_2,k_2'}) \neq \emptyset$, then there is no state in C_h having transitions to states not in C_h, so for $j \in \{1,2\}$ we have

$$\sum_{s \in C \cap S_{E_j,k_j'}} P_{E_j,k_j'}(s) = \sum_{s \in C' \cap S_{E_j,k_j}} P_{E_j,k_j}(s)$$

From the induction hypothesis, it follows that

$$\sum_{s \in C \cap S_{E_1,k_1'}} P_{E_1,k_1'}(s) = \sum_{s \in C \cap S_{E_2,k_2'}} P_{E_2,k_2'}(s)$$

  * If $C_h \cap (S_{E_1,k_1'} \cup S_{E_2,k_2'}) = \emptyset$, given $s_h \in C_h \cap S_{E_1,k_1}$ let $p, q \in \mathbb{R}_{[0,1]}$ be defined by

$$\infty_{l,p} = Min\{\!|\infty_{l,p'} \mid s_h \xrightarrow{\infty_{l,p'}}{}_{E_1,k_1} s' \wedge s' \in C' \cap S_{E_1,k_1}|\!\}$$

and

$$\infty_{l,q} = Min\{\!|\infty_{l,p'} \mid s_h \xrightarrow{\infty_{l,p'}}{}_{E_1,k_1} s' \wedge s' \notin C_h \cap S_{E_1,k_1}|\!\}$$

Note that q is well defined because each state in C_h has transitions to states not in C_h, whereas p could be undefined and in this case p is taken to be 0 for convenience. Then for $j \in \{1,2\}$ we have

$$\sum_{s \in C \cap S_{E_j,k_j'}} P_{E_j,k_j'}(s) = \sum_{s \in C' \cap S_{E_j,k_j}} P_{E_j,k_j}(s) + \frac{p}{q} \cdot \sum_{s_h \in C_h \cap S_{E_j,k_j}} P_{E_j,k_j}(s_h)$$

From the induction hypothesis, it follows that

$$\sum_{s \in C \cap S_{E_1,k_1'}} P_{E_1,k_1'}(s) = \sum_{s \in C \cap S_{E_2,k_2'}} P_{E_2,k_2'}(s)$$

- Let $(s_1, s_2) \in \mathscr{B}_h \cap (S_{E_1,k_1'} \times S_{E_2,k_2'})$. Let $C \in (S_{E_1,k_1'} \cup S_{E_2,k_2'})/\mathscr{B}_h$, and C' be the corresponding equivalence class in $(S_{E_1,k_1} \cup S_{E_2,k_2})/\mathscr{B}_{h-1}$. Note that $(s_1, s_2) \in \mathscr{B}_{h-1} \cap (S_{E_1,k_1} \times S_{E_2,k_2})$.
  * If $C_h \cap (S_{E_1,k_1'} \cup S_{E_2,k_2'}) \neq \emptyset$, then there is no state in C_h having transitions to states not in C_h, so for $j \in \{1,2\}$ we have

$$Min\{\!|\tilde{\lambda} \mid s_j \xrightarrow{\tilde{\lambda}}{}_{E_j,k_j'} s_j' \wedge s_j' \in C \cap S_{E_j,k_j'}|\!\}$$

$$= Min\{\!|\tilde{\lambda} \mid s_j \xrightarrow{\tilde{\lambda}}{}_{E_j,k_j} s_j' \wedge s_j' \in C' \cap S_{E_j,k_j}|\!\}$$

From the induction hypothesis, it follows that

$$Min\{\!|\tilde{\lambda} \mid s_1 \xrightarrow{\tilde{\lambda}}_{E_1,k_1'} s_1' \wedge s_1' \in C \cap S_{E_1,k_1'}|\!\}$$

$$= Min\{\!|\tilde{\lambda} \mid s_2 \xrightarrow{\tilde{\lambda}}_{E_2,k_2'} s_2' \wedge s_2' \in C \cap S_{E_2,k_2'}|\!\}$$

* If $C_h \cap (S_{E_1,k_1'} \cup S_{E_2,k_2'}) = \emptyset$, given $s_h \in C_h \cap S_{E_1,k_1}$ let $p, q \in \mathbb{R}_{[0,1]}$ be defined by

$$\infty_{l,p} = Min\{\!|\infty_{l,p'} \mid s_h \xrightarrow{\infty_{l,p'}}_{E_1,k_1} s' \wedge s' \in C' \cap S_{E_1,k_1}|\!\}$$

and

$$\infty_{l,q} = Min\{\!|\infty_{l,p'} \mid s_h \xrightarrow{\infty_{l,p'}}_{E_1,k_1} s' \wedge s' \notin C_h \cap S_{E_1,k_1}|\!\}$$

Note that q is well defined because each state in C_h has transitions to states not in C_h, whereas p could be undefined and in this case p is taken to be 0 for convenience. Then for $j \in \{1,2\}$ we have

$$Min\{\!|\tilde{\lambda} \mid s_j \xrightarrow{\tilde{\lambda}}_{E_j,k_j'} s_j' \wedge s_j' \in C \cap S_{E_j,k_j'}|\!\}$$

$$= Min\{\!|\tilde{\lambda} \mid s_j \xrightarrow{\tilde{\lambda}}_{E_j,k_j} s_j' \wedge s_j' \in C' \cap S_{E_j,k_j}|\!\}\ Min$$

$$Split\left(Min\{\!|\tilde{\lambda} \mid s_j \xrightarrow{\tilde{\lambda}}_{E_j,k_j} s_h \wedge s_h \in C_h \cap S_{E_j,k_j}|\!\}, \frac{p}{q}\right)$$

