Rate-Based Transition Systems for Stochastic Process Calculi*

Rocco De Nicola[1], Diego Latella[2], Michele Loreti[1], and Mieke Massink[2]

[1] Dipartimento di Sistemi e Informatica - Università di Firenze
[2] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"- CNR

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 classical formalisms, namely CSP and CCS. We also show that our semantics for stochastic CCS guarantees associativity of parallel composition. Similarly, in contrast with the original definition by Priami, we argue that a semantics for stochastic π-calculus can be provided that guarantees associativity of parallel composition.

1 Introduction

Performance and dependability issues are of utmost importance for "network-aware" computing, due to the enormous size of systems—networks typically consist of thousands or even millions of nodes—and their strong dependence on mobility and interaction. Spontaneous computer crashes may easily lead to failure of remote execution or process movement, while spurious network failures may cause loss of code fragments or unpredictable delays. The enormous magnitude of computing devices involved in global computing yields failure rates that no longer can be ignored. The presence of such random phenomena implies that correctness of global computing software and their safety guarantees are no longer rigid notions.

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. CMTC have, instead, been used mainly in performance evaluation, and thus for the analysis of *quantitative properties* taking into account

* Research partially funded by EU IP SENSORIA (contract n. 016004), EU project RESIST/FAERUS (IST-2006-026764), the CNR-RSTL project XXL, the Italian national projects FIRB-MUR TOCAI.IT and PRIN PACO.

S. Albers et al. (Eds.): ICALP 2009, Part II, LNCS 5556, pp. 435–446, 2009.

also aspects related to both time and probability. Examples of stochastic process algebras include TIPP [8], PEPA [14], EMPA [2], stochastic π-calculus [22] and StoKlaim [5]. Semantics of these calculi have been provided by resorting to variants of the Structured Operational Semantics (SOS) approach but, as noticed in [18], 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 random variables (r.v.) characterising their mean duration. On the other hand, they differ for the way synchronization rates are determined, the actions performed by processes are counted, etc.. Moreover, 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 [18], 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 [18] conditions are put forward for guaranteeing associativity of the parallel composition operator in the SGSOS framework. It is then proved that one cannot guarantee 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* [14]. This implies for instance that parallel composition of Stochastic π is not associative. And, it has to be said that associativity of parallel composition is a higly desirable property in particular for networks and distributed systems, especially in presence of dynamic process creation.

In the present paper, we introduce a variant of RTS where the transition relation $\longrightarrow$ associates to a given process P and a given transition label α a function, denoted by $\mathscr{P}$, $\mathscr{Q}$,..., mapping each term into a non-negative real number. The reduction $P \xrightarrow{\alpha} \mathscr{P}$ has the following meaning: if $\mathscr{P}(Q) = v$, $(with\ v \neq 0)$, then Q is reachable from P by executing α, the duration of such execution being exponentially distributed with rate v; if $\mathscr{P}(Q) = 0$, then Q is not reachable from P via α. We have then that if $P \xrightarrow{\alpha} \mathscr{P}$ then $\oplus\mathscr{P} \stackrel{def}{=} \sum_Q \mathscr{P}(Q)$ represents the total rate of α in P. Moreover, we adapt the *apparent rate* approach to calculi like CCS and, consequently, π-calculus. This adaptation guarantees associativity and commutativity properties of parallel composition. The approach is somewhat reminiscent of 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 the two classical formalisms CSP and CCS. 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 that in our approach associativity of the parallel composition operator can be guaranteed also in the stochastic extensions of calculi based on a two party synchronisation pattern, like 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. For the same reason, the treatment of stochastic π-calculus is omitted; the complete RTS semantics, related results and proofs can be found in detail in [6].

2 Rate Transitions Systems

The semantics of process algebras is classically described by means of Labelled Transitions Systems (LTS). The semantics of stochastic process algebras [11,15] are classically defined by means of Continuous Time Markov Chains (CTMC). Here we assume the reader is familiar with basic notions concerning CTMC and exponentially distributed r.v. [9]; we only recall our working definition of CTMC:

Definition 1. *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* $s \in S$, $\sum_{s' \in S} \mathbf{R}[s, s']$ *converges*[1].

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: $\mathbf{R}[s, C] \stackrel{def}{=} \sum_{s' \in C} \mathbf{R}[s, s']$.

2.1 Rate Transition Systems and Markov Chains

We now present RTS, a generalisation of LTS, specifically designed for describing stochastic behaviours of process algebras and instrumental to generate CMTC to be associated to given systems. RTS have been introduced in [18], 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:

[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.

Definition 2 (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* $[S \to \mathbb{R}_{\geq 0}]$ *of total functions from* S *to* $\mathbb{R}_{\geq 0}$.

In the sequel RTS will be denoted by $\mathcal{R}$,$\mathcal{R}_1$,$\mathcal{R}'$,..., while $\mathscr{P}, \mathscr{Q}, \mathscr{R}, \ldots$ will range over the elements of Σ_S. Intuitively, $s_1 \xrightarrow{\alpha} \mathscr{P}$ and $\mathscr{P}(s_2) = v \in \mathbb{R}_{>0}$ means that s_2 is reachable from s_1 via the execution of α with rate v. On the other hand, $\mathscr{P}(s_2) = 0$ means that s_2 is not reachable from s_1 via α. Notice that the above definition, differently from the original one in [18], 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* $[s_1 \mapsto v_1, \ldots, s_n \mapsto v_n]$ *will denote a function associating* v_i *to* s_i *and* 0 *to all the other states. Moreover, for* $X \subseteq S$ *and* $\mathscr{P} \in \Sigma_S$ $\mathscr{P}(X) = \sum_{s \in X} \mathscr{P}(s)$ *and* $\oplus\mathscr{P}$ *denotes* $\mathscr{P}(S)$.

Definition 3. *Let* $\mathcal{R} = (S, A, \longrightarrow)$ *be an RTS, then:*

- $\mathcal{R}$ *is* well defined *if and only if for each* $s \in S$, $\alpha \in A$ *and* $\mathscr{P} \in \Sigma_S$ *such that* $s \xrightarrow{\alpha} \mathscr{P}$ *we have:* $\exists x : \oplus\mathscr{P} \leq x$
- $\mathcal{R}$ *is* image finite *if and only if for each* $s \in S$, $\alpha \in A$ *and* $\mathscr{P}$ *such that* $s \xrightarrow{\alpha} \mathscr{P}$ *either* $\mathscr{P} = \emptyset$ *or* $\mathscr{P} = [s_1 \mapsto \lambda_1, \ldots, s_n \mapsto \lambda_n]$
- $\mathcal{R}$ *is* fully stochastic *if and only if for each* $s \in S$, $\alpha \in A$, $\mathscr{P}$ *and* $\mathscr{Q}$ *we have:* $s \xrightarrow{\alpha} \mathscr{P}, s \xrightarrow{\alpha} \mathscr{Q} \Longrightarrow \mathscr{P} = \mathscr{Q}$

In the following we will only consider *well defined* RTS.

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 4. *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* $\mathscr{Q} \in \Sigma_S$ *such that* $s \xrightarrow{\alpha} \mathscr{Q}$ *then* $\{s' \mid \mathscr{Q}(s') > 0\} \subseteq Der(C, A')$

Definition 5. *Let* $\mathcal{R} = (S, A, \longrightarrow)$ *be a* fully stochastic *RTS, for* $C \subseteq S$*, the CTMC of* C*, when one considers only actions in a finite set* $A' \subseteq A$ *is defined as* $CTMC[C, A'] \stackrel{def}{=} (Der(C, A'), \mathbf{R})$ *where for all* $s_1, s_2 \in Der(C, A')$*:* $\mathbf{R}[s_1, s_2] \stackrel{def}{=} \sum_{\alpha \in A'} \mathscr{P}^{\alpha}(s_2)$ *with* $s_1 \xrightarrow{\alpha} \mathscr{P}^{\alpha}$.

Notice that RTS are naturally mapped to Continuous Time Markov Decision Processes [23,12]. Moreover, it turns out that general, non-fully stochastic, RTS are a convenient framework for automatic time bounded reachability probability analysis of Interactive Markov Chains [10,12], where nondeterminism and time are treated in an orthogonal way.

2.2 Rate Aware Bisimulation

Two key concepts in the theory of process algebras are the notions of *behavioural equivalence and congruence*. In the literature, many behavioural equivalences have been proposed which differ in what they consider essential aspects of *observable* behaviour. More recently, such behavioural equivalences have been extended to *Markovian* process algebras.

In this paper, we focus on *Strong Markovian Bisimulation Equivalence* [4,14], which has a direct correspondence with the notion of *lumpability*—a successful minimisation technique—of CTMCs [14,17], and for which efficient algorithms have been devised for computing the best possible lumping [13].

Definition 6 (Strong Markovian bisimilarity [4]). *Given 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:* $\mathbf{R}[s_1, C] \leq \mathbf{R}[s_2, C]$.
- *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* $(s_1, s_2) \in \mathcal{E}$.

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 7 (Rate Aware Bisimilarity). *Given RTS* $(S, A, \longrightarrow)$

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

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

- *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* $(s_1, s_2) \in \mathcal{E}$.

For instance, if we consider the RTS with set of states $\{s_i | 1 \leq i \leq 7\}$, where $s_1 \xrightarrow{\alpha} [s_3 \mapsto \lambda_1, s_2 \mapsto \lambda_2]$, $s_4 \xrightarrow{\alpha} [s_5 \mapsto \lambda_3, s_6 \mapsto \lambda_4]$ and $s_7 \xrightarrow{\alpha} [s_8 \mapsto \lambda_5]$, states s_1, s_4 and s_7 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* $\mathcal{R} = (S, A, \longrightarrow)$ *be a* fully stochastic *RTS, for each* $A' \subseteq A$ *and for each* $s_1, s_2 \in S$ *and* $CTMC[\{s_1, s_2\}, A']$*:* $s_1 \sim s_2 \Longrightarrow s_1 \sim_M s_2$

Notice that the reverse is not true. For example, if one considers RTS with states $\{s_i | 1 \leq i \leq 6\}$ where $s_1 \xrightarrow{\alpha} [s_3 \mapsto \lambda_1]$, $s_1 \xrightarrow{\gamma} [s_2 \mapsto \lambda_2]$, $s_4 \xrightarrow{\beta} [s_5 \mapsto \lambda_5]$, and $s_4 \xrightarrow{\alpha} [s_5 \mapsto \lambda_1]$, , 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 [14]. This algebra enriches CSP [16] with combinators useful for modeling performance related features.

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 $\alpha, \alpha', \alpha_1, \ldots$, then $\mathcal{P}_{PEPA}$ is the set of process terms $P, P', P_1, \ldots$ 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 $A \stackrel{\triangle}{=} P$ for some process term P, where *constants* occur only guarded in P, i.e. under the scope of a 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 [14], 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 $A \stackrel{\triangle}{=} P$ 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. 1. 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, $\mathscr{P} + \mathscr{Q}$ denotes the next state function $\mathscr{R}$ such that: $\mathscr{R}(R) = \mathscr{P}(R) + \mathscr{Q}(R)$.

$$\frac{}{(\alpha,\lambda).P \xrightarrow{\alpha} [P \mapsto \lambda]}\ (\textsc{Act}) \qquad \frac{\alpha \neq \beta}{(\alpha,\lambda).P \xrightarrow{\beta} \emptyset}\ (\emptyset\text{-}\textsc{Act})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q}}{P+Q \xrightarrow{\alpha} \mathscr{P}+\mathscr{Q}}\ (\textsc{Sum}) \qquad \frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \notin L}{P \bowtie_L Q \xrightarrow{\alpha} \mathscr{P} \bowtie_L Q + P \bowtie_L \mathscr{Q}}\ (\textsc{Int})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad Q \xrightarrow{\alpha} \mathscr{Q} \quad \alpha \in L}{P \bowtie_L Q \xrightarrow{\alpha} \mathscr{P} \bowtie_L \mathscr{Q} \cdot \frac{\min\{\oplus\mathscr{P}, \oplus\mathscr{Q}\}}{\oplus\mathscr{P} \cdot \oplus\mathscr{Q}}}\ (\textsc{Coop})$$

$$\frac{P \xrightarrow{\alpha} \mathscr{P} \quad \alpha \notin L}{P/L \xrightarrow{\alpha} \mathscr{P}/L}\ (\text{P-}\textsc{Hide}) \qquad \frac{\alpha \in L}{P/L \xrightarrow{\alpha} \emptyset}\ (\emptyset\text{-}\textsc{Hide})$$

$$\frac{P \xrightarrow{\tau} \mathscr{P}_\tau \quad \forall \alpha \in L.P \xrightarrow{\alpha} \mathscr{P}_\alpha}{P/L \xrightarrow{\tau} \mathscr{P}_\tau/L + \sum_{\alpha \in L} \mathscr{P}_\alpha/L}\ (\textsc{Hide}) \qquad \frac{P \xrightarrow{\alpha} \mathscr{P} \quad A \triangleq P}{A \xrightarrow{\alpha} \mathscr{P}}\ (\textsc{Call})$$

Fig. 1. PEPA Operational Semantics Rules

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 $\mathscr{P}$ and $\mathscr{Q}$ 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 $\mathscr{P} \bowtie_L Q$ and $P \bowtie_L \mathscr{Q}$, 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 $\mathscr{P}$, process algebra operator op and process Q we let $\mathscr{P}$ op Q (resp. Q op $\mathscr{P}$, op $\mathscr{P}$) be the function $\mathscr{R}$ such that $\mathscr{R}(R)$ is $\mathscr{P}(P)$ if $R = P$ op Q (resp. Q op P, op P) and 0 otherwise.*

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 $\mathscr{P} \bowtie_L \mathscr{Q}$, as defined below.

Notation 3. *For next state functions $\mathscr{P}$, $\mathscr{Q}$ and set $L \subseteq \mathcal{A}$, $\mathscr{P} \bowtie_L \mathscr{Q}$ is the function such that $\mathscr{P} \bowtie_L \mathscr{Q}(R)$ is $\mathscr{P}(P) \cdot \mathscr{Q}(Q)$ if $R = P \bowtie_L Q$, 0 otherwise.*

As described in [14], actual rates in $\mathscr{P} \bowtie_L \mathscr{Q}$ are multiplied by the minimum of the apparent rate of α in P and Q and divided by their product.

Notation 4. *For next state functions $\mathscr{P}$, and $x, y \in \mathbb{R}_{\geq 0}$ $\mathscr{P} \cdot \frac{x}{y}$ is the function $\mathscr{R}$ such that $\mathscr{R}(R) = \mathscr{P}(R) \cdot \frac{x}{y}$ if $y \neq 0$, $\emptyset$ otherwise.*

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 [14], the apparent rate of α in a process P is computed by using an auxiliary function $r_\alpha(P)$. By using our RTS approach, if $P \xrightarrow{\alpha} \mathscr{P}$, then the apparent rate of α in P is determined as: $\oplus\mathscr{P} = \sum_Q \mathscr{P}(Q)$.

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 $\forall \alpha \in L. P \xrightarrow{\alpha} \mathscr{P}_\alpha$ 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 [14].

Theorem 2. *For all $P, Q \in \mathcal{P}_{PEPA}$ and $\alpha \in \mathcal{A}$ the following holds: $P \xrightarrow{\alpha} \mathscr{P} \wedge \mathscr{P}(Q) = \lambda > 0$ if and only if $P \xrightarrow{\alpha,\lambda}_{PEPA} Q$.*

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 [14].

Let $\mathcal{C}$ be a set of channels ranged over by $a, b, c, \ldots$, $\overline{\mathcal{C}}$ denotes the *co-names* of $\mathcal{C}$. Elements in $\overline{\mathcal{C}}$ are ranged over by $\overline{a}, \overline{b}, \overline{c}, \ldots$. A synchronization between processes P and Q occurs when P *sends* a signal over channel (action $\overline{a}$) 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 $\overleftrightarrow{a}$. The reasons for this choice will be clarified later. We let $\overleftrightarrow{\mathcal{C}}$ be $\{\overleftrightarrow{a} | a \in \mathcal{C}\}$. The set of labels $\mathcal{L}$ is then $\mathcal{C} \cup \overline{\mathcal{C}} \cup \{\tau\} \cup \overleftrightarrow{\mathcal{C}}$, while its elements are ranged over by $\ell, \ell', \ell_1, \ldots$.

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

$$P, Q \ ::= \ \mathbf{0} \mid G \mid P|Q \mid P[f] \mid P\backslash L \mid A \qquad G \ ::= \ a^\omega.P \mid \overline{a}^\lambda.P \mid G + G$$

where $L \subseteq \mathcal{C}$ while f is a *renaming function*, i.e. a function in $\mathcal{L} \to \mathcal{L}$ such that $f(\overline{a}) = \overline{f(a)}$, $f(\overleftrightarrow{a}) = \overleftrightarrow{f(a)}$ and $f(\tau) = \tau$. A is a *constant* which is assumed being

$$\frac{}{a^\omega.P \xrightarrow{a} [P \mapsto \omega]}\ (\text{In}) \qquad \frac{\ell \neq a}{a^\omega.P \xrightarrow{\ell} \emptyset}\ (\emptyset\text{-In})$$

$$\frac{}{\overline{a}^\lambda.P \xrightarrow{\overline{a}} [P \mapsto \lambda]}\ (\text{Out}) \qquad \frac{\ell \neq \overline{a}}{\overline{a}^\lambda.P \xrightarrow{\ell} \emptyset}\ (\emptyset\text{-Out})$$

$$\frac{P \xrightarrow{\ell} \mathscr{P} \quad Q \xrightarrow{\ell} \mathscr{Q}}{P+Q \xrightarrow{\ell} \mathscr{P}+\mathscr{Q}}\ (\text{Sum}) \qquad \frac{P \xrightarrow{\ell} \mathscr{P} \quad Q \xrightarrow{\ell} \mathscr{Q} \quad \ell \neq \overleftarrow{a}}{P|Q \xrightarrow{\ell} \mathscr{P}|Q + P|\mathscr{Q}}\ (\text{Int})$$

$$\frac{P \xrightarrow{\overleftarrow{a}} \mathscr{P} \quad P \xrightarrow{a} \mathscr{P}_i \quad P \xrightarrow{\overline{a}} \mathscr{P}_o \quad Q \xrightarrow{\overleftarrow{a}} \mathscr{Q} \quad Q \xrightarrow{a} \mathscr{Q}_i \quad Q \xrightarrow{\overline{a}} \mathscr{Q}_o}{P|Q \xrightarrow{\overleftarrow{a}} \mathscr{P}|Q + P|\mathscr{Q} + \frac{\mathscr{P}_i \cdot \mathscr{Q}_o}{\oplus \mathscr{P}_i} + \frac{\mathscr{P}_o \cdot \mathscr{Q}_i}{\oplus \mathscr{Q}_i}}\ (\text{Sync})$$

$$\frac{}{\mathbf{0} \xrightarrow{\ell} \emptyset}\ (\text{Nil}) \qquad \frac{\ell \in L}{P\backslash L \xrightarrow{\ell} \emptyset}\ (\emptyset\text{-Res})$$

$$\frac{P \xrightarrow{\ell} \mathscr{P} \quad \ell \notin L}{P\backslash L \xrightarrow{\ell} \mathscr{P}\backslash L}\ (\text{P-Res}) \qquad \frac{P \xrightarrow{\tau} \mathscr{P}_\tau \quad \forall \ell \in L.P \xrightarrow{\overleftarrow{\ell}} \mathscr{P}_{\overleftarrow{\ell}}}{P\backslash L \xrightarrow{\tau} \mathscr{P}_\tau\backslash L + \sum_{\ell \in L} \mathscr{P}_{\overleftarrow{\ell}}\backslash L}\ (\text{Res})$$

$$\frac{\forall \ell : P \xrightarrow{\ell} \mathscr{P}_\ell}{P \xrightarrow{\beta} \sum_{\ell : f(\ell)=\beta} \mathscr{P}_\ell[f]}\ (\text{Ren}) \qquad \frac{A \triangleq P \quad P \xrightarrow{\ell} \mathscr{P}}{A \xrightarrow{\ell} \mathscr{P}}\ (\text{Call})$$

Fig. 2. StoCCS Operational Semantics

defined by a proper defining equation $A \stackrel{\triangle}{=} P$ for some process term P, where each constant can occur only guarded in P. For the sake of notational simplicity, we 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 $\overline{a}.P + a.Q$ are forbidden. This does not introduce a significant restriction because such processes do not have an obvious meaning in the context of stochastic process algebras.

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 ($P\backslash L$) 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. 2.

The proposed semantics follows the same approach used by Priami in [22] for the stochastic π-calculus and makes use of the PEPA notions of active and passive actions. 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 $P \bowtie_L Q$ 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 $\overleftrightarrow{a}$, i.e. after a synchronisation over channel a, are: (1) the next states of P alone after $\overleftrightarrow{a}$, in parallel with Q; (2) the next states of Q alone after $\overleftrightarrow{a}$, in parallel with P; (3) the next states of P after $\overline{a}$ 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 $\overline{a}$. 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. $\mathscr{P}_i$, by the output rates of Q, i.e. $\mathscr{Q}_o$, over the total weight of all the inputs in P, i.e. $\oplus \mathscr{P}_i$ (and vice-versa). As an example, consider $P \triangleq \overline{a}^2.P_1$ and $Q \triangleq a^4.Q_1|a^2.Q_2$, then we have that $\overleftrightarrow{a}$ 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.

It is easy to prove that the stochastic semantics of Fig. 2 coincides with the one proposed in [18]. Unfortunately, the proposed semantics, like in [18], 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 $\overline{a}^{\lambda}.P|(a^{\omega_1}.Q_1|a^{\omega_2}.Q_2)$ and $(\overline{a}^{\lambda}.P|a^{\omega_1}.Q_1)|a^{\omega_2}.Q_2$ exhibit different stochastic behaviours. The former, after $\overleftrightarrow{a}$, 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 [18] it follows that it is impossible to define an SGSOS semantics that guarantees the associativity of CCS parallel composition. It is moreover worth pointing out that the definition of the semantics for the stochastic π-calculus, as in [22], suffers of the same problem [18]. 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:

$$\frac{P \xrightarrow{\overleftrightarrow{a}} \mathscr{P} \quad P \xrightarrow{a} \mathscr{P}_i \quad P \xrightarrow{\overline{a}} \mathscr{P}_o \quad Q \xrightarrow{\overleftrightarrow{a}} \mathscr{Q} \quad Q \xrightarrow{a} \mathscr{Q}_i \quad Q \xrightarrow{\overline{a}} \mathscr{Q}_o}{P|Q \xrightarrow{\overleftrightarrow{a}} \frac{\mathscr{P}|Q \cdot \oplus \mathscr{P}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{P|\mathscr{Q} \cdot \oplus \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_i \cdot \mathscr{Q}_o}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i} + \frac{\mathscr{P}_o \cdot \mathscr{Q}_i}{\oplus \mathscr{P}_i + \oplus \mathscr{Q}_i}}$$

Using this rule, the associativity of parallel composition, up to *rate aware bisimulation*, is guaranteed. E.g., in the case of $\overline{a}^{\lambda}.P|(a^{\omega_1}.Q_1|a^{\omega_2}.Q_2)$ and $(\overline{a}^{\lambda}.P|a^{\omega_1}.Q_1)|a^{\omega_2}.Q_2$, after $\overleftrightarrow{a}$, the following rate functions are reachable:

$$\left[P|(Q_1|a^{\omega_2}.Q_2) \mapsto \frac{\lambda \cdot \omega_1}{\omega_1+\omega_2}\ ,\ P|(a^{\omega_1}.Q_1|Q_2) \mapsto \frac{\lambda \cdot \omega_2}{\omega_1+\omega_2}\right]$$

$$\left[(P|Q_1)|a^{\omega_2}.Q_2 \mapsto \frac{\lambda \cdot \omega_1}{\omega_1+\omega_2}\ ,\ (P|a^{\omega_1}.Q_1)|Q_2 \mapsto \frac{\lambda \cdot \omega_2}{\omega_1+\omega_2}\right]$$

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 [18] 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 synchronisation transitions ