

Rate-based Transition Systems for Stochastic Process Calculi^{*}

FULL VERSION

Rocco De Nicola¹, Diego Latella², Michele Loreti¹, and Mieke Massink²

Dipartimento di Sistemi e Informatica - Università di Firenze¹
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"- CNR²

Abstract. A variant of Rate Transition Systems (RTS), proposed by Klin and Sassone, is introduced and used as the basic model for defining stochastic behaviour of processes. The transition relation used in our variant associates to each process, for each action, the set of possible futures paired with a measure indicating their rates. We show how RTS can be used for providing the operational semantics of stochastic extensions of three classical formalisms, namely CSP, CCS and π -calculus. It is also shown that, in contrast with the original definition by Priami, our semantics for stochastic π -calculus guarantees associativity of parallel composition.

1 Introduction

A number of stochastic process algebras have been proposed in the last two decades with the aim of combining two very successful approaches to concurrent systems specification and analysis, namely Labeled Transition Systems (LTS) and Continuous Time Markov Chains (CTMC). Indeed, LTS have proved to be a very convenient framework for providing compositional semantics of languages for specifying large complex system and for the analysis of their *qualitative properties* of systems. CTMC have, instead, been used mainly in performance evaluation, and thus for the analysis of *quantitative properties* taking into account also aspects of both time and probability. Examples of stochastic process algebras include TIPP [8], PEPA [13], EMPA [2], stochastic π -calculus [22] and StoKlaim [6]. Semantics of these calculi have been given by variants of the Structured Operational Semantics (SOS) approach but, as noticed in [17], they are not based on any general framework for operational semantics descriptions of stochastic processes, and indeed differ substantially from one another. Moreover, due to the different underlying models, it is rather difficult to appreciate differences and similarities of such semantics.

The common feature of all the above mentioned approaches is that the actions used to label transitions are enriched with rates of exponentially distributed

^{*} Research partially funded by EU IP SENSORIA (contract n. 016004), EU project RESIST/FAERUS (IST-2006-026764), the CNR-RSTL project XXL, the FIRB-MUR project TOCALIT and by the PRIN PACO.

random variables (r.v.) characterising their duration. On the other hand, they differ for the way synchronization rates are determined, the actions performed by processes are counted, etc. . Although the same class of r.v. is assumed, i.e. exponentially distributed ones, we have that the underlying models and notions are significantly different, ranging, e.g. from multi relations for PEPA, to *proved transition systems* for stochastic π -calculus, to *unique rate names* for STOKLAIM.

In [17], a variant of Labelled Transition Systems is introduced, namely Rate Transition Systems (RTS), which is used for defining the stochastic semantics of process calculi. The main feature of RTS is that the transition relation is actually a function ρ associating a rate value in $\mathbb{R}_{\geq 0}$ to each *state-action-state* triple: $\rho(P, \alpha, Q) = \lambda > 0$ if and only if P evolves via action α to Q with rate λ . Stochastic semantics of process calculi are defined by relying on the general framework of SGSOS. Moreover, in [17] they give conditions for guaranteeing associativity of parallel composition operator in the SGSOS framework. A consequence of this result is that one cannot guarantee the associativity of parallel composition operator up to stochastic bisimilarity when the synchronisation paradigm of CCS is used in combination with the synchronisation rate computation based on *apparent rates* [13]. This implies for instance that Stochastic π suffers of that problem.

Notice that, associativity of parallel composition is a very desirable property in particular in the context of network and distributed systems, for instance in presence of dynamic process creation.

In the present paper, we introduce a variant of Rate Transition Systems (RTS) where the transition relation associates terms and actions to functions from terms to rates; any such function assigns to each term the rate with which it can be reached from the source of the transition via the action. Moreover, we adapt the *apparent rate* approach to calculi like CCS and π -calculus. This adaptation guarantees associativity and commutativity properties of parallel composition.

The stochastic semantics of the considered process algebras is defined by a transition relation \longrightarrow that associates to a given process P and a given transition label α a function denoted by $\mathcal{P}, \mathcal{Q}, \dots$ mapping each term into a non-negative real number. The reduction $P \xrightarrow{\alpha} \mathcal{P}$ has the following meaning:

- if $\mathcal{P}(Q) = v$, (*with* $v \neq 0$), then Q is reachable from P by executing α , the duration of such execution being exponentially distributed with rate v ;
- if $\mathcal{P}(Q) = 0$, then Q is not reachable from P via α

Moreover, we have that if $P \xrightarrow{\alpha} \mathcal{P}$ then $\oplus \mathcal{P} \stackrel{def}{=} \sum_Q \mathcal{P}(Q)$ represents the total rate of α in P . The approach is somewhat reminiscent to that of Deng et al. [7] where probabilistic process algebra terms are associated to a discrete probability distribution over such terms.

In the rest of the paper, after introducing Rate Transition Systems, we show how they can be used for providing the stochastic operational semantics of three classical formalisms, namely CSP, CCS and π -calculus. We prove that our characterizations of the stochastic variants of the above mentioned process calculi

either are in full agreement with the originally proposed ones or show the differences. Furthermore, we show in our approach associativity of the parallel composition operator can be guaranteed also in the stochastic extensions of CCS and π -calculus. We also introduce a natural notion of bisimulation over RTS that is finer than Markovian bisimulation and use it to establish the associativity results. Due to space limitation, all proofs are omitted from the main body of the paper. The most relevant, less obvious, ones are reported in the appendix.

2 Rate transitions systems

The semantics of process algebras is classically described by means of Labelled Transitions Systems (LTS). These consist of a set of states, a set of transition labels and a transition relation. States correspond to the configurations systems can reach. Labels describe the actions systems can perform internally or that are used to interact with the environment. Transition relations describe system evolution as determined by the execution of specific actions.

The semantics of stochastic process algebras [11, 14] are classically defined by means of Continuous Time Markov Chains (CTMC), one of the most popular models for the evaluation of performance and dependability aspects of information processing systems. CTMC are in turn based on the notion of exponentially distributed r.v. [18, 9].

Definition 1. *A positive real-valued r.v. X is exponentially distributed with rate $\lambda \in \mathbb{R}$, with $\lambda > 0$, if the probability of X being at most t , i.e. $\text{Prob}(X \leq t)$, is $1 - e^{-\lambda t}$ if $t \geq 0$ and is 0 otherwise, where t is a real number. The expected value of X is $\frac{1}{\lambda}$.*

Exponentially distributed random variables enjoy the *memoryless property*, i.e. $\text{Prob}(X > t + t' \mid X > t) = \text{Prob}(X > t')$, for $t, t' \geq 0$.

Definition 2. *A Continuous-Time Markov Chain (CTMC) is a tuple (S, \mathbf{R}) where S is a countable set of states and \mathbf{R} a rate matrix assigning non-negative values to pairs of states, such that for all, $s \in S$, $\sum_{s' \in S} \mathbf{R}[s, s']$ converges¹.*

Intuitively, (S, \mathbf{R}) models a stochastic process where, for any state $s \in S$, whenever $\sum_{s' \in S} \mathbf{R}[s, s'] > 0$, the probability to take an outgoing transition from s by (continuous) time t is $1 - e^{-\sum_{s' \in S} \mathbf{R}[s, s'] \cdot t}$, i.e. the s -residence time is exponentially distributed with rate $\sum_{s' \in S} \mathbf{R}[s, s']$, and the probability to take a transition from state s to state s' , given that s is left, is $\frac{\mathbf{R}(s, s')}{\sum_{s'' \in S} \mathbf{R}[s, s']}$. If $\sum_{s' \in S} \mathbf{R}[s, s'] = 0$, then s is said to be *absorbing*, i.e. if the process enters state s , it remains in s forever. In what follows, the rate matrix function \mathbf{R} of any CTMC (S, \mathbf{R}) is lifted to sets of states $C \subseteq S$ in the natural way: $\mathbf{R}[s, C] \stackrel{\text{def}}{=} \sum_{s' \in C} \mathbf{R}[s, s']$.

¹ Notice that this definition allows self loops in CTMC, i.e. $\mathbf{R}[s, s] > 0$ is allowed. We refer the reader to [1] for details.

2.1 Rate Transition Systems and Markov Chains

We now present *Rated transitions system* (RTS), a generalisation of LTS, specifically designed for describing stochastic behaviours of process algebras and instrumental to generate CMTS to be associated to given systems. RTS have been introduced in [17], however, in that work, a rate is associated to each transition, while in our approach the transition relation associates to each state and to each action a function mapping each state to a non negative real number. Formally:

Definition 3 (Rate Transition Systems). *A rate transition systems is a triple (S, A, \longrightarrow) where S is a set of states, A a set of transition labels, \longrightarrow a subset of $S \times A \times \Sigma_S$ and Σ_S is the set $[S \rightarrow \mathbb{R}_{\geq 0}]$ of (total) functions from S to $\mathbb{R}_{\geq 0}$.*

In the sequel RTS will be denoted by $\mathcal{R}, \mathcal{R}_1, \mathcal{R}', \dots$, while $\mathcal{P}, \mathcal{Q}, \mathcal{R}, \dots$ will range over the elements of Σ_S . Intuitively, $s_1 \xrightarrow{\alpha} \mathcal{P}$ and $\mathcal{P}(s_2) = v \in \mathbb{R}_{>0}$ means that s_2 is reachable from s_1 via the execution of α with rate v . On the other hand, $\mathcal{P}(s_1) = 0$ means that s_2 is not reachable from s_1 via α . Notice that the above definition, differently from the original one in [17], includes also *nondeterministic* systems where from a certain state the same actions can lead to different rate functions.

Notation 1. *In the sequel, we will use \emptyset to denote the constant function 0, while $[s_1 \mapsto v_1, \dots, s_n \mapsto v_n]$ will denote a function associating v_i to s_i and 0 to all the other states. Moreover, if $\mathcal{P} \in \Sigma_S$ and $\mathcal{R} \subseteq S \times S$ is an equivalence relation, $\mathcal{P}_{/\mathcal{R}}$ denotes a function in $\Sigma_{[S]_{\mathcal{R}}} = [[S]_{\mathcal{R}} \rightarrow \mathbb{R}_{\geq 0}]$ such that:*

$$\mathcal{P}_{/\mathcal{R}}([s]_{\mathcal{R}}) = \sum_{s' \in [s]_{\mathcal{R}}} \mathcal{P}(s')$$

where $[S]_{\mathcal{R}}$ denotes the quotient of \mathcal{R} , and $[s]_{\mathcal{R}}$ the equivalence class of s .

Definition 4. *Let $\mathcal{R} = (S, A, \longrightarrow)$ be an RTS, then:*

- \mathcal{R} is fully stochastic if and only if for each $s \in S$, $\alpha \in A$, \mathcal{P} and \mathcal{Q} we have: $s \xrightarrow{\alpha} \mathcal{P}, s \xrightarrow{\alpha} \mathcal{Q} \implies \mathcal{P} = \mathcal{Q}$
- \mathcal{R} is image finite if and only if for each $s \in S$, $\alpha \in A$ and \mathcal{P} such that $s \xrightarrow{\alpha} \mathcal{P}$ we have: $\{s' \mid \mathcal{P}(s') > 0\}$ is finite

The following definition characterises the *quotient* RTS w.r.t. a given equivalence relation \mathcal{R} on states.

Definition 5. *Let $\mathcal{R} = (S, A, \longrightarrow)$ be an RTS and $\mathcal{R} \subseteq S \times S$ be an equivalence relation, then $[\mathcal{R}]_{\mathcal{R}}$ is the RTS $([S]_{\mathcal{R}}, A, \longrightarrow)$ where the transition relation is uniquely characterised, for all $s \in S$, by:*

$$[s]_{\mathcal{R}} \xrightarrow{\alpha} \mathcal{P}_{\mathcal{R}} \Leftrightarrow s \xrightarrow{\alpha} \mathcal{P}$$

In general, given RTS (S, A, \longrightarrow) we will be interested in the CTMC composed by the states reachable from a subset C of S only via the actions in $A' \subseteq A$. To that purpose we use the following two definitions:

Definition 6. For sets $C \subseteq S$ and $A' \subseteq A$, the set of derivatives of C through A' , denoted $Der(C, A')$, is the smallest set such that:

- $C \subseteq Der(C, A')$,
- if $s \in Der(C, A')$ and there exists $\alpha \in A'$ and $\mathcal{Q} \in \Sigma_S$ such that $s \xrightarrow{\alpha} \mathcal{Q}$ then $\{s' \mid \mathcal{Q}(s') > 0\} \subseteq Der(C, A')$

Definition 7. Let $\mathcal{R} = (S, A, \longrightarrow)$ be a fully stochastic RTS, for $C \subseteq S$, the CTMC of C , when one considers only actions $A' \subseteq A$ is defined as $CTMC[C, A'] \stackrel{def}{=} (Der(C, A'), \mathbf{R})$ where for all $s_1, s_2 \in Der(C, A')$:

$$\mathbf{R}[s_1, s_2] \stackrel{def}{=} \sum_{\alpha \in A'} \mathcal{P}^\alpha(s_2) \quad \text{with } s_1 \xrightarrow{\alpha} \mathcal{P}^\alpha.$$

Notice that, RTS which are not fully stochastic are naturally mapped to Continuous Time Markov Decision Processes [23].

2.2 Rate aware bisimulation

Two key concepts in the theory of process algebras are the notions of *behavioural equivalence and congruence* which permit identifying *different* terms corresponding to processes exhibiting *similar* behaviour. These notions have been very useful for formal reasoning about processes, for minimising process representations and for replacing equivalent components with “better” ones according to specific quality criteria. In the literature, many behavioural equivalences have been proposed which differ in what they consider the essential aspects of *observable* behaviour. More recently, such behavioural equivalences have been extended to *Markovian* process algebras where “better” would refer to better performance.

In this paper, we focus on *Strong Markovian Bisimulation Equivalence* [13], which has a direct correspondence with the notion of *lumpability*—a successful minimisation technique—of CTMCs [13, 16], and for which efficient algorithms have been devised for computing the best possible lumping [12]. We introduce *Rate Aware Bisimulation Equivalence* as the natural equivalence induced by the next state function and show that it implies Strong Markovian Bisimulation Equivalence. We point out that our semantic approach makes the definition of the Rate Aware Bisimulation Equivalence very natural.

Definition 8 (Strong Markovian bisimilarity [5]). Given a generic CTMC (S, \mathbf{R})

- An equivalence relation \mathcal{E} on S is a Markovian bisimulation on S if and only if for all $(s_1, s_2) \in \mathcal{E}$ and for all equivalence classes $C \in S_{/\mathcal{E}}$ the following condition holds: $\mathbf{R}[s_1, C] \leq \mathbf{R}[s_2, C]$.

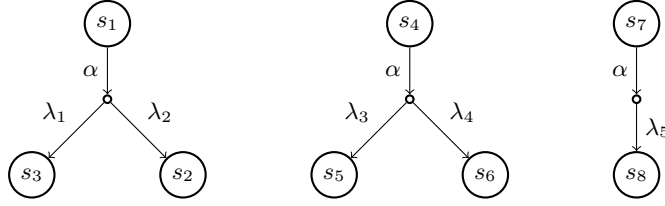


Fig. 1. $s_1 \sim s_4 \sim s_7$ when $\lambda_1 + \lambda_2 = \lambda_3 + \lambda_4 = \lambda_5$

- Two states $s_1, s_2 \in S$ are strong Markovian bisimilar, written $s_1 \sim_M s_2$, if and only if there exists a Markovian bisimulation \mathcal{E} on S with $(s_1, s_2) \in \mathcal{E}$.

Definition 9 (Rate Aware Bisimilarity).

- An equivalence relation \mathcal{E} on \mathcal{C} is a rate aware bisimulation if and only if, for all $(s_1, s_2) \in \mathcal{E}$, for all α and \mathcal{P} :

$$s_1 \xrightarrow{\alpha} \mathcal{P} \implies \exists \mathcal{Q} : s_2 \xrightarrow{\alpha} \mathcal{Q} \wedge \forall C \in \mathcal{C}_{/\mathcal{E}} \mathcal{P}(C) = \mathcal{Q}(C)$$

- Two states $s_1, s_2 \in S$ are rate aware bisimilar ($s_1 \sim s_2$) if there exists a rate aware bisimulation \mathcal{E} such that $(s_1, s_2) \in \mathcal{E}$.

States s_1, s_4 and s_7 of the RTS shown in Fig. 1² are rate aware bisimilar whenever $\lambda_1 + \lambda_2 = \lambda_3 + \lambda_4 = \lambda_5$.

Notice that *rate aware bisimilarity* and *strong bisimilarity* [19] coincide when one does not take rates into account, i.e. when the range of rate functions is $\{0, 1\}$. The following proposition guarantees that if two processes are *rate aware* equivalent, then the corresponding states in the generated CTMC are *strong Markovian equivalent*.

Proposition 1. Let $\mathcal{R} = (S, A, \longrightarrow)$, for each $A' \subseteq A$ and for each $s_1, s_2 \in S$ and $CTMC[\{s_1, s_2\}, A']$: $s_1 \sim s_2 \implies s_1 \sim_M s_2$

Notice that the reverse is not true. For example, with the reference of Figure 2, states s_1 and s_4 are Markovian equivalent in the $CTMC[\{s_1, s_4\}, \{\alpha\}]$, which does not contain states s_2 and s_5 , but $s_1 \not\sim s_4$.

3 PEPA: A Process Algebra for Performance Evaluation

The first process algebra we take into account is the Performance Evaluation Process Algebra (PEPA) developed by Hillston [13]. This algebra enriches CSP [15] with combinators useful for modeling performance related features.

² In the Fig. 1 we have $s_1 \xrightarrow{\alpha} [s_3 \mapsto \lambda_1, s_2 \mapsto \lambda_2]$, $s_4 \xrightarrow{\alpha} [s_5 \mapsto \lambda_3, s_6 \mapsto \lambda_4]$ and $s_7 \xrightarrow{\alpha} [s_8 \mapsto \lambda_5]$.

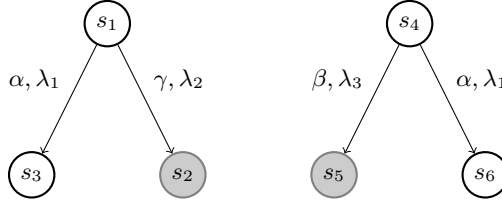


Fig. 2. $s_1 \sim_M s_4$ and $s_1 \not\sim s_4$

Like in CSP, in PEPA systems are described as interactions of *components* that may engage in *activities*. Components reflect the behaviour of relevant parts of the system, while activities capture the actions that the components perform. The specification of a PEPA activity consists of a pair (α, λ) in which *action* α symbolically denotes the performed action, while *rate* λ characterises the negative *exponential* distribution of its duration.

If \mathcal{A} is a set of *actions*, ranged over by $\alpha, \alpha', \alpha_1, \dots$, then \mathcal{P}_{PEPA} is the set of process terms P, P', P_1, \dots defined according to the following grammar

$$P ::= (\alpha, \lambda).P \mid P + P \mid P \bowtie_L P \mid P/L \mid A$$

where λ is a positive real number, L is a subset of \mathcal{A} and A is a *constant* which is assumed defined by an appropriate equation $A \triangleq P$ for some process term P , where *constants* occur only guarded in P , i.e. under the scope of an action prefix.

Component $(\alpha, \lambda).P$ models a process that perform action α and then behaves like P . The action duration is determined by a random variable exponentially distributed with rate λ .

Component $P + Q$ models a system that may behave either as P or as Q , representing a *race condition* between components. The cooperation operator $P \bowtie_L Q$ defines the set of action types L on which components P and Q must synchronise (or *cooperate*); both components proceed independently with any activity not occurring in L . The expected duration of a cooperation of activities $\alpha \in L$ is a function of the expected durations of the corresponding activities in the components. Roughly speaking, it corresponds to the longest one (the actual definition can be found in [13], where the interested reader can find all formal details of PEPA). Components P/L behaves as P except that activities in L are hidden and appearing as τ transitions. The behaviour of process variable A is that of P , provided that a definition $A \triangleq P$ is available for A .

We now provide the stochastic semantics of PEPA in terms of RTS. To this aim, we consider the RTS $\mathcal{R}_{PEPA} = (\mathcal{P}_{PEPA}, \mathcal{A}, \longrightarrow)$ where \longrightarrow is formally defined in Fig. 3. These rules permit deriving with a single proof *all* possible configurations reachable from a process with a given transition label.

Rule (ACT) states that $(\alpha, \lambda).P$ evolves with α to $[P \mapsto \lambda]$ (see Notation 1). Rule (\emptyset -ACT) states that no process is reachable from $(\alpha, \lambda).P$ by performing activity $\beta \neq \alpha$.

Rule (SUM) permits modeling stochastic behaviors of non deterministic choice. This rule states that the states reachable from $P + Q$ via α are all those that can be reached either by P or by Q . Moreover, transition rates are determined by summing local rates of transitions occurring either in P or in Q . Indeed, $\mathcal{P} + \mathcal{Q}$ denotes the next state function \mathcal{R} such that:

$$\mathcal{R}(R) = \mathcal{P}(R) + \mathcal{Q}(R)$$

For instance, if one considers process $((\alpha, \lambda_1).P_1 + (\beta, \lambda_2).P_2) + (\alpha, \lambda_3).P_3$ (where $\alpha \neq \beta$ and $P_1 \neq P_3$), the following derivation can be proved:

$$\frac{(\alpha, \lambda_1).P_1 \xrightarrow{\alpha} [P_1 \mapsto \lambda_1] \quad (\beta, \lambda_2).P_2 \xrightarrow{\alpha} \emptyset}{(\alpha, \lambda_1).P_1 + (\beta, \lambda_2).P_2 \xrightarrow{\alpha} [P_1 \mapsto \lambda_1] \quad (\alpha, \lambda_3).P_3 \xrightarrow{\alpha} [P_3 \mapsto \lambda_3]}}{((\alpha, \lambda_1).P_1 + (\beta, \lambda_2).P_2) + (\alpha, \lambda_3).P_3 \xrightarrow{\alpha} [P_1 \mapsto \lambda_1, P_3 \mapsto \lambda_3]}$$

Notice that, for applying rule (SUM) it is crucial to have rule (\emptyset -ACT). For instance, in the example above, we have applied rule (\emptyset -ACT) for proving the transition $(\beta, \lambda_2).P_2 \xrightarrow{\alpha} \emptyset$.

Rules (INT) and (COOP) govern cooperation. Rule (INT) states that if $\alpha \notin L$ computations of $P \bowtie_L Q$ are obtained by considering the interleaving of the transitions of P and Q . Hence, if we let \mathcal{P} and \mathcal{Q} be the next state functions of P and Q after α ($\alpha \notin L$), the next state function of $P \bowtie_L Q$ after α is obtained by combining $\mathcal{P} \bowtie_L Q$ and $P \bowtie_L \mathcal{Q}$, i.e. the next state function of P , composed with Q , and the next state function of Q , composed with P , respectively, as defined below.

Notation 2. For next state function \mathcal{P} , process algebra operator op and process Q we let $\mathcal{P} op Q$ (resp. $Q op \mathcal{P}$, $op \mathcal{P}$) be the function \mathcal{R} such that:

$$\mathcal{R}(R) = \begin{cases} \mathcal{P}(P) & R = P op Q \text{ (resp. } Q op P, op P) \\ 0 & \text{otherwise} \end{cases}$$

Rule (COOP) is used for computing the next state function when a synchronization between P and Q occurs. In that case, the next state function of $P \bowtie_L Q$ is determined as $\mathcal{P} \bowtie_L \mathcal{Q}$, as defined below.

Notation 3. For next state functions \mathcal{P} , \mathcal{Q} and set $L \subseteq \mathcal{A}$, $\mathcal{P} \bowtie_L \mathcal{Q}$ is the function such that:

$$\mathcal{P} \bowtie_L \mathcal{Q}(R) = \begin{cases} \mathcal{P}(P) \cdot \mathcal{Q}(Q) & R = P \bowtie_L Q \\ 0 & \text{otherwise} \end{cases}$$

As described in [13], actual rates in $\mathcal{P} \bowtie_L \mathcal{Q}$ are multiplied by the minimum of the apparent rate of α in P and Q and divided by their product. The apparent rates of α in a process P is defined as the total capacity of P to carry out activities of type α . In [13], the apparent rate of α in a process P is computed

$$\begin{array}{c}
\frac{}{(\alpha, \lambda).P \xrightarrow{\alpha} [P \mapsto \lambda]} \text{ (ACT)} \qquad \frac{\alpha \neq \beta}{(\alpha, \lambda).P \xrightarrow{\beta} \emptyset} \text{ (\emptyset-ACT)} \\
\frac{P \xrightarrow{\alpha} \mathcal{P} \quad Q \xrightarrow{\alpha} \mathcal{Q}}{P + Q \xrightarrow{\alpha} \mathcal{P} + \mathcal{Q}} \text{ (SUM)} \qquad \frac{P \xrightarrow{\alpha} \mathcal{P} \quad Q \xrightarrow{\alpha} \mathcal{Q} \quad \alpha \notin L}{P \boxtimes_L Q \xrightarrow{\alpha} (\mathcal{P} \boxtimes_L Q) + (P \boxtimes_L \mathcal{Q})} \text{ (INT)} \\
\frac{P \xrightarrow{\alpha} \mathcal{P} \quad Q \xrightarrow{\alpha} \mathcal{Q} \quad \alpha \in L}{P \boxtimes_L Q \xrightarrow{\alpha} \mathcal{P} \boxtimes_L \mathcal{Q} \cdot \frac{\min\{\oplus \mathcal{P}, \oplus \mathcal{Q}\}}{\oplus \mathcal{P} \cdot \oplus \mathcal{Q}}} \text{ (COOP)} \\
\frac{P \xrightarrow{\alpha} \mathcal{P} \quad \alpha \notin L}{P/L \xrightarrow{\alpha} \mathcal{P}/L} \text{ (P-HIDE)} \qquad \frac{\alpha \in L}{P/L \xrightarrow{\alpha} \emptyset} \text{ (\emptyset-HIDE)} \\
\frac{P \xrightarrow{\tau} \mathcal{P}_\tau \quad \forall \alpha \in L. P \xrightarrow{\alpha} \mathcal{P}_\alpha}{P/L \xrightarrow{\tau} \mathcal{P}_\tau/L + \sum_{\alpha \in L} \mathcal{P}_\alpha/L} \text{ (HIDE)} \qquad \frac{P \xrightarrow{\alpha} \mathcal{P} \quad A \triangleq P}{A \xrightarrow{\alpha} \mathcal{P}} \text{ (CALL)}
\end{array}$$

Fig. 3. PEPA Operational Semantics Rules

by using an auxiliary function $r_\alpha(P)$. By using our RTS approach, if $P \xrightarrow{\alpha} \mathcal{P}$, then the apparent rate of α in P is determined as:

$$\oplus \mathcal{P} = \sum_Q \mathcal{P}(Q)$$

Rule (P-HIDE) states that the set of processes reachable from P/L with α is determined by the set of processes reachable from P with α . Rule (\emptyset -HIDE) states that no process is reachable from P/L with $\alpha \in L$. Rule (HIDE) states that the set of processes reachable from P/L with a τ is determined by the set of processes reachable from P with τ and by considering, for each α in L , the set of processes reachable from P with α .

Notice that $\forall \alpha \in L. P \xrightarrow{\alpha} \mathcal{P}_\alpha$ in the premises of rule (HIDE) denotes that to prove a transition one has to prove a transition for each $\alpha \in L$. Theorem 1 below guarantees the finiteness of the proposed semantics.

Theorem 1. \mathcal{R}_{PEPA} is fully stochastic and image finite.

In the sequel by \longrightarrow_{PEPA} we mean the transition relation defined in [13].

Theorem 2. For all $P, Q \in \mathcal{P}_{PEPA}$ and $\alpha \in \mathcal{A}$ the following holds:

$$P \xrightarrow{\alpha} \mathcal{P} \wedge \mathcal{P}(Q) = \lambda > 0 \Leftrightarrow P \xrightarrow{\alpha, \lambda}_{PEPA} Q$$

The RTS associated to PEPA processes can be used for associating to each process P a CTMC. This is obtained by considering $CTMC[\{P\}, \mathcal{A}]$ where \mathcal{A} is the set of all activities that process P can perform.

4 Stochastic CCS

The second stochastic process algebra we consider in this paper is a stochastic extension of the Calculus of Communicating System (CCS) [19]. Differently from CSP, where processes composed in parallel *cooperate* in a *multi-party* synchronization, in CCS parallel processes interact with each other by means of a *two-party* synchronisation.

In Stochastic CCS (StoCCS), *output actions* are equipped with a parameter (a *rate*, $\lambda \in \mathbb{R}^+$) characterising a random variable with a negative exponential distribution, modeling the duration of the action. *Input actions* are annotated with a *weight* ($\omega \in \mathbb{N}^+$): a positive integer that will be used for determining the probability that the specific input is selected when a complementary output is executed. This approach is inspired by the *passive actions* presented in [13].

Let \mathcal{C} be a set of channels ranged over by a, b, c, \dots , $\bar{\mathcal{C}}$ denotes the *co-names* of \mathcal{C} . Elements in $\bar{\mathcal{C}}$ are ranged over by $\bar{a}, \bar{b}, \bar{c}, \dots$. A synchronization between processes P and Q occurs when P *sends* a signal over channel (action \bar{a}) while Q *receives* a signal over the same channel (action a). The result of a synchronization is an *internal*, or *silent*, transition that is labeled τ . In StoCCS a synchronization over channel a is rendered by the label \overleftarrow{a} . The reasons of this choice will be clarified later. We let $\overleftarrow{\mathcal{C}}$ be $\{\overleftarrow{a} \mid a \in \mathcal{C}\}$. The set of labels \mathcal{L} is then $\mathcal{C} \cup \bar{\mathcal{C}} \cup \{\tau\} \cup \overleftarrow{\mathcal{C}}$, while its elements are ranged over by $\ell, \ell', \ell_1, \dots$.

\mathcal{P}_{CCS} is the set of *Stochastic CCS* process terms $P, P', P_1, Q, Q', Q_1 \dots$ defined according to the following grammar:

$$\begin{aligned} P, Q &::= \mathbf{0} \mid G \mid P|Q \mid P[f] \mid P \setminus L \mid A \\ G &::= a^\omega.P \mid \bar{a}^\lambda.P \mid G + G \end{aligned}$$

where $L \subseteq \mathcal{C}$ while f is a *renaming function*, i.e. a function in $\mathcal{L} \rightarrow \mathcal{L}$ such that $f(\bar{a}) = \overleftarrow{f(a)}$ and $f(\tau) = \tau$. A is a *constant* which is assumed being defined by a proper defining equation $A \triangleq P$ for some process term P , where each constant can occur only guarded in P . We shall assume that each process G never contains at the same time an input and an output action on the same channel. In other words, processes of the form $\bar{a}.P + a.Q$ are forbidden.

Action prefixing and non-deterministic choice have the same meaning as in PEPA. Process $P|Q$ models a system where P and Q proceed in parallel and interact with each other using the two-parties synchronisation described above. Restriction ($P \setminus L$) and renaming ($P[f]$) are respectively used for inhibiting interactions of P over channels in L and for renaming channels in P according to function f .

Following a similar approach as the one used for PEPA, we now define the stochastic semantics of StoCCS in term of RTS. We let $\mathcal{R}_{StoCCS} = (\mathcal{P}_{CCS}, \mathcal{L}, \longrightarrow)$, where \longrightarrow is formally defined in Fig. 4.

The proposed semantics follows the same approach used by Priami in [22] and makes use of the PEPA notions of active and passive actions.

$$\begin{array}{c}
\frac{}{a^\omega.P \xrightarrow{a} [P \mapsto \omega]} \text{ (IN)} \qquad \frac{\ell \neq a}{a^\omega.P \xrightarrow{\ell} \emptyset} \text{ (\emptyset-IN)} \\
\frac{}{\bar{a}^\lambda.P \xrightarrow{\bar{a}} [P \mapsto \lambda]} \text{ (OUT)} \qquad \frac{\ell \neq \bar{a}}{\bar{a}^\lambda.P \xrightarrow{\ell} \emptyset} \text{ (\emptyset-OUT)} \\
\frac{P \xrightarrow{\ell} \mathcal{P} \quad Q \xrightarrow{\ell} \mathcal{Q}}{P+Q \xrightarrow{\ell} \mathcal{P}+\mathcal{Q}} \text{ (SUM)} \qquad \frac{P \xrightarrow{\ell} \mathcal{P} \quad Q \xrightarrow{\ell} \mathcal{Q} \quad \ell \neq \overleftarrow{a}}{P|Q \xrightarrow{\ell} \mathcal{P}|Q+P|\mathcal{Q}} \text{ (INT)} \\
\frac{P \xrightarrow{\overleftarrow{a}} \mathcal{P} \quad P \xrightarrow{a} \mathcal{P}_i \quad P \xrightarrow{\bar{a}} \mathcal{P}_o \quad Q \xrightarrow{\overleftarrow{a}} \mathcal{Q} \quad Q \xrightarrow{a} \mathcal{Q}_i \quad Q \xrightarrow{\bar{a}} \mathcal{Q}_o}{P|Q \xrightarrow{\overleftarrow{a}} \mathcal{P}|Q+P|\mathcal{Q} + \frac{\mathcal{P}_i \cdot \mathcal{Q}_o}{\oplus \mathcal{P}_i} + \frac{\mathcal{P}_o \cdot \mathcal{Q}_i}{\oplus \mathcal{Q}_i}} \text{ (SYNC)} \\
\frac{}{\mathbf{0} \xrightarrow{\ell} \emptyset} \text{ (NIL)} \qquad \frac{\ell \in L}{P \setminus L \xrightarrow{\ell} \emptyset} \text{ (\emptyset-RES)} \\
\frac{P \xrightarrow{\ell} \mathcal{P} \quad \ell \notin L}{P \setminus L \xrightarrow{\ell} \mathcal{P} \setminus L} \text{ (P-RES)} \qquad \frac{P \xrightarrow{\tau} \mathcal{P}_\tau \quad \forall \ell \in L. P \xrightarrow{\overleftarrow{\ell}} \mathcal{P}_{\overleftarrow{\ell}}}{P \setminus L \xrightarrow{\tau} \mathcal{P}_\tau \setminus L + \sum_{\ell \in L} \mathcal{P}_{\overleftarrow{\ell}} \setminus L} \text{ (RES)} \\
\frac{\forall \ell : P \xrightarrow{\ell} \mathcal{P}_\ell}{P \xrightarrow{\beta} \sum_{\ell: f(\ell)=\beta} \mathcal{P}_\ell[f]} \text{ (REN)} \qquad \frac{A \triangleq P \quad P \xrightarrow{\ell} \mathcal{P}}{A \xrightarrow{\ell} \mathcal{P}} \text{ (CALL)}
\end{array}$$

Fig. 4. StoCCS Operational Semantics

All the rules have the expected meaning and are similar to those defined for PEPA and simply render the CCS semantics in a context where all the possible next processes are computed in a single derivation.

More attention has to be paid to rule (SYNC) that is used for deriving synchronisations of parallel processes. In PEPA we have *multi-party* synchronisations. Hence, the next states of $P \boxtimes_L Q$ after $\ell \in L$ can be simply obtained by combining the possible next states of P and Q after ℓ . In CCS we have two-party synchronisations, thus the next states of $P|Q$ after \overleftarrow{a} , i.e. after a synchronisation over channel a , are:

1. the next states of P alone after \overleftarrow{a} , in parallel with Q ;
2. the next states of Q alone after \overleftarrow{a} , in parallel with P ;
3. the next states of P after \bar{a} in parallel with the next states of Q after a ;
4. the next states of P after a in parallel with the next states of Q after \bar{a} .

Moreover, synchronisation rates between inputs in P and outputs in Q (and vice-versa) are obtained by multiplying the input weights of P , i.e. \mathcal{P}_i , by the output rates of Q , i.e. \mathcal{Q}_o , over the total weight of all the inputs in P , i.e. $\oplus \mathcal{P}_i$ (and vice-versa).

As an example, consider $P \triangleq \bar{a}^2.P_1$ and $Q \triangleq a^4.Q_1|a^2.Q_2$, then we have that \overleftarrow{a} leads process $P|Q$ to $P_1|(Q_1|a^2.Q_2)$ with rate $\frac{4}{3}$ and to $P_1|(a^4.Q_1|Q_2)$ with rate $\frac{2}{3}$.

Theorem 3. \mathcal{R}_{StoCCS} is fully stochastic and image finite.

Unfortunately, the proposed semantics, like in [17], does not respect a standard and expected property of the CCS parallel composition. Indeed, using the above semantics, this operator is not associative. For instance $\bar{a}^\lambda.P|(a^{\omega_1}.Q_1|a^{\omega_2}.Q_2)$ and $(\bar{a}^\lambda.P|a^{\omega_1}.Q_1)|a^{\omega_2}.Q_2$ exhibit different stochastic behaviours. The former, after \overleftarrow{a} , reaches $P|(Q_1|a^{\omega_2}.Q_2)$ with rate $\frac{\lambda \cdot \omega_1}{\omega_1 + \omega_2}$ and $P|(a^{\omega_1}.Q_1|Q_2)$ with rate $\frac{\lambda \cdot \omega_2}{\omega_1 + \omega_2}$. The latter reaches both $(P|Q_1)|a^{\omega_2}.Q_2$ and $(P|a^{\omega_1}.Q_1)|Q_2$ with rate λ . From the results in [17] it follows that it is impossible to define an SGSOS semantics that guarantees the associativity of CCS parallel composition.

In the sequel, we show that this problem can be overcome by using our approach. To that purpose we modify rule (SYNC) in such way that:

- the rates of the synchronisations occurring in P and Q are updated in order to take into account the inputs available in both P and Q .
- the rates of the synchronisations between outputs in P and inputs in Q (and vice-versa) have to be divided by the *total* rate of *input* in both P and Q .

Rule (SYNC) can be reformulated as follows:

$$\frac{P \xrightarrow{\overleftarrow{a}} \mathcal{P} \quad P \xrightarrow{a} \mathcal{P}_i \quad P \xrightarrow{\bar{a}} \mathcal{P}_o \quad Q \xrightarrow{\overleftarrow{a}} \mathcal{Q} \quad Q \xrightarrow{a} \mathcal{Q}_i \quad Q \xrightarrow{\bar{a}} \mathcal{Q}_o}{P|Q \xrightarrow{\overleftarrow{a}} \frac{\mathcal{P}|\mathcal{Q} \oplus \mathcal{P}_i}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i} + \frac{P|\mathcal{Q} \oplus \mathcal{Q}_i}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i} + \frac{\mathcal{P}_i \cdot \mathcal{Q}_o}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i} + \frac{\mathcal{P}_o \cdot \mathcal{Q}_i}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i}}$$

Using this rule, the associativity of parallel composition is guaranteed.

Theorem 4. In *StoCCS* parallel composition is associative up to rate aware bisimilarity, i.e. for each P, Q and R , $P|(Q|R) \sim (P|Q)|R$

Notice that this result is not in contradiction with the one presented in [17] where it is proved that associativity of parallel composition does not hold if one uses PEPA-like synchronisation rates for CCS. Indeed, our result is obtained thanks to the use of a specific explicit label for synchronisations transitions (\overleftarrow{a}) that in [17] are labelled by τ . Our choice permits updating synchronisation rates while taking into account possible new inputs popping up along the derivation. Notice finally that, it is easy to prove that \sim is a congruence for each operator of *StoCCS*.

The CTMC associated to a *StoCCS* process P is obtained by considering $CTMC[\{P\}, \overleftarrow{\mathcal{C}} \cup \{\tau\}]$.

5 Stochastic π -calculus

In this section we consider a stochastic extension of π -calculus [20]. π -calculus is an algebra that extends CCS in such a way that *names* can be exchanged over channels. Like in CCS, in the π -calculus processes are composed by means of non-deterministic choice and parallel composition. However, restriction and

	fn	bn
0	\emptyset	\emptyset
$a^\omega(x).P$	$\{a\} \cup \text{fn}(P) - \{x\}$	$\{x\} \cup \text{bn}(P)$
$\bar{a}^\lambda(b).P$	$\{a, b\} \cup \text{fn}(P)$	$\text{bn}(P)$
$\tau^\lambda.P$	$\text{fn}(P)$	$\text{bn}(P)$
$G_1 + G_2$	$\text{fn}(G_1) \cup \text{fn}(G_2)$	$\text{bn}(G_1) \cup \text{bn}(G_2)$
$P + Q$	$\text{fn}(P) \cup \text{fn}(Q)$	$\text{bn}(P) \cup \text{bn}(Q)$

Fig. 5. Process free and bound names

renaming are replaced by *name restriction* $((\nu a)P)$. This is used for declaring a name ‘ a ’, that is *new*, or *private*, within process P .

Like for Stochastic CCS, in our proposal for the Stochastic π -calculus *output actions* are equipped with a *rate* while *input actions* are annotated with a *weight*.

Let \mathcal{N} be the set of names ranged over by a, b, \dots, x, y, \dots , we let $\mathcal{P}_{S\pi}$ be the set of process terms defined by the following syntax:

$$\begin{aligned}
P, Q &::= \mathbf{0} \mid G \mid P \mid Q \mid (\nu a)P \mid A \\
G &::= a^\omega.P \mid \bar{a}^\lambda.P \mid G + G
\end{aligned}$$

We assume that each process G never contains at the same time an input and an output action on the same channel; A is a *constant* which is assumed being defined by a proper defining equation $A \triangleq P$ for some process term P , where each constant can occur only guarded in P .

Let P be a process, $\text{fn}(P)$ and $\text{bn}(P)$ denote the set of *free* and *bound* names in P . Function $\text{fn}(P)$ and $\text{bn}(P)$ are formally defined in Fig. 5.

The semantics of Stochastic π -calculus processes is defined by means of RTS $\mathcal{R}_{Sto\pi} = (\mathcal{P}_{S\pi}, A, \longrightarrow)$ where \longrightarrow is formally defined by the rules in Fig. 6 while A is the set of transition labels α defined by means of the following syntax:

$$\alpha ::= \bar{a}b \mid \bar{a}(b) \mid ab \mid a(b) \mid \overleftarrow{a}(b) \mid \overleftarrow{a}(\bullet) \mid \tau$$

where $\bar{a}b$ (ab) denotes output (input) of name b over a ; $\bar{a}(b)$ ($a(b)$) denotes the output (input) of a private name b over a ; $\overleftarrow{a}(b)$ ($\overleftarrow{a}(\bullet)$) denotes a synchronisation over a where name b (a private name) has been exchanged. Indeed, to guarantee associativity of parallel composition, we have to make explicit the name exchanged in an interaction. However, if this name is private, we do not care about its exact value.

Let $\alpha \in A$; functions $\text{fn}(\alpha)$ and $\text{bn}(\alpha)$, denoting the free and bound names in α , are formally defined as follows: $\text{fn}(\bar{a}b) = \text{fn}(ab) = \text{fn}(\overleftarrow{a}(b)) = \{a, b\}$; $\text{fn}(\bar{a}(b)) = \text{fn}(a(b)) = \text{fn}(\overleftarrow{a}(\bullet)) = \{a\}$; $\text{fn}(\tau) = \text{bn}(\tau) = \emptyset$; $\text{bn}(\bar{a}b) = \text{bn}(ab) = \text{bn}(\overleftarrow{a}(b)) = \text{bn}(\overleftarrow{a}(\bullet)) = \emptyset$; $\text{bn}(\bar{a}(b)) = \text{bn}(a(b)) = \{b\}$. In the following we will use $n(\alpha)$ to denote $\text{fn}(\alpha) \cup \text{bn}(\alpha)$. We will also use $ch(\alpha)$ to denote the channel used in the action α .

The rules in Fig.6 extend those of CCS (Fig. 4) to take name passing into account. In these rules, an *early* approach is used [20]. Notice that, differently

$$\begin{array}{c}
\frac{}{\bar{a}b^\lambda.P \xrightarrow{\bar{a}b} [P \mapsto \lambda]} \text{ (OUT)} \\
\frac{}{a(x)^\omega.P \xrightarrow{ab} [P[b/x] \mapsto \omega]} \text{ (IN)} \\
\frac{b \notin \text{fn}(P)}{a(x)^\omega.P \xrightarrow{a(b)} [P[b/x] \mapsto \omega]} \text{ (B-IN)} \\
\frac{}{\mathbf{0} \xrightarrow{\alpha} \emptyset} \text{ (NIL)} \\
\frac{A \triangleq P \quad P \xrightarrow{\alpha} \mathcal{P}}{A \xrightarrow{\alpha} \mathcal{P}} \text{ (REC)} \\
\frac{P \xrightarrow{\alpha} \mathcal{P} \quad b \notin n(\alpha)}{(\nu b)P \xrightarrow{\alpha} (\nu b)\mathcal{P}} \text{ (P-RES)} \\
\frac{P \xrightarrow{\bar{a}(b)} \mathcal{P}_b \quad P \xrightarrow{\bar{a}(\bullet)} \mathcal{P}_\nu}{(\nu b)P \xrightarrow{\bar{a}(\bullet)} (\nu b)(\mathcal{P}_b + \mathcal{P}_\nu)} \text{ (H-RES2)} \\
\frac{P \xrightarrow{\bar{a}(b)} \mathcal{P} \quad P \xrightarrow{ab} \mathcal{P}_i \quad P \xrightarrow{\bar{a}b} \mathcal{P}_o \quad Q \xrightarrow{\bar{a}(b)} \mathcal{Q} \quad Q \xrightarrow{ab} \mathcal{Q}_i \quad Q \xrightarrow{\bar{a}b} \mathcal{Q}_o}{P|Q \xrightarrow{\bar{a}b} \frac{\mathcal{P}|\mathcal{Q} \oplus \mathcal{P}_i + P|\mathcal{Q} \oplus \mathcal{Q}_i + \mathcal{P}_i|\mathcal{Q}_o + \mathcal{P}_o|\mathcal{Q}_i}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i}} \text{ (SYNC)} \\
\frac{P \xrightarrow{\bar{a}(\bullet)} \mathcal{P} \quad P \xrightarrow{a(b)} \mathcal{P}_i \quad P \xrightarrow{\bar{a}(b)} \mathcal{P}_o \quad Q \xrightarrow{\bar{a}(\bullet)} \mathcal{Q} \quad Q \xrightarrow{a(b)} \mathcal{Q}_i \quad Q \xrightarrow{\bar{a}(b)} \mathcal{Q}_o}{P|Q \xrightarrow{\bar{a}(\bullet)} \frac{\mathcal{P}|\mathcal{Q} \oplus \mathcal{P}_i + P|\mathcal{Q} \oplus \mathcal{Q}_i + (\nu b)(\mathcal{P}_i|\mathcal{Q}_o) + (\nu b)(\mathcal{P}_o|\mathcal{Q}_i)}{\oplus \mathcal{P}_i \oplus \oplus \mathcal{Q}_i}} \text{ (R-SYNC)} \\
\frac{\alpha \neq \bar{a}(b) \quad \text{fn}(\bar{a}b^\lambda.P) \cap \text{bn}(\alpha) = \emptyset}{\bar{a}b^\lambda.P \xrightarrow{\alpha} \emptyset} \text{ (\emptyset-OUT)} \\
\frac{\alpha \neq ab \quad \alpha \neq a(b) \quad \text{fn}(a(x)^\omega.P) \cap \text{bn}(\alpha) = \emptyset}{a(x)^\omega.P \xrightarrow{\alpha} \emptyset} \text{ (\emptyset-IN)} \\
\frac{P_1 \xrightarrow{\alpha} \mathcal{P}_1 \quad P_2 \xrightarrow{\alpha} \mathcal{P}_2}{P_1 + P_2 \xrightarrow{\alpha} \mathcal{P}_1 + \mathcal{P}_2} \text{ (SUM)} \\
\frac{P_1 \xrightarrow{\alpha} \mathcal{P}_1 \quad P_2 \xrightarrow{\alpha} \mathcal{P}_2 \quad \alpha \neq \bar{a}}{P_1|P_2 \xrightarrow{\alpha} \mathcal{P}_1|P_2 + P_1|\mathcal{P}_2} \text{ (INT)} \\
\frac{P \xrightarrow{\bar{a}b} \mathcal{P} \quad b \neq a \quad c \notin \text{fn}(P)}{(\nu b)P \xrightarrow{\bar{a}(c)} \mathcal{P}[c/b]} \text{ (R-OPEN)} \\
\frac{P \xrightarrow{\tau} \mathcal{P} \quad \forall b.P \xrightarrow{\bar{a}(b)} \mathcal{P}_b \quad P \xrightarrow{\bar{a}(\bullet)} \mathcal{P}_\nu}{(\nu a)P \xrightarrow{\tau} (\nu a)\mathcal{P} + \sum_b (\nu a)\mathcal{P}_b + \mathcal{P}_\nu} \text{ (H-RES1)} \\
\frac{ch(\alpha) = b \vee (b \in n(\alpha) \wedge \alpha \neq \bar{a}b)}{(\nu b)P \xrightarrow{\alpha} \emptyset} \text{ (\emptyset-RES)}
\end{array}$$

Fig. 6. π -calculus Operational Semantics Rules

$$\begin{aligned}
P|(\nu n)Q &\equiv (\nu n)(P|Q) \text{ if } n \notin \text{fn}(P) \\
(\nu n)(\nu m)P &\equiv (\nu m)(\nu n)P \\
(\nu m)P &\equiv (\nu n)P[n/m] \text{ if } n \notin \text{fn}(P)
\end{aligned}$$

Fig. 7. Structural congruence laws.

from CCS, (\emptyset -IN) and (\emptyset -OUT) have a side condition introduced for avoiding the un-wanted binding of free names.

Moreover, *name restriction* hides interactions over restricted channels ((H-RES1)) and private names exchanged over public ones ((H-RES2)). Finally, rules (SYNC) and (R-SYNC) handle the synchronisations over channels when a public or a private name is exchanged, respectively.

Theorem 5. $\mathcal{R}_{Sto\pi}$ is image finite.

The proposed semantics is not *fully-stochastic*. Indeed, different names can be selected when private names are communicated. For instance, $(\nu a)\bar{b}a^\lambda.P|ax^\omega.Q$ reduces with $\overleftarrow{a}(\bullet)$ to $[(\nu c)(P[c/a]|Q[c/x]) \mapsto \lambda]$ and to $[(\nu d)(P[d/a]|Q[d/x]) \mapsto \lambda]$. However, the next state functions only differ for the names of private channels. We let \equiv be the smallest equivalence relation satisfying the Rules in Fig. 7. This equivalence, permits identifying those terms that are equivalent *up-to* renaming of restricted channels. We can finally prove the following result:

Theorem 6. $[\mathcal{R}_{Sto\pi}]_{\equiv}$ is image finite and fully stochastic.

It is worth noticing that the semantics of [22] that does not guarantee associativity of parallel composition can be captured within our approach by replacing rule (SYNC) of Figure 6 with a rule similar to (SYNC) of Figure 4 used to model binary synchronization for CCS.

Finally, the CTMC associated to a process P is obtained by considering only the synchronisations occurring in P . Let $\mathcal{A} = \{\overleftarrow{a}(b)|a, b \in \mathcal{N}\} \cup \{\overleftarrow{a}(\bullet)|a \in \mathcal{N}\} \cup \{\tau\}$, the CTMC associated to P is $CTMC[\{P\}, \mathcal{A}]$.

6 Conclusions

We have introduced Rate Transition Systems and have used them as the basic model for defining stochastic behaviour of processes. The transition relation associates to each process, for each action, the set of possible futures paired with a measure indicating their rates. An evident pleasant characteristic of our approach is that all the target terms derivable from a specific source term and an action are computed in a single relation. Moreover, the incorporation of functions into transitions becomes compact and some extent simplifies the definition of the operational semantics. We have then shown how RTS can be used to provide the stochastic operational semantics of CSP, CCS and pi-calculus and have proved that our characterizations of the stochastic variants coincide with the originally

proposed ones or that the differences are only those needed to guarantee nicer properties of the considered semantics, such as associativity of parallel composition. We have also introduced a natural notion of bisimulation over RTS that is finer than Markovian bisimulation and useful for reasoning about stochastic behaviours.

Even if in the present paper we have considered a synchronisation mechanism implicitly based on *active* and *passive* actions, other synchronisation patterns proposed in the literature can be easily dealt with using our approach. For instance, one could associate proper rates both to output and input actions and define the synchronisation rate as a suitable function of such rates.

RTS can be easily adapted for expressing stochastic models other than CTMCs, e.g. Interactive Markov Chains [10]. One way for doing that is to extend the range of rate functions with a distinguish element models interactive transitions.

As future work, we plan to study the format of our rules and see whether we could get similar general results about bisimulation congruence as in [17]. We plan also to apply our framework to richer formalism for service oriented programming where quality of service and performance measures are key issues. Indeed, we have already started research in this direction within the Sensoria project by providing a Markovian semantics to CASPIS [3] to obtain what we called MARCASPIS [21] and for investigating the integration of probabilistic and time-stochastic extensions of Tuple Space based coordination languages [4]. Moreover we plan to consider different distribution functions than exponential ones for action durations and/or delays.

References

1. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.
2. M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoret. Comput. Sci.*, 202(1-2):1–54, 1998.
3. M. Boreale, R. Bruni, R. De Nicola, and M. Loreti. Sessions and pipelines for structured service programming. In G. Barthe and F. S. de Boer, editors, *Proc. of FMOODS'08*, volume 5051 of *Lecture Notes in Computer Science*, pages 19–38. Springer, 2008.
4. M. Bravetti, D. Latella, M. Loreti, M. Massink, and G. Zavattaro. Combining timed coordination primitives and probabilistic tuple spaces. In *Proceedings of TGC 2008*, Lecture Notes in Computer Science, Barcellona, Spain, 2008. Springer. To appear.
5. E. Brinksma and H. Hermanns. Process algebra and markov chains. In E. Brinksma, H. Hermanns, and J.-P. Katoen, editors, *Euro Summer School on Trends in Computer Science*, volume 2090 of *Lecture Notes in Computer Science*, pages 183–231. Springer, 2001.
6. R. De Nicola, J.-P. Katoen, D. Latella, M. Loreti, and M. Massink. Model checking mobile stochastic logic. *Theoretical Computer Science*, 382(1):42–70, 2007.

7. Y. Deng, R. van Glabbeek, M. Hennessy, C. Morgan, and C. Zhang. Characterising testing preorders for finite probabilistic processes. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*, volume 313-325. IEEE Computer Society, 2007.
8. N. Glotz, U. Herzog, and M. Rettelbach. Multiprocessor and distributed systems design: The integration of functional specification and performance analysis using stochastic process algebras. In L. Donatiello and R. Nelson, editors, *Performance Evaluation of Computer and Communication Systems. - Joint Tutorial Papers of Performance '93 and Sigmetrics '93*, volume 729 of *Lect. Notes in Comput. Sci.* Springer, 1993.
9. B. Haverkort. Markovian Models for Performance and Dependability Evaluation. In E. Brinksma, H. Hermanns, and J. Katoen, editors, *Lectures on Formal Methods and Performance Analysis*, volume 2090 of *Lect. Notes in Comput. Sci.*, pages 38–83. Springer, 2001.
10. H. Hermanns. *Interactive Markov Chains*. Springer, Berlin/New York, 2002. LNCS 2428.
11. H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theoret. Comput. Sci.*, 274(1-2):43–87, 2002.
12. H. Hermanns and M. Siegle. Bisimulation Algorithms for Stochastic Process Algebras and Their BDD-Based Implementation. In *ARTS, Lecture Notes in Computer Science*, pages 244–264. Springer, 1999.
13. J. Hillston. A compositional approach to performance modelling, 1996. Distinguished Dissertation in Computer Science. Cambridge University Press.
14. J. Hillston. Process algebras for quantitative analysis. In *IEEE Symposium on Logic in Computer Science*, pages 239–248. IEEE, Computer Society Press, 2005.
15. C. Hoare. *Communicating Sequential Processes*. Series in Computer Science. Prentice Hall, 1985.
16. J. Kemeny and J. Snell. *Finite Markov Chains*. Springer, 1976.
17. B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In *Proceedings of FOSSACS 2008*, volume 4968 of *Lecture Notes in Computer Science*. Springer, 2008.
18. V. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman & Hall, 1995.
19. R. Milner. *Communication and Concurrency*. Series in Computer Science. Prentice Hall, 1989.
20. R. Milner, J. Parrow, and J. Walker. A Calculus of Mobile Processes, I and II. *Information and Computation*, 100(1):1–40, 41–77, 1992.
21. R. D. Nicola, D. Latella, M. Loreti, and M. Massink. MARCASPIS: a markovian extension of a calculus for services. In *Proceedings of SOS 2008*, Electronic Notes In Theoretical Computer Science, Reykjavik, Iceland, 2008. Elsevier.
22. C. Priami. Stochastic π -Calculus. *The Computer Journal*, 38(7):578–589, 1995.
23. M. Puterman. *Markov Decision Processes*. 1994.

A Rate aware bisimulation

Definition 10. Let $R \subseteq X \times X$, R^T is the transitive closure of X .

Lemma 1. Let $R_1, R_2 \subseteq X \times X$ be equivalence relations, then $(R_1 \cup R_2)^T$:

1. is an equivalence relation;
2. for each $C \in X/R$ there exists $\mathcal{A} \subseteq X/R_1$ and $\mathcal{B} \subseteq X/R_2$ such that:

$$C = \bigcup_{A \in \mathcal{A}} A \quad C = \bigcup_{B \in \mathcal{B}} B$$

3. for each R' such that $R_1 \subseteq R'$ and $R_2 \subseteq R'$: $R \subseteq R'$.

Proof.

1. It is easy to prove that $R = (R_1 \cup R_2)^T$ is an equivalence relation. Indeed R is reflexive, symmetric and transitive.
2. We have to prove that each equivalence class C of R in X can be partitioned into equivalence classes of R_1 (resp. R_2) in X . Formally: for each $C \in X/R$ there exists $\mathcal{A} \subseteq X/R_1$ (resp. $\mathcal{B} \subseteq X/R_2$) such that:

$$C = \bigcup_{A \in \mathcal{A}} A \quad (\text{resp. } C = \bigcup_{B \in \mathcal{B}} B)$$

Let $C \in X/R$. For each $x \in C$ there exists $A \in X/R_1$ such that $x \in A$. We let:

$$\{A \mid A \in X/R_1 : C \cap A_i \neq \emptyset\}$$

We have that $C \subseteq \bigcup_{A \in \mathcal{A}} A$ (it is clear that for each $A_1, A_2 \in \mathcal{A}$, if $A_1 \neq A_2$, $A_1 \cap A_2 = \emptyset$). Let us assume that there exists $A \in \mathcal{A}$ and $y \in A$ such that $y \notin C$. Let $x \in A \cap C$. Since A is an equivalence class for R_1 we have that for each $x \in A$, $(y, x) \in R_1$. Since $R_1 \subseteq R$ we have that $(y, x) \in R$. At the same time, C is an equivalence class of R . Hence, for each $a \in C$ if there exists $b \in X : (a, b) \in R$ then $b \in C$. Hence, $y \in C$. This contracts the hypothesis that there exists $A \in \mathcal{A}$ and $y \in A$ such that $y \notin C$ and \mathcal{A} is a partition of C .

3. Let $R' \subseteq X \times X$ be an equivalence relation such that $R_1 \subseteq R'$ and $R_2 \subseteq R'$. We have to prove that for each $(x, y) \in R \implies (x, y) \in R'$. Let $(x, y) \in R$, we can distinguish the following cases:
 - $(x, y) \in R_1$
 $\implies \{R_1 \subseteq R'\}$
 $(x, y) \in R'$
 - $(x, y) \in R_2$
 $\implies \{R_2 \subseteq R'\}$
 $(x, y) \in R'$

- there exist $a_0, \dots, a_{n+1} \in X$ such that: $x = a_0, y = a_{n+1}$ and for each $0 \leq j \leq n$ $(a_j, a_{j+1}) \in R_i$ ($i \in \{1, 2\}$):
 - $\implies \{R_1 \subseteq R' \text{ and } R_2 \subseteq R'\}$
 - $\forall j(0 \leq j \leq n)(a_j, a_{j+1}) \in R'$
 - $\implies \{R' \text{ is an equivalence relation, } x = a_0, \text{ and } y = a_{n+1}\}$
 - $(x, y) \in R'$

Lemma 2. Let R_1 and R_2 be rate aware bisimulations on $\mathcal{R} = (S, A, \rightarrow)$, the transitive closure of $R_1 \cup R_2$, $R = (R_1 \cup R_2)^T$ is a rate aware bisimulation.

Proof. We have to prove that for each $(x, y) \in R$:

$$x \xrightarrow{\alpha} \mathcal{P} \implies y \xrightarrow{\alpha} \mathcal{Q} \wedge \forall C \in S_{/R} \mathcal{P}(C) = \mathcal{Q}(C)$$

If $(x, y) \in R$ we can distinguish two cases:

1. $(x, y) \in R_i$ ($i \in \{1, 2\}$):

$$x \xrightarrow{\alpha} \mathcal{P}$$

$$\implies \{R_i \text{ is a rate aware bisimulation}\}$$

$$y \xrightarrow{\alpha} \mathcal{Q} \wedge \forall A \in S_{/R_i} \mathcal{P}(A) = \mathcal{Q}(A)$$

$$\implies \{\text{Lemma 1}\}$$

$$\forall C \in S_R : \exists \mathcal{A} \subseteq S_{/R_i} : \sum_{A \in \mathcal{A}} \mathcal{P}(A) = \sum_{A \in \mathcal{A}} \mathcal{Q}(A)$$

$$\implies \{\mathcal{A} \text{ is a partition of } C\}$$

$$\forall C \in S_{/R} \mathcal{P}(C) = \mathcal{Q}(C)$$

2. $(x, y) \in R$ and there exists x_0, \dots, x_{n+1} such that: $x = x_0, y = x_{n+1}$ and for each $0 \leq j \leq n$ $(x_j, x_{j+1}) \in R_i$ ($i \in \{1, 2\}$). For all j we have:

$$x_j \xrightarrow{\alpha} \mathcal{P}_j$$

$$\implies \{R_i \text{ is a rate aware bisimulation}\}$$

$$x_{j+1} \xrightarrow{\alpha} \mathcal{P}_{j+1} \wedge \forall A \in S_{/R_i} \mathcal{P}_j(A) = \mathcal{P}_{j+1}(A)$$

$$\implies \{\text{Lemma 1}\}$$

$$\forall C \in S_R : \exists \mathcal{A} \subseteq S_{/R_i} : \sum_{A \in \mathcal{A}} \mathcal{P}_j(A) = \sum_{A \in \mathcal{A}} \mathcal{P}_{j+1}(A)$$

$$\implies \{\mathcal{A} \text{ is a partition of } C\}$$

$$\forall C \in S_{/R} \mathcal{P}_j(C) = \mathcal{P}_{j+1}(C)$$

$$\implies \{\text{by transitivity}\}$$

$$\begin{aligned}
& \forall C \in S/R \mathcal{P}_0(C) = \mathcal{P}_{n+1}(C) \\
\implies & \{x = x_0, y = y_0\} \\
& x \xrightarrow{\alpha} \mathcal{P} \implies y \xrightarrow{\alpha} \mathcal{Q} \wedge \forall C \in S/R : \mathcal{P}(C) = \mathcal{Q}(C)
\end{aligned}$$

Hence, R is a *rate aware bisimulation*.

Corollary 1. $\sim = (\bigcup \{R \mid R \text{ is a rate aware bisimulation}\})^T$

B Proof Theorem 1

The theorem follows by proving that

$$\mathcal{E} = \{(s, s') \mid s, s' \in \text{Der}(\{s_1, s_2\}, A') : s \sim s'\}$$

is a Markovian bisimulation. Since \sim is an equivalence relation, \mathcal{E} is an equivalence too. Hence, we have to prove that for each $(x, y) \in \mathcal{E}$ and for each $C \in \text{Der}(\{s_1, s_2\}, A')/\mathcal{E}$: $\mathbf{R}[x, C] \leq \mathbf{R}[y, C]$.

Since, \mathcal{R} is *fully stochastic* it follows that for each s and a , one is able to identify a single $\mathcal{P}^{(s,a)}$ such that $s \xrightarrow{a} \mathcal{P}^{(s,a)}$.

$$(x, y) \in \mathcal{E}$$

\Rightarrow { \mathcal{E} is a rate aware bisimulation}

$$\forall a \in A' \forall C \subseteq S/\mathcal{E}. \mathcal{P}^{(x,a)}(C) = \mathcal{P}^{(y,a)}(C)$$

\equiv {Def. $\mathcal{P}(C)$ }

$$\forall a \in A' \forall C \subseteq S/\mathcal{E}. \sum_{s \in C} \mathcal{P}^{(x,a)}(s) = \sum_{s \in C} \mathcal{P}^{(y,a)}(s)$$

\equiv { $\forall i. x_i = y_i \implies \sum_i x_i = \sum_i y_i$ }

$$\forall C \subseteq S/\mathcal{E}. \sum_{a \in A'} \sum_{s \in C} \mathcal{P}^{(x,a)}(s) = \sum_{a \in A'} \sum_{s \in C} \mathcal{P}^{(y,a)}(s)$$

\equiv {Commutativity of +}

$$\forall C \subseteq S/\mathcal{E}. \sum_{s \in C} \sum_{a \in A'} \mathcal{P}^{(x,a)}(s) = \sum_{s \in C} \sum_{a \in A'} \mathcal{P}^{(y,a)}(s)$$

\equiv {Def. $\text{CTMC}[\{s_1, s_2\}, A']$ and \mathbf{R} }

$$\forall C \subseteq S/\mathcal{E}. \sum_{s \in C} \mathbf{R}[x, s] = \sum_{s \in C} \mathbf{R}[y, s]$$

\equiv {Def. $\mathbf{R}[s, C]$ }

$$\forall C \subseteq S/\mathcal{E}. \mathbf{R}[x, C] = \mathbf{R}[y, C]$$

\Rightarrow {Def. strong Markovian bisimulation}

\mathcal{E} is a strong Markovian bisimulation

C Proof Theorem 1

Image finiteness. We have to prove that for each P and α if $P \xrightarrow{\alpha} \mathcal{P}$ then $|\mathcal{P}| = |\{Q \mid \mathcal{P}(Q) > 0\}|$ is finite. The proof proceeds by induction on the syntax of P .

Base Cases:

$P = (\beta, \lambda).Q$: We can distinguish two cases.

1. $\alpha = \beta$

$$\Rightarrow \{\text{Def. } \longrightarrow\}$$

$$\mathcal{P} = [Q \mapsto \lambda]$$

$$\Rightarrow \{\text{Def. } [Q \mapsto \lambda]\}$$

$$|\mathcal{P}| = 1$$

2. $\alpha \neq \beta$

$$\Rightarrow \{\text{Def. } \longrightarrow\}$$

$$\mathcal{P} = \emptyset$$

$$\Rightarrow \{\text{Def. } \emptyset\}$$

$$|\emptyset| = 0$$

Inductive Hypothesis: Let P_1 and P_2 be such that for each α :

$$P_i \xrightarrow{\alpha} \mathcal{P} \implies |\mathcal{P}| \text{ is finite}$$

Inductive Step: According to the syntax of P we can distinguish the following cases:

$P = P_1 + P_2$:

$$\Rightarrow \{\text{Def. } \longrightarrow\}$$

$$\mathcal{P} = \mathcal{P}_1 + \mathcal{P}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\Rightarrow \{\text{Def. } \mathcal{P} + \mathcal{Q}\}$$

$$|\mathcal{P}| \leq |\mathcal{P}_1| + |\mathcal{P}_2|$$

$$\Rightarrow \{\text{Inductive Hypothesis}\}$$

$$|\mathcal{P}| \text{ is finite.}$$

$P = P_1 \boxtimes_L P_2$:

– $\alpha \in L$

$$\Rightarrow \{\text{Def. } \longrightarrow\}$$

$$\mathcal{P} = \mathcal{P}_1 \boxtimes_L \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \cdot \oplus \mathcal{P}_2} \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\Rightarrow \{\text{Def. } \mathcal{P} \boxtimes_L \mathcal{Q}\}$$

$$\begin{aligned}
& |\mathcal{P}| \leq |\mathcal{P}_1| \cdot |\mathcal{P}_2| \\
& \Rightarrow \quad \{\text{Inductive Hypothesis}\} \\
& \quad |\mathcal{P}| \text{ is finite.} \\
- \alpha \notin L \\
& \Rightarrow \quad \{\text{Def. } \longrightarrow \} \\
& \quad \mathcal{P} = \mathcal{P}_1 \boxtimes_L P_2 + P_1 \boxtimes_L \mathcal{P}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2 \\
& \Rightarrow \quad \{\text{Def. } \mathcal{P} + \mathcal{Q}\} \\
& \quad |\mathcal{P}| \leq |\mathcal{P}_1| + |\mathcal{P}_2| \\
& \Rightarrow \quad \{\text{Inductive Hypothesis}\} \\
& \quad |\mathcal{P}| \text{ is finite.}
\end{aligned}$$

$$\begin{aligned}
P = P_1/L: \\
- \alpha \in L \\
& \Rightarrow \quad \{\text{Def. } \longrightarrow \} \\
& \quad \mathcal{P} = \emptyset \\
& \Rightarrow \quad \{\text{Def. } \emptyset\} \\
& \quad |\mathcal{P}| = 0 \\
- \alpha \notin L \\
& \Rightarrow \quad \{\text{Def. } \longrightarrow \} \\
& \quad \mathcal{P} = \mathcal{P}_1/L \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \\
& \Rightarrow \quad \{\text{Def. } \mathcal{P}/L\} \\
& \quad |\mathcal{P}| = |\mathcal{P}_1| \\
& \Rightarrow \quad \{\text{Inductive Hypothesis}\} \\
& \quad |\mathcal{P}| \text{ is finite.}
\end{aligned}$$

$$\begin{aligned}
P = A: \text{ let } A \triangleq P_1 \\
& \Rightarrow \quad \{\text{Def. } \longrightarrow \} \\
& \quad \mathcal{P} = \mathcal{P}_1 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \\
& \Rightarrow \quad \{\text{Inductive Hypothesis}\} \\
& \quad \mathcal{P} \text{ is finite.}
\end{aligned}$$

Fully stochastic. We have to prove that for each P and α if $P \xrightarrow{\alpha} \mathcal{P}$ and $P \xrightarrow{\alpha} \mathcal{Q}$ then $\mathcal{P} = \mathcal{Q}$. The proof proceeds by induction on the syntax of P .

Base Cases:

$P = (\beta, \lambda).Q$: We can distinguish two cases.

1. $\alpha = \beta$

\Rightarrow {only rule (ACT) can be applied}

$$\mathcal{P} = [Q \mapsto \lambda] = \mathcal{Q}$$

2. $\alpha \neq \beta$

\Rightarrow {only rule (\emptyset -ACT) can be applied}

$$\mathcal{P} = \emptyset = \mathcal{Q}$$

Inductive Hypothesis: Let P_1 and P_2 be such that for each α :

$$P_i \xrightarrow{\alpha} \mathcal{P} \wedge P_i \xrightarrow{\alpha} \mathcal{Q} \implies \mathcal{P} = \mathcal{Q}$$

Inductive Step: We prove here $P = P_1 \boxtimes_L P_2$. The other cases are similar.

$P = P_1 \boxtimes_L P_2$:

- $\alpha \in L$

\Rightarrow {Def. \longrightarrow }

$$\mathcal{P} = \mathcal{P}_1 \boxtimes_L \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \oplus \mathcal{P}_2} \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\mathcal{Q} = \mathcal{Q}_1 \boxtimes_L \mathcal{Q}_2 \cdot \frac{\min\{\oplus \mathcal{Q}_1, \oplus \mathcal{Q}_2\}}{\oplus \mathcal{Q}_1 \oplus \mathcal{Q}_2} \text{ where } P_1 \xrightarrow{\alpha} \mathcal{Q}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{Q}_2$$

\Rightarrow {Inductive Hypothesis}

$$\mathcal{P}_1 = \mathcal{Q}_1, \mathcal{P}_2 = \mathcal{Q}_2$$

\Rightarrow {Def. $\mathcal{P} \boxtimes_L \mathcal{Q}$ }

$$\mathcal{P} = \mathcal{Q}$$

- $\alpha \notin L$

\Rightarrow {Def. \longrightarrow }

$$\mathcal{P} = \mathcal{P}_1 \boxtimes_L P_2 + P_1 \boxtimes_L \mathcal{P}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\mathcal{Q} = \mathcal{Q}_1 \boxtimes_L P_2 + P_1 \boxtimes_L \mathcal{Q}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{Q}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{Q}_2$$

\Rightarrow {Inductive Hypothesis}

$$\mathcal{P}_1 = \mathcal{Q}_1, \mathcal{P}_2 = \mathcal{Q}_2$$

\Rightarrow {Def. $\mathcal{P} + \mathcal{Q}$ }

$$\mathcal{P} = \mathcal{Q}$$

D Proof Theorem 2

In a multi-transition system the relation is replaced by a multi-relation in which the number of instances of a transition between states is recognised. In [13] operational semantics of PEPA is defined by means of the labelled multi-transition system induced by the following rules:

$$\begin{array}{c}
(\alpha, \lambda).P \xrightarrow{(\alpha, \lambda)} P \qquad \frac{P_1 \xrightarrow{(\alpha, \lambda)} P'}{P_1 + P_2 \xrightarrow{(\alpha, \lambda)} P'} \\
\\
\frac{P_2 \xrightarrow{(\alpha, \lambda)} P'}{P_1 + P_2 \xrightarrow{(\alpha, \lambda)} P'} \qquad \frac{P_1 \xrightarrow{(\alpha, \lambda)} P', \alpha \notin L}{P_1 \boxtimes_L P_2 \xrightarrow{(\alpha, \lambda)} P' \boxtimes_L P_2} \\
\\
\frac{P_1 \xrightarrow{(\alpha, \lambda_1)} P'_1 \quad P_2 \xrightarrow{(\alpha, \lambda_2)} P'_2 \quad \alpha \in L}{P_1 \boxtimes_L P_2 \xrightarrow{(\alpha, R)} P'_1 \boxtimes_L P'_2} \\
\\
\frac{P_1 \boxtimes_L P_2 \xrightarrow{(\alpha, \lambda)} P_1 \boxtimes_L P'}{P_2 \xrightarrow{(\alpha, \lambda)} P', \alpha \notin L} \qquad \frac{A \xrightarrow{(\alpha, \lambda)} P'}{P \xrightarrow{(\alpha, \lambda)} P', (A \stackrel{def}{=} P)}
\end{array}$$

In the rule for cooperation, with $\alpha \in L$, R stands for the following value:

$$R = \frac{r_1}{r_\alpha(P_1)} \cdot \frac{r_2}{r_\alpha(P_2)} \cdot \min(r_\alpha(P_1), r_\alpha(P_2))$$

where, for process P , $r_\alpha(P)$ denotes the *apparent rate* of α in P , i.e. the total capacity of P to carry out activities of type α :

1. $r_\alpha((\beta, \lambda).P) = \begin{cases} \lambda & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases}$
2. $r_\alpha(P + Q) = r_\alpha(P) + r_\alpha(Q)$
3. $r_\alpha(P/L) = \begin{cases} r_\alpha(P) & \text{if } \alpha \in L \\ 0 & \text{if } \alpha \notin L \end{cases}$
4. $r_\alpha(P \boxtimes_L Q) = \begin{cases} \min(r_\alpha(P), r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P), r_\alpha(Q) & \text{if } \alpha \notin L \end{cases}$

Notice that in [13] multi-transition system is not defined formally. For this reason, we cannot prove a direct correspondence between our semantics and the one proposed for PEPA. However, we prove that if $P \xrightarrow{\alpha} \mathcal{P}$ then:

1. $r_\alpha = \oplus \mathcal{P}$;
2. $\mathcal{P}(Q) > 0 \Leftrightarrow P \xrightarrow{(\alpha, r)} Q$.

1: $P \xrightarrow{\alpha} \mathcal{P} \Rightarrow r_\alpha = \oplus \mathcal{P}$: we proceed by induction on the syntax of P

Base Cases:

$P = (\beta, \lambda).Q$: We can distinguish two cases.

1. $\alpha = \beta$

$\Rightarrow \{ \text{Def. } \longrightarrow \}$

$\mathcal{P} = [Q \mapsto \lambda]$

$\Rightarrow \{ \text{Def. } [Q \mapsto \lambda] \}$

$\oplus \mathcal{P} = \lambda = r_\alpha(P)$

2. $\alpha \neq \beta$

$\Rightarrow \{ \text{Def. } \longrightarrow \}$

$\mathcal{P} = \emptyset$

$\Rightarrow \{ \text{Def. } \emptyset \}$

$\oplus \emptyset = 0 = r_\alpha(P)$

Inductive Hypothesis: Let P_1 and P_2 be such that for each α :

$$P_i \xrightarrow{\alpha} \mathcal{P} \implies \oplus \mathcal{P} = r_\alpha(P)$$

Inductive Step: According to the syntax of P we can distinguish the following cases:

$P = P_1 + P_2$:

$\Rightarrow \{ \text{Def. } \longrightarrow \}$

$\mathcal{P} = \mathcal{P}_1 + \mathcal{P}_2$ where $P_1 \xrightarrow{\alpha} \mathcal{P}_1$ and $P_2 \xrightarrow{\alpha} \mathcal{P}_2$

$\Rightarrow \{ \text{Def. } \mathcal{P} + \mathcal{Q} \}$

$\oplus \mathcal{P} = \oplus \mathcal{P}_1 + \oplus \mathcal{P}_2$

= {Inductive Hypothesis}

$r_\alpha(P_1) + r_\alpha(P_2) = r_\alpha(P)$

$P = P_1 \bowtie_L P_2$:

- $\alpha \in L$

$\Rightarrow \{ \text{Def. } \longrightarrow \}$

$\mathcal{P} = \mathcal{P}_1 \bowtie_L \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \cdot \oplus \mathcal{P}_2}$ where $P_1 \xrightarrow{\alpha} \mathcal{P}_1$ and $P_2 \xrightarrow{\alpha} \mathcal{P}_2$

$\Rightarrow \{ \text{Def. } \mathcal{P} \bowtie_L \mathcal{Q} \}$

$\oplus \mathcal{P} = \oplus \mathcal{P}_1 \cdot \oplus \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \cdot \oplus \mathcal{P}_2} = \min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}$

= {Inductive Hypothesis}

$$\begin{aligned}
& \min\{r_\alpha(P_1), r_\alpha(P_2)\} = r_\alpha(P_1 \boxtimes_L P_2) \\
- \alpha \notin L & \\
\Rightarrow & \quad \{\text{Def. } \longrightarrow \} \\
& \mathcal{P} = \mathcal{P}_1 \boxtimes_L P_2 + P_1 \boxtimes_L \mathcal{P}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2 \\
\Rightarrow & \quad \{\text{Def. } \mathcal{P} + \mathcal{Q}\} \\
& \oplus \mathcal{P} = \oplus \mathcal{P}_1 \oplus \mathcal{P}_2 \\
= & \quad \{\text{Inductive Hypothesis}\} \\
& r_\alpha(P_1) + r_\alpha(P_2) = r_\alpha(P_1 \boxtimes_L P_2)
\end{aligned}$$

$P = P_1/L$:

$$\begin{aligned}
- \alpha \in L & \\
\Rightarrow & \quad \{\text{Def. } \longrightarrow \} \\
& \mathcal{P} = \emptyset \\
\Rightarrow & \quad \{\text{Def. } \emptyset\} \\
& \oplus \mathcal{P} = 0 = r_\alpha(P) \\
- \alpha \notin L & \\
\Rightarrow & \quad \{\text{Def. } \longrightarrow \} \\
& \mathcal{P} = \mathcal{P}_1/L \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \\
\Rightarrow & \quad \{\text{Def. } \mathcal{P}/L\} \\
& \oplus \mathcal{P} = \oplus \mathcal{P}_1 \\
\Rightarrow & \quad \{\text{Inductive Hypothesis}\} \\
& r_\alpha(P_1) = r_\alpha(P)
\end{aligned}$$

2: $P \xrightarrow{\alpha} \mathcal{P} \Rightarrow \mathcal{P}(Q) \neq 0 \Leftrightarrow P \xrightarrow{\alpha, \lambda} Q$: we proceed by induction on the syntax of P

Base Cases:

$P = (\beta, \lambda).Q$: We can distinguish two cases.

$$\begin{aligned}
1. \alpha = \beta & \\
\Rightarrow & \quad \{\text{Def. } \longrightarrow \} \\
& \mathcal{P} = [Q \mapsto \lambda] \\
\Rightarrow & \quad \{\text{Def. } [Q \mapsto \lambda]\} \\
& \mathcal{P}(R) \neq 0 \Leftrightarrow R = Q
\end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \{ \text{Def. } \mapsto \} \\
&\quad P \xrightarrow{\alpha, \lambda} Q \\
2. \alpha \neq \beta \\
&\Rightarrow \{ \text{Def. } \longrightarrow \} \\
&\quad \mathcal{P} = \emptyset \\
&\Rightarrow \{ \text{Def. } \emptyset \} \\
&\quad \forall R. \mathcal{P}(R) = 0 \\
&\Leftrightarrow \{ \text{Def. } \mapsto \} \\
&\quad \exists Q. P \xrightarrow{\alpha, \lambda} Q
\end{aligned}$$

Inductive Hypothesis: Let P_1 and P_2 be such that for each α :

$$P_i \xrightarrow{\alpha} \mathcal{P} \implies \mathcal{P}(Q) \neq 0 \Leftrightarrow P \xrightarrow{\alpha, \lambda} Q$$

Inductive Step: We prove here $P = P_1 \boxtimes_L P_2$. The other cases are similar.

$P = P_1 \boxtimes_L P_2$:

– $\alpha \in L$

$$\Rightarrow \{ \text{Def. } \longrightarrow \}$$

$$\mathcal{P} = \mathcal{P}_1 \boxtimes_L \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \oplus \mathcal{P}_2} \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\Rightarrow \{ \text{Def. } \mathcal{P}_1 \boxtimes_L \mathcal{P}_2 \cdot \frac{\min\{\oplus \mathcal{P}_1, \oplus \mathcal{P}_2\}}{\oplus \mathcal{P}_1 \oplus \mathcal{P}_2} \}$$

$$\mathcal{P}(R) \neq 0 \Leftrightarrow R = R_1 \boxtimes_L R_2 \wedge \mathcal{P}_1(R_1) \neq 0 \wedge \mathcal{P}_1(R_2) \neq 0$$

$$\Leftrightarrow \{ \text{Inductive Hypothesis} \}$$

$$\mathcal{P}_i(Q) \neq 0 \Leftrightarrow P_i \xrightarrow{\alpha, \lambda} Q \quad i \in \{1, 2\}$$

$$\Leftrightarrow \{ \text{Def. } \mapsto \text{ and } \oplus \mathcal{P} = r_\alpha(P) \}$$

$$P \xrightarrow{\alpha, R} Q_1 \boxtimes_L Q_2$$

– $\alpha \notin L$

$$\Rightarrow \{ \text{Def. } \longrightarrow \}$$

$$\mathcal{P} = \mathcal{P}_1 \boxtimes_L P_2 + P_1 \boxtimes_L \mathcal{P}_2 \text{ where } P_1 \xrightarrow{\alpha} \mathcal{P}_1 \text{ and } P_2 \xrightarrow{\alpha} \mathcal{P}_2$$

$$\Leftrightarrow \{ \text{Inductive Hypothesis} \}$$

$$\mathcal{P}_i(Q) \neq 0 \Leftrightarrow P_i \xrightarrow{\alpha, \lambda} Q \quad i \in \{1, 2\}$$

$$\Leftrightarrow \{ \text{Def. } \mapsto \}$$

either $P \xrightarrow{\alpha, \lambda} Q \otimes_L P_2 \Leftrightarrow \mathcal{P}(Q \otimes_L P_2) \neq 0$
or $P \xrightarrow{\alpha, \lambda} P_1 \otimes_L Q \Leftrightarrow \mathcal{P}(P_1 \otimes_L Q) \neq 0$

E Proof Theorem 3

We have to prove that for each P and α

- if $P \xrightarrow{\ell} \mathcal{P}$ then $|\mathcal{P}| = |\{Q \mid \mathcal{P}(Q) > 0\}|$ is finite;
- if $P \xrightarrow{\ell} \mathcal{P}$ and $P \xrightarrow{\ell} \mathcal{Q}$ then $\mathcal{P} = \mathcal{Q}$.

The proof proceeds by induction on the syntax of P and it is similar to that for PEPA.

F Proof Theorem 4

The statement follows by proving that

$$\begin{aligned} \mathcal{E} = & \{ \langle P \mid (Q \mid R), (P \mid Q) \mid R \rangle \mid P, Q, R \in \mathcal{P}_{CCS} \} \\ & \cup \{ \langle (P \mid Q) \mid R, P \mid (Q \mid R) \rangle \mid P, Q, R \in \mathcal{P}_{CCS} \} \\ & \cup \{ \langle P \mid (Q \mid R), P \mid (Q \mid R) \rangle \mid P, Q, R \in \mathcal{P}_{CCS} \} \\ & \cup \{ \langle (P \mid Q) \mid R, (P \mid Q) \mid R \rangle \mid P, Q, R \in \mathcal{P}_{CCS} \} \end{aligned}$$

is a *rate aware* bisimulation in \mathcal{R}_{StoCCS} . It is easy to prove that \mathcal{E} is an equivalent relation. Indeed:

- for each P , $\langle P, P \rangle \in \mathcal{E}$;
- $\langle P, Q \rangle \in \mathcal{E} \Rightarrow \langle Q, P \rangle \in \mathcal{E}$
- $\langle P, Q \rangle \in \mathcal{E} \wedge \langle Q, R \rangle \in \mathcal{E} \Rightarrow \langle P, R \rangle \in \mathcal{E}$

We have to prove that if $\langle P, Q \rangle \in \mathcal{E}$ then for each $C \in \mathcal{P}_{CCS/\mathcal{E}}$:

$$P \xrightarrow{\alpha} \mathcal{P} \Longrightarrow Q \xrightarrow{\alpha} \mathcal{Q} \wedge \mathcal{P}(C) = \mathcal{Q}(C)$$

We can distinguish three cases:

- $P = Q$
 - $\Rightarrow \{ \mathcal{R}_{StoCCS} \text{ is fully stochastic} \}$
 - $P \xrightarrow{\alpha} \mathcal{P} \wedge Q \xrightarrow{\alpha} \mathcal{Q} \Longrightarrow \mathcal{P} = \mathcal{Q}$
 - $\Rightarrow \{ \text{Def. } \mathcal{P}(C) \}$
 - $\mathcal{P}(C) = \mathcal{Q}(C)$
- $P = (P_1 \mid P_2) \mid P_3$ and $Q = P_1 \mid (P_2 \mid P_3)$:
 - $\ell \neq \overleftarrow{a}$
 - $\Rightarrow \{ \text{Def. } \longrightarrow \}$
 - $\mathcal{P} = ((\mathcal{P}_1 \mid P_2) + (P_1 \mid \mathcal{P}_2)) \mid P_3 + (P_1 \mid P_2) \mid \mathcal{P}_3$ where $P_i \xrightarrow{\ell} \mathcal{P}_i$
 - $\Rightarrow \{ \text{Def. } \longrightarrow \}$

$$Q \xrightarrow{\ell} \mathcal{P}_1|(P_1|P_2) + P_1|(\mathcal{P}_2|P_3) + P_1|(P_2|\mathcal{P}_3)$$

$$\Rightarrow \{C/\mathcal{E} = \{X|(Y|Z), (X|Y)|Z\}\}$$

$$\mathcal{P}(C/\mathcal{E}) = \begin{cases} \mathcal{P}_1(X) & Y = P_2 \wedge Z = P_3 \\ \mathcal{P}_2(Y) & X = P_1 \wedge Z = P_3 \\ \mathcal{P}_3(Z) & X = P_1 \wedge Y = P_2 \end{cases}$$

$$\mathcal{Q}(C/\mathcal{E}) = \begin{cases} \mathcal{P}_1(X) & Y = P_2 \wedge Z = P_3 \\ \mathcal{P}_2(Y) & X = P_1 \wedge Z = P_3 \\ \mathcal{P}_3(Z) & X = P_1 \wedge Y = P_2 \end{cases}$$

$$\Rightarrow \mathcal{P}(C/\mathcal{E}) = \mathcal{Q}(C/\mathcal{E})$$

$$\bullet \ell = \overleftarrow{a}$$

$$\Rightarrow \{\text{Def.} \longrightarrow\}$$

$$\begin{aligned} \mathcal{P} &= \frac{\frac{\mathcal{P}_1|P_2 \cdot \oplus \mathcal{P}_1^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i} | P_3 \cdot (\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{\frac{P_1|\mathcal{P}_2 \cdot \oplus \mathcal{P}_2^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i} | P_3 \cdot (\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \\ &\frac{\frac{\mathcal{P}_1^i | \mathcal{P}^o}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i} | P_3 \cdot (\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{\frac{\mathcal{P}_1^o | \mathcal{P}_2^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i} | P_3 \cdot (\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \\ &\frac{(P_1|P_2)|\mathcal{P}_3 \cdot \oplus \mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \oplus \mathcal{P}_3^i} + \frac{(\mathcal{P}_1^i|P_2+P_1|\mathcal{P}_2^i)|\mathcal{P}_3^o}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{(\mathcal{P}_1^o|P_2+P_1|\mathcal{P}_2^o)|\mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} \\ &= \frac{((\mathcal{P}_1|P_2)|P_3) \cdot \oplus \mathcal{P}_1^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{((P_1|\mathcal{P}_2)|P_3) \cdot \oplus \mathcal{P}_2^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{(\mathcal{P}_1^i|\mathcal{P}^o)|P_3}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{(\mathcal{P}_1^o|\mathcal{P}_2^i)|P_3}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \\ &\frac{(P_1|P_2)|\mathcal{P}_3 \cdot \oplus \mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \oplus \mathcal{P}_3^i} + \frac{(\mathcal{P}_1^i|P_2+P_1|\mathcal{P}_2^i)|\mathcal{P}_3^o}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{(\mathcal{P}_1^o|P_2+P_1|\mathcal{P}_2^o)|\mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} \\ &= \frac{((\mathcal{P}_1|P_2)|P_3) \cdot \oplus \mathcal{P}_1^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{((P_1|\mathcal{P}_2)|P_3) \cdot \oplus \mathcal{P}_2^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{(\mathcal{P}_1^i|\mathcal{P}^o)|P_3}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{(\mathcal{P}_1^o|\mathcal{P}_2^i)|P_3}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \\ &\frac{(P_1|P_2)|\mathcal{P}_3 \cdot \oplus \mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \oplus \mathcal{P}_3^i} + \frac{(\mathcal{P}_1^i|P_2)|\mathcal{P}_3^o}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{(P_1|\mathcal{P}_2^i)|\mathcal{P}_3^o}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{(\mathcal{P}_1^o|P_2)|\mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} + \frac{(P_1|\mathcal{P}_2^o)|\mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} \end{aligned}$$

where:

$$P_j \xrightarrow{\overleftarrow{a}} \mathcal{P}_j, P_j^i \xrightarrow{a} \mathcal{P}_j^i, P_j \xrightarrow{\overline{a}} \mathcal{P}_j^o$$

$$\Rightarrow \{\text{Def.} \longrightarrow\}$$

$$\begin{aligned} \mathcal{Q} &= \frac{\mathcal{P}_1|(P_2|P_3) \cdot \oplus \mathcal{P}_1^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{P_1|(\mathcal{P}_2|P_3) \cdot \oplus \mathcal{P}_2^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{\mathcal{P}_1^i|(\mathcal{P}^o|P_3)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3} + \frac{\mathcal{P}_1^o|(\mathcal{P}_2^i|P_3)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \\ &\frac{P_1|(P_2|\mathcal{P}_3) \cdot \oplus \mathcal{P}_3^i}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \oplus \mathcal{P}_3^i} + \frac{\mathcal{P}_1^i|(P_2|\mathcal{P}_3^o)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{P_1|(\mathcal{P}_2^i|\mathcal{P}_3^o)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^o} + \frac{\mathcal{P}_1^o|(P_2|\mathcal{P}_3^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} + \frac{P_1|(\mathcal{P}_2^o|\mathcal{P}_3^i)}{\oplus \mathcal{P}_1^i \oplus \oplus \mathcal{P}_2^i \oplus \mathcal{P}_3^i} \end{aligned}$$

$$\Rightarrow \{C/\mathcal{E} = \{X|(Y|Z), (X|Y)|Z\}\}$$

$$\mathcal{P}(C/\mathcal{E}) = \mathcal{Q}(C/\mathcal{E})$$

– $P = X|(Y|Z)$ and $Q = (X|Y)|Z$: follows like the previous one.

G Proof Theorem 5

This theorem can be proved by induction on the syntax of $\text{Sto}\pi$ processes P following the same schema proposed for PEPA and StoCCS.

H Proof Theorem 6

The statement follows by proving, by induction on the syntax of P , that:

- if $P \xrightarrow{\alpha} \mathcal{P}, P \xrightarrow{\alpha} \mathcal{Q}$ and $\alpha \in \{\bar{a}b, ab, \overleftarrow{a}(b)\} \implies \mathcal{P} = \mathcal{Q}$;
- if $P \xrightarrow{a(b)} \mathcal{P}, P \xrightarrow{a(c)} \mathcal{Q}$ then $\mathcal{P}(Q) = \mathcal{Q}(Q[c/b])$;
- if $P \xrightarrow{\bar{a}(b)} \mathcal{P}$ and $P \xrightarrow{\bar{a}(c)} \mathcal{Q}$ then $\mathcal{P}(Q) = \mathcal{Q}(Q[c/b])$;
- if $P \xrightarrow{\alpha} \mathcal{P}, P \xrightarrow{\alpha} \mathcal{Q}$ and $\alpha \in \{\tau, \overleftarrow{a}(\bullet)\}$ then $\mathcal{P}/\equiv = \mathcal{Q}/\equiv$;