From the induction hypothesis, it follows that

$$Min\{\!|\tilde{\lambda} \mid s_1 \xrightarrow{\tilde{\lambda}}_{E_1,k_1'} s_1' \wedge s_1' \in C \cap S_{E_1,k_1'}|\!\}$$

$$= Min\{\!|\tilde{\lambda} \mid s_2 \xrightarrow{\tilde{\lambda}}_{E_2,k_2'} s_2' \wedge s_2' \in C \cap S_{E_2,k_2'}|\!\} \qquad \square$$

Proof of Theorem 5.26. ($\Rightarrow$) Since $E_1 \sim_{EMB} E_2$ and $\sim_{EMB}$ is a congruence, for all $F \in \mathscr{G}$ and $S \subseteq AType - \{\tau\}$ we have $E_1 + F \sim_{EMB} E_2 + F$ and $E_1 \|_S F \sim_{EMB} E_2 \|_S F$. Since $\sim_{EMB} \subset \sim_{FP}$ in $\mathscr{E} \times \mathscr{E}$, the result follows.

($\Leftarrow$) We prove the contrapositive, so we assume that $E_1 \not\sim_{EMB} E_2$ and we demonstrate that E_1 and E_2 are distinguishable with respect to $\sim_{FP}$ by means of an appropriate context based on the alternative composition operator or the parallel composition operator. We proceed by induction on the number n of actions that E_1 and E_2 have to execute in order to become E_1' and E_2', respectively, such that $E_1' \not\sim_{EMB} E_2'$ because they violate the necessary condition expressed by Proposition 5.11 (ii).[5]

- Let $n = 0$, i.e. assume that there exist $a \in AType$ and $l \in APLev$ such that $Min\{\!|Rate(E_1, a, l, \{E\}) \mid E \in \mathscr{E}_{-\tau\infty}|\!\} \neq Min\{\!|Rate(E_2, a, l, \{E\}) \mid E \in \mathscr{E}_{-\tau\infty}|\!\}$. Since both sides of

[5] If such E_1' and E_2' did not exist, then $E_1 \not\sim_{EMB} E_2$ would not hold. We recall that we are considering guardedly closed terms, which are finitely branching.

the inequality cannot be $\perp$, we assume that $Min\{\!|Rate(E_1,a,l,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}$ $\neq \perp$. There are several cases:

- Assume that $Min\{\!|Rate(E_2,a,l,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} \neq \perp$.
  - Assume that $l=0$.
    - If $a=\tau$, or $a\neq\tau$ and $Min\{\!|Rate(E_1,\tau,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} \neq Min\{\!|Rate(E_2,\tau,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}$, then

$$E_1 \,\|_{AType-\{\tau\}}\, \underline{0} \not\sim_P E_2 \,\|_{AType-\{\tau\}}\, \underline{0}$$

because the aggregated rates of these two terms are $Min\{\!|Rate(E_1,\tau,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}$ and $Min\{\!|Rate(E_2,\tau,0,\{E\})|E\in\mathscr{E}_{-\tau\infty}|\!\}$, respectively, hence p-bisimilarity is violated.

    - If $a\neq\tau$ and $Min\{\!|Rate(E_1,\tau,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} = Min\{\!|Rate(E_2,\tau,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}$, then

$$E_1 \,\|_{AType-\{\tau\}}\, \langle a,*\rangle.\underline{0} \not\sim_P E_2 \,\|_{AType-\{\tau\}}\, \langle a,*\rangle.\underline{0}$$

because the aggregated rates of these two terms are $Min\{\!|Rate(E_1,a,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}\ Min\ Min\{\!|Rate(E_1,\tau,0,\{E\})\,|E\in\mathscr{E}_{-\tau\infty}|\!\}$ and $Min\{\!|Rate(E_2,a,0,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}\ Min\ Min\{\!|Rate(E_2,\tau,0,\{E\})\,|E\in\mathscr{E}_{-\tau\infty}|\!\}$, respectively, hence p-bisimilarity is violated.

  - Assume that $l\geqslant 1$. From $E_1,E_2\in\mathscr{E}_{-\tau\infty}$, it follows that $a\neq\tau$ and $Min\{\!|Rate(E_1,\tau,l,\{E\})|E\in\mathscr{E}_{-\tau\infty}|\!\} = \perp = Min\{\!|Rate(E_2,\tau,l,\{E\})|E\in\mathscr{E}_{-\tau\infty}|\!\}$. As a consequence, given $b\in AType-\{\tau\}$ occurring neither in E_1 nor in E_2, and $\lambda\in\mathbb{R}_+$ smaller than the smallest rate of the possible exponentially timed actions occurring in E_1 and E_2, we have that

$$E_1 \,\|_{AType-\{\tau\}-\{b\}}\, (\langle a,*\rangle.\underline{0} + \langle b,\infty_{l,w}\rangle.\langle b,\lambda\rangle.\underline{0}) \not\sim_P$$

$$E_2 \,\|_{AType-\{\tau\}-\{b\}}\, (\langle a,*\rangle.\underline{0} + \langle b,\infty_{l,w}\rangle.\langle b,\lambda\rangle.\underline{0})$$

due to the fact that the state having the transition labeled with b,λ has initial state probability $w/(Min\{\!|Rate(E_1,a,l,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} + w)$ in the Markovian semantics of the left-hand side term, $w/(Min\{\!|Rate(E_2,a,l,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} + w)$ in the Markovian semantics of the right-hand side term. [6]

- Assume that $Min\{\!|Rate(E_2,a,l,\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\} = \perp$.
  - Assume that there exists $l'\in APLev-\{l\}$ such that $Min\{\!|Rate(E_2,a,l',\{E\})\,|\,E\in\mathscr{E}_{-\tau\infty}|\!\}\neq\perp$. From $E_1,E_2\in\mathscr{E}_{-\tau\infty}$, it follows that $a\neq\tau$. Then, given $b\in AType-\{\tau\}$ occurring neither in E_1 nor in E_2, we have that

$$E_1 \,\|_{AType-\{\tau\}-\{b\}}\, (\langle a,*\rangle.\underline{0} + \langle b,\infty_{\max(l,l'),w}\rangle.\underline{0}) \not\sim_F$$

[6] The choice of b makes the determination of initial state probabilities easier, while the choice of λ is important in the case we are interested in the lumped Markovian semantics (see Section 5.6). If we ignore the constraint on λ, then e.g. $E_1 \equiv \langle a,\infty_{1,1}\rangle.\langle\tau,\mu\rangle.\underline{0}$ and $E_2 \equiv \langle a,\infty_{1,2}\rangle.\langle\tau,\mu\rangle.\underline{0}$ may not be distinguished with respect to $\sim_P$ by means of the context above, because the ordinary lumping may come into play and the state considered above may have the same initial state probability in the Markovian semantics of both resulting terms.

$$E_2 \,\|_{AType-\{\tau\}-\{b\}}\, (\langle a,*\rangle.\underline{0} + \langle b,\infty_{\max(l,l'),w}\rangle.\underline{0})$$

because one of these two terms can execute an action with type b while the other term cannot.

* Assume that $Min\{\!|Rate(E_2,a,l',\{E\}) \mid l' \in APLev \wedge E \in \mathscr{E}_{-\tau\infty}|\!\} = \bot$.
  · If $a = \tau$, then

$$E_1 \,\|_{AType-\{\tau\}}\, \underline{0} \not\sim_F E_2 \,\|_{AType-\{\tau\}}\, \underline{0}$$

because the left-hand side term can execute an action with type τ while the right-hand side term cannot.

  · If $a \neq \tau$, then

$$E_1 \,\|_{AType-\{\tau\}}\, \langle a,*\rangle.\underline{0} \not\sim_F E_2 \,\|_{AType-\{\tau\}}\, \langle a,*\rangle.\underline{0}$$

because the left-hand side term can execute an action with type a while the right-hand side term cannot.

- Let $n \geqslant 1$, i.e. assume that for all $a \in AType$ and $l \in APLev$ it turns out that $Min\{\!|Rate(E_1,a,l,\{E\}) \mid E \in \mathscr{E}_{-\tau\infty}|\!\} = Min\{\!|Rate(E_2,a,l,\{E\}) \mid E \in \mathscr{E}_{-\tau\infty}|\!\}$. From $E_1 \not\sim EMBE_2$ and the hypothesis above, it follows that there exist $a \in AType$, $l \in APLev$ and $C, C' \in \mathscr{E}_{-\tau\infty}/\sim_{EMB}$ such that $Rate(E_1,a,l,C) \neq Rate(E_2,a,l,C)$, $Rate(E_1,a,l,C') \neq Rate(E_2,a,l,C')$, $C \cap C' = \emptyset$, and $Rate(E_1,a,l,C)\, Min\, Rate(E_1,a,l,C') \neq \bot \neq Rate(E_2,a,l,C)\, Min\, Rate(E_2,a,l,C')$. As a consequence, there exist $F_1 \in C$ (C') and $F_2 \in C'$ (C) reachable from E_1 and E_2, respectively, by executing an action having type a and priority level l, such that we can apply the induction hypothesis to F_1 and F_2. Let "$\_ \,\|_{AType-\{\tau\}-S}\, F$" be the context that distinguishes F_1 and F_2 with respect to $\sim_{FP}$. If $a = \tau$, then

$$E_1 \,\|_{AType-\{\tau\}-S}\, F \not\sim_{FP} E_2 \,\|_{AType-\{\tau\}-S}\, F$$

else

$$E_1 \,\|_{AType-\{\tau\}-S}\, \langle a,*\rangle.F \not\sim_{FP} E_2 \,\|_{AType-\{\tau\}-S}\, \langle a,*\rangle.F$$

provided that the constraint on the possible element of S, and the constraint on the rate of the possible exponentially timed action occurring in F, are satisfied. □

Proof of Lemma 5.32. Given $E \in \mathscr{G}_{\Theta,nrec}$, we proceed by induction on $size(E)$:

- Let $size(E) = 1$. The result follows by proving by induction on the syntactical structure of E that $\mathscr{A} \vdash E = \underline{0}$:
  - If $E \equiv \underline{0}$, then we take $F \equiv \underline{0}$ and the result follows by reflexivity.
  - The case $E \equiv \langle a,\tilde{\lambda}\rangle.E'$ is not possible because it contradicts the hypothesis $size(E) = 1$.
  - If $E \equiv E'/L$, then E' is a subterm of E such that $size(E') = 1$, hence $\mathscr{A} \vdash E' = \underline{0}$ by structural induction. The result follows by substitutivity, $\mathscr{A}_5$ and transitivity.

- If $E \equiv E'[\varphi]$ or $E \equiv \Theta(E')$, then the result can be proved by proceeding as in the previous point and by exploiting $\mathscr{A}_8$ in the first case and $\mathscr{A}_{11}$ in the second case.
- If $E \equiv E_1 + E_2$, then E_1 and E_2 are subterms of E such that $size(E_1) = 1$ and $size(E_2) = 1$, hence $\mathscr{A} \vdash E_1 = \underline{0}$ and $\mathscr{A} \vdash E_2 = \underline{0}$ by structural induction. The result follows by substitutivity, $\mathscr{A}_3$ and transitivity.
- The case $E \equiv E_1 \|_S E_2$ is not possible because it contradicts the hypothesis $size(E) = 1$.

- Let the result hold whenever $size(E) \leqslant n \in \mathbb{N}_+$, and assume $size(E) = n+1$. The result follows by proceeding by induction on the syntactical structure of E:
  - The case $E \equiv \underline{0}$ is not possible because it contradicts the hypothesis $size(E) = n + 1 \geqslant 2$.
  - If $E \equiv \langle a, \tilde{\lambda} \rangle.E'$, then $size(E') = n$ hence by the induction hypothesis there exists $F' \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash E' = F'$. The result follows by substitutivity.
  - If $E \equiv E'/L$, then E' is a subterm of E hence by structural induction there exists $F' \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash E' = F'$. By substitutivity we obtain $\mathscr{A} \vdash E = F'/L$. Assuming $F' \equiv \sum_{i \in I} \langle a_i, \tilde{\lambda}_i \rangle.F_i'$ where every F_i' is in snf, by exploiting $\mathscr{A}_7$, $\mathscr{A}_6$ and transitivity we obtain $\mathscr{A} \vdash F'/L = \sum_{i \in I}((\langle a_i, \tilde{\lambda}_i \rangle.F_i')/L) = \sum_{i \in I \wedge a_i \in L} \langle \tau, \tilde{\lambda}_i \rangle.(F_i'/L) + \sum_{i \in I \wedge a_i \notin L} \langle a_i, \tilde{\lambda}_i \rangle.(F_i'/L)$. Since for every $i \in I$ we have $size(F_i'/L) = size(F_i') < size(F') = size(F'/L) \leqslant size(E) = n + 1$, by the induction hypothesis it follows that for every $i \in I$ there exists $F_i'' \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash F_i'/L = F_i''$. If we take $F \equiv \sum_{i \in I \wedge a_i \in L} \langle \tau, \tilde{\lambda}_i \rangle.F_i'' + \sum_{i \in I \wedge a_i \notin L} \langle a_i, \tilde{\lambda}_i \rangle.F_i''$, then the result follows by substitutivity.
  - If $E \equiv E'[\varphi]$ or $E \equiv \Theta(E')$, then the result can be proved by proceeding as in the previous point and by exploiting $\mathscr{A}_{10}$ and $\mathscr{A}_9$ in the first case and $\mathscr{A}_{12}$ in the second case.
  - If $E \equiv E_1 + E_2$, then E_1 and E_2 are subterms of E hence by structural induction there exist $F_1, F_2 \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash E_1 = F_1$ and $\mathscr{A} \vdash E_2 = F_2$. By substitutivity we obtain $\mathscr{A} \vdash E = F_1 + F_2$ and the result follows after a possible application of $\mathscr{A}_3$.
  - If $E \equiv E_1 \|_S E_2$, then $size(E_1) \leqslant n$ and $size(E_2) \leqslant n$ hence by the induction hypothesis there exist $F_1', F_2' \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash E_1 = F_1'$ and $\mathscr{A} \vdash E_2 = F_2'$. By substitutivity we obtain $\mathscr{A} \vdash E = F_1' \|_S F_2'$. There are three cases:
    * If $size(E_1) = size(E_2) = 1$, then $F_1' \equiv F_2' \equiv \underline{0}$ hence $\mathscr{A} \vdash E = \underline{0}$ by $\mathscr{A}_{13}$, $\mathscr{A}_3$ and transitivity.
    * If $size(E_1) = 1$ and $size(E_2) > 1$, then $F_1' \equiv \underline{0}$ and $F_2' \equiv \sum_{i \in I} \langle a_i, \tilde{\lambda}_i \rangle.F_{2,i}'$ where every $F_{2,i}'$ is in snf. By $\mathscr{A}_{13}$, $\mathscr{A}_3$ and transitivity we obtain $\mathscr{A} \vdash F_1' \|_S F_2' = \sum_{j \in J} \langle a_j, \tilde{\lambda}_j \rangle.(\underline{0} \|_S F_{2,j}')$. Since for every $j \in J$ we have $size(\underline{0} \|_S F_{2,j}') < size(\underline{0} \|_S F_2) \leqslant size(E) = n + 1$, by the induction hypothesis it follows that for every $j \in J$ there exists $F_{2,j}''$ in snf such that $\mathscr{A} \vdash \underline{0} \|_S F_{2,j}' = F_{2,j}''$, hence the result follows by substitutivity. The proof for the symmetric case is similar.
    * If $size(E_1) > 1$ and $size(E_2) > 1$, then $F_1' \equiv \sum_{i \in I_1} \langle a_i, \tilde{\lambda}_i \rangle.F_{1,i}'$ and $F_2' \equiv \sum_{i \in I_2} \langle a_i, \tilde{\lambda}_i \rangle.F_{2,i}'$ where every $F_{1,i}'$ and $F_{2,i}'$ is in snf. By $\mathscr{A}_{13}$ and transitivity we obtain

$\mathscr{A} \vdash F_1' \|_S F_2' = \sum_{j\in J_1}\langle a_j,\tilde{\lambda}_j\rangle.(F_{1,j}' \|_S F_2') + \sum_{j\in J_2}\langle a_j,\tilde{\lambda}_j\rangle.(F_1' \|_S F_{2,j}') + \sum_{k\in K_1} \sum_{h\in H_k}\langle a_k, Split(\tilde{\lambda}_k, 1/n_k)\rangle.(F_{1,k}' \|_S F_{2,h}') + \sum_{k\in K_2}\sum_{h\in H_k}\langle a_k, Split(\tilde{\lambda}_k, 1/n_k)\rangle. (F_{1,h}' \|_S F_{2,k}')$. Since the size of every term surrounded by parentheses is at most n, by the induction hypothesis each such term can be proved equal via $\mathscr{A}$ to a term in snf, hence the result follows by substitutivity.

Proof of Theorem 5.33. We must prove that for any $E_1, E_2 \in \mathscr{G}_{\Theta,nrec}$ we have $\mathscr{A} \vdash E_1 = E_2 \Leftrightarrow E_1 \sim_{EMB} E_2$.

($\Rightarrow$) It is a straightforward consequence of the fact that reflexivity, symmetry, transitivity and substitutivity of $Ded(\mathscr{A})$ are matched by the reflexive, symmetric, transitive and congruence properties of $\sim_{EMB}$, and the fact that every axiom in $\mathscr{A}$ can be restated as a property of $\sim_{EMB}$.

($\Leftarrow$) Assume $E_1 \sim_{EMB} E_2$. There are two cases:

- If E_1 and E_2 are both in snf, the result follows by proceeding by induction on $size(E_1)$:
  * If $size(E_1) = 1$, then $E_1 \equiv \underline{0} \equiv E_2$ since they are both in snf. The result follows by reflexivity.
  * If $size(E_1) > 1$, then $E_1 \equiv \sum_{i\in I_1}\langle a_{1,i}, \tilde{\lambda}_{1,i}\rangle.E_{1,i}$ and $E_2 \equiv \sum_{i\in I_2}\langle a_{2,i}, \tilde{\lambda}_{2,i}\rangle. E_{2,i}$. It is not restrictive to assume that for $k \in \{1,2\}$ it holds $a_{k,i} = a_{k,j} \wedge PL(\langle a_{k,i}, \tilde{\lambda}_{k,i}\rangle) = PL(\langle a_{k,j}, \tilde{\lambda}_{k,j}\rangle) \wedge E_{k,i} \sim_{EMB} E_{k,j} \Rightarrow i = j$ because if this were not the case, then it would suffice to resort to finitely many applications of $\mathscr{A}_1$, $\mathscr{A}_2$ and $\mathscr{A}_4$. Since $E_1 \sim_{EMB} E_2$, we have that $|I_1| = |I_2|$ and for every summand $\langle a_{1,i}, \tilde{\lambda}_{1,i}\rangle.E_{1,i}$ in E_1 there exists exactly one summand $\langle a_{2,j}, \tilde{\lambda}_{2,j}\rangle.E_{2,j}$ in E_2 such that $a_{1,i} = a_{2,j} \wedge \tilde{\lambda}_{1,i} = \tilde{\lambda}_{2,j} \wedge E_{1,i} \equiv E_{2,j}$ (hence $E_{1,i} \sim_{EMB} E_{2,j}$), and viceversa. By the induction hypothesis we obtain that $\mathscr{A} \vdash E_{1,i} = E_{2,j}$, and the result follows by substitutivity.
- If E_1 or E_2 is not in snf, then by Lemma 5.32 there exist $F_1, F_2 \in \mathscr{G}_{\Theta,nrec}$ in snf such that $\mathscr{A} \vdash E_1 = F_1$ and $\mathscr{A} \vdash E_2 = F_2$. We then obtain $E_1 \sim_{EMB} F_1$ and $E_2 \sim_{EMB} F_2$ by soundness and $F_1 \sim_{EMB} F_2$ by the transitive property, hence $\mathscr{A} \vdash F_1 = F_2$ by the previous point. By transitivity we finally obtain $\mathscr{A} \vdash E_1 = E_2$.

Proof of Proposition 5.35. It suffices to take the reflexive, symmetric and transitive closure of the relation that associates every microstate of M_1 with the macrostate of M_2 that contains it.

Proof of Theorem 5.37. From Theorem 5.22 it follows that $\mathscr{M}[\![E_1]\!]$ is p-bisimilar to $\mathscr{M}[\![E_2]\!]$ via a given equivalence relation $\mathscr{B}_{\mathscr{M}}$. The result follows by proving that

$$\mathscr{B}_{\mathscr{M}_l} = \{(C_1, C_2) \in (S_{E_1,\mathscr{M}_l} \cup S_{E_2,\mathscr{M}_l}) \times (S_{E_1,\mathscr{M}_l} \cup S_{E_2,\mathscr{M}_l}) \mid \exists s_1 \in C_1. \exists s_2 \in C_2. (s_1, s_2) \in \mathscr{B}_{\mathscr{M}}\}$$

is a p-bisimulation between $\mathcal{M}_l[\![E_1]\!]$ and $\mathcal{M}_l[\![E_2]\!]$. Note that $\mathcal{B}_{\mathcal{M}_l}$ is an equivalence relation because so is $\mathcal{B}_{\mathcal{M}}$.

- Let $D \in (S_{E_1,\mathcal{M}_l} \cup S_{E_2,\mathcal{M}_l})/\mathcal{B}_{\mathcal{M}_l}$. For $j \in \{1,2\}$ we have

$$\sum_{C \in D \cap S_{E_j,\mathcal{M}_l}} P_{E_j,\mathcal{M}_l}(C) = \sum_{C \in D \cap S_{E_j,\mathcal{M}_l}} \sum_{s \in C} P_{E_j,\mathcal{M}}(s) = \sum_{s \in (\bigcup_{C \in D} C) \cap S_{E_j,\mathcal{M}}} P_{E_j,\mathcal{M}}(s)$$

  Since $\bigcup_{C \in D} C$ is the union of some equivalence classes with respect to $\mathcal{B}_{\mathcal{M}}$, and $\mathcal{B}_{\mathcal{M}}$ is a p-bisimulation, it follows that

$$\sum_{C \in D \cap S_{E_1,\mathcal{M}_l}} P_{E_1,\mathcal{M}_l}(C) = \sum_{C \in D \cap S_{E_2,\mathcal{M}_l}} P_{E_2,\mathcal{M}_l}(C)$$

- Let $(C_1, C_2) \in \mathcal{B}_{\mathcal{M}_l} \cap (S_{E_1,\mathcal{M}_l} \times S_{E_2,\mathcal{M}_l})$ due to the existence of $s_1 \in C_1$ and $s_2 \in C_2$ such that $(s_1, s_2) \in \mathcal{B}_{\mathcal{M}}$. Let $D \in (S_{E_1,\mathcal{M}_l} \cup S_{E_2,\mathcal{M}_l})/\mathcal{B}_{\mathcal{M}_l}$. For $j \in \{1,2\}$ we have

$$\begin{aligned}
&\sum \{\!|\lambda \mid C_j \xrightarrow{\lambda}_{E_j,\mathcal{M}_l} C'_j \wedge C'_j \in D \cap S_{E_j,\mathcal{M}_l}|\!\} \\
&\quad = \sum \{\!|\lambda \mid s_j \xrightarrow{\lambda}_{E_j,\mathcal{M}} s'_j \wedge s'_j \in C'_j \in D \cap S_{E_j,\mathcal{M}_l}|\!\} \\
&\quad = \sum \left\{\!\left|\lambda \mid s_j \xrightarrow{\lambda}_{E_j,\mathcal{M}} s'_j \wedge s'_j \in \left(\bigcup_{C \in D} C\right) \cap S_{E_j,\mathcal{M}}\right|\!\right\}
\end{aligned}$$

  where the first equality holds whichever is $s_j \in C_j$ because $\mathcal{M}_l[\![E_j]\!]$ is obtained from $\mathcal{M}[\![E_j]\!]$ via ordinary lumping. Since $\bigcup_{C \in D} C$ is the union of some equivalence classes with respect to $\mathcal{B}_{\mathcal{M}}$, and $\mathcal{B}_{\mathcal{M}}$ is a p-bisimulation, it follows that

$$\begin{aligned}
&\sum \{\!|\lambda \mid C_1 \xrightarrow{\lambda}_{E_1,\mathcal{M}_l} C'_1 \wedge C'_1 \in D \cap S_{E_1,\mathcal{M}_l}|\!\} \\
&\quad = \sum \{\!|\lambda \mid C_2 \xrightarrow{\lambda}_{E_2,\mathcal{M}_l} C'_2 \wedge C'_2 \in D \cap S_{E_2,\mathcal{M}_l}|\!\}
\end{aligned}$$

Proof of Corollary 5.38. The searched bijection is relation $\mathcal{B}_{\mathcal{M}_l}$ built in the proof of the previous theorem. The reason is that $\mathcal{B}_{\mathcal{M}_l}$ is a p-bisimulation and $\mathcal{M}_l[\![E_1]\!]$ and $\mathcal{M}_l[\![E_2]\!]$ are the coarsest lumpings of $\mathcal{M}[\![E_1]\!]$ and $\mathcal{M}[\![E_2]\!]$, respectively.

References

[1] M. Ajmone Marsan, G. Balbo, G. Conte, A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems, ACM Trans. Comput. Syst. 2 (1984) 143–172.

[2] M. Ajmone Marsan, A. Bianco, L. Ciminiera, R. Sisto, A. Valenzano, A LOTOS extension for the performance analysis of distributed systems, IEEE/ACM Trans. Networking 2 (1994) 151–164.

[3] J.C.M. Baeten, J.A. Bergstra, Real time process algebra, Formal Aspects Comput. 3 (1991) 142–188.

[4] J.C.M. Baeten, J.A. Bergstra, J.W. Klop, Syntax and defining equations for an interrupt mechanism in process algebra, Fund. Inform. IX (1986) 127–168.

[5] J.C.M. Baeten, J.A. Bergstra, S.A. Smolka, Axiomatizing probabilistic processes: ACP with generative probabilities, Inform. Comput. 121 (1995) 234–255.

[6] J.A. Bergstra, J.W. Klop, Process algebra for synchronous communication, Inform. and Control 60 (1984) 109–137.

[7] M. Bernardo, On the coexistence of exponential, immediate and passive actions in EMPA, in: [53], pp. 58–76.
[8] M. Bernardo, L. Donatiello, R. Gorrieri, Integrating Performance and Functional Analysis of Concurrent Systems with EMPA, Technical Report UBLCS-95-14, University of Bologna, Italy, 1995.
[9] M. Bernardo, R. Gorrieri, Extended Markovian Process Algebra, Proc. 7th Int. Conf. Concurrency Theory (CONCUR '96), Pisa, Italy, Lecture Notes in Computer Science, vol. 1119, Springer, Berlin, 1996, pp. 315–330.
[10] T. Bolognesi, E. Brinksma, Introduction to the ISO specification language LOTOS, Comput. Networks and ISDN Systems 14 (1987) 25–59.
[11] E. Brinksma, J.-P. Katoen, R. Langerak, D. Latella, A stochastic causality-based process algebra, in: [18], pp. 553–565.
[12] P. Buchholz, Markovian process algebra: Composition and equivalence, in: [28], pp. 11–30.
[13] J. Camilleri, G. Winskel, CCS with priority choice, Inform. and Comput. 116 (1995) 26–37.
[14] R. Cleaveland, M. Hennessy, Priorities in process algebras, Inform. and Comput. 87 (1990) 58–77.
[15] R. Cleaveland, G. Lüttgen, V. Natarajan, A process algebra with distributed priorities, Proc. 7th Int. Conf. Concurrency Theory (CONCUR '96), Pisa, Italy, Lecture Notes in Computer Science, vol. 1119, Springer, Berlin, 1996, pp. 34–49.
[16] J.-C. Fernandez, An Implementation of an efficient algorithm for bisimulation equivalence, Sci. Comput. Programming 13 (1989) 219–236.
[17] R. Gerber, I. Lee, A resource-based prioritized bisimulation for real-time systems, Inform. and Comput. 113 (1994) 102–142.
[18] S. Gilmore, J. Hillston (Eds.), Proc. of the 3rd Workshop on Process Algebras and Performance Modelling (PAPM '95), Comput. J. 38(7), Edinburgh, UK, 1995.
[19] R. van Glabbeek, S.A. Smolka, B. Steffen, Reactive, generative and stratified models of probabilistic processes, Inform. and Comput. 121 (1995) 59–80.
[20] R. Gorrieri, M. Roccetti, E. Stancampiano, A theory of processes with durational actions, Theoret. Comput. Sci. 140 (1995) 73–94.
[21] H. Hansson, B. Jonsson, A calculus for communicating systems with time and probabilities, Proc. 11th Real-Time Systems Symp. (RTSS '90), IEEE-Computer Society Press, Orlando, FL, 1990.
[22] P.G. Harrison, B. Strulo, Stochastic Process Algebra for Discrete Event Simulation, in: Quantitative Methods in Parallel Systems, Springer, Berlin, 1995, pp. 18–37.
[23] M. Hennessy, Algebraic Theory of Processes, MIT Press, Cambridge, MA, 1988.
[24] M. Hennessy, T. Regan, A process algebra for timed systems, Inform. and Comput. 117 (1995) 221–239.
[25] H. Hermanns, M. Rettelbach, Syntax, semantics, equivalences, and axioms for MTIPP, in: [28], pp. 71–87.
[26] H. Hermanns, M. Rettelbach, Towards a Superset of Basic LOTOS for Performance Prediction, in: [53], pp. 77–94.
[27] H. Hermanns, M. Rettelbach, T. Weiß, Formal characterisation of immediate actions in SPA with nondeterministic branching, in: [18], pp. 530–541.
[28] U. Herzog, M. Rettelbach (Eds.), Proc. 2nd Workshop on Process Algebras and Performance Modelling (PAPM '94), Erlangen, Germany, 1994.
[29] J. Hillston, A Compositional Approach to Performance Modelling, Cambridge Univ. Press, Cambridge, 1996.
[30] J. Hillston, The Nature of synchronisation, in: [28], pp. 51–70.
[31] C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[32] C.-C. Jou, S.A. Smolka, Equivalences, congruences, and complete axiomatizations for probabilistic processes, Proc. 1st Int. Conf. on Concurrency Theory (CONCUR '90), Amsterdam, The Netherlands, Lecture Notes in Computer Science, vol. 458, Springer, Berlin, 1990, pp. 367–383.
[33] J.-P. Katoen, E. Brinksma, R. Langerak, D. Latella, Stochastic simulation of event structures, in: [53], pp. 21–40.
[34] L. Kleinrock, Queueing Systems, Wiley, New York, 1975.
[35] K.G. Larsen, A. Skou, Bisimulation through probabilistic testing, Inform. and Comput. 94 (1991) 1–28.
[36] K.G. Larsen, A. Skou, Compositional verification of probabilistic processes, Proc. 3rd Int. Conf. Concurrency Theory (CONCUR '92), Stony Brook, NY, Lecture Notes in Computer Science, vol. 630, Springer, Berlin, 1992, pp. 456–471.

[37] D. Lehmann, M. Rabin, On the advantage of free choice: A symmetric and fully distributed solution to the dining philosophers problem, in Proc. 8th Symp. on Principles of Programming Languages (POPL '81), ACM Press, New York, 1981, pp. 133–138.
[38] G. Lowe, Probabilistic and prioritized models of timed CSP, Theoret. Comput. Sci. 138 (1995) 315–352.
[39] C. Miguel, A. Fernandez, L. Vidaller, LOTOS extended with probabilistic behaviours, Formal Aspects Comput. 5 (1993) 253–281.
[40] R. Milner, Communication and Concurrency, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[41] F. Moller, C. Tofts, A Temporal Calculus of Communicating Systems, Proc. 1st Int. Conf. on Concurrency Theory (CONCUR '90), Amsterdam, The Netherlands, Lecture Notes in Computer Science, vol. 458, Springer, Berlin, 1990, pp. 401–415.
[42] M.F. Neuts, Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach, John Hopkins Univ. Press, Baltimore City, 1981.
[43] X. Nicollin, J. Sifakis, The algebra of timed processes ATP: Theory and application, Inform. and Comput. 114 (1994) 131–178.
[44] N. Nounou, Y. Yemini, Algebraic specification-based performance analysis of communication protocols, Proc. 5th Int. Conf. Protocol Specification, Testing and Verification (PSTV V), North-Holland, Amsterdam, 1985, pp. 541–560.
[45] R. Paige, R.E. Tarjan, Three partition refinement algorithms, SIAM J. Comput. 16 (1987) 973–989.
[46] D. Park, Concurrency and automata on infinite sequences, Proc. 5th GI-Conf. on Theoret. Comput. Sci., Lecture Notes in Computer Science, vol. 104, Springer, Berlin, 1981, pp. 167–183.
[47] C. Priami, Stochastic π-calculus, in: [18], pp. 578–589.
[48] C. Priami, Stochastic π-calculus with general distributions, in: [53], pp. 41–57.
[49] S. Purushothaman, P.A. Subrahmanyam, Reasoning about probabilistic behavior in concurrent systems, IEEE Trans. Software Engng. 13 (1987) 740–745.
[50] J. Quemada, D. de Frutos, A. Azcorra, TIC: A timed calculus for LOTOS, Formal Aspects Comput. 5 (1993) 212–252.
[51] G.M. Reed, A.W. Roscoe, A timed model for communicating sequential processes, Theoret. Comput. Sci. 58 (1988) 249–261.
[52] M. Rettelbach, Probabilistic branching in markovian process algebras, in: [18], pp. 590–599.
[53] M. Ribaudo (Ed.), Proc. 4th Workshop on Process Algebras and Performance Modelling (PAPM '96), CLUT, Torino, Italy, 1996.
[54] P. Schweitzer, Aggregation methods for large markov chains, in: Mathematical Computer Performance and Reliability, North-Holland, Amsterdam, 1984, pp. 275–286.
[55] R. Segala, Modeling and Verification of Randomized Distributed Real-Time Systems, Ph.D. Thesis, MIT, 1995.
[56] K. Seidel, Probabilistic communicating processes, Theoret. Comput. Sci. 152 (1995) 219–249.
[57] S.A. Smolka, B. Steffen, Priority as extremal probability, in: Proc. 1st Int. Conf. Concurrency Theory (CONCUR '90), Amsterdam, The Netherlands, Lecture Notes in Computer Science, vol. 458, Springer, Berlin, 1990, pp. 456–466.
[58] C. Tofts, Processes with probabilities, priority and time, Formal Aspects Comput. 6 (1994) 536–564.
[59] C.A. Vissers, G. Scollo, M. van Sinderen, E. Brinksma, Specification styles in distributed systems design and verification, Theoret. Comput. Sci. 89 (1991) 179–206.
[60] W. Yi, CCS + Time = An Interleaving Model for Real Time Systems, Proc. 18th Int. Coll. on Automata, Languages and Programming (ICALP '91), Madrid (Spain), Lecture Notes in Computer Science, vol. 510, Springer, Berlin, 1991, pp. 217–228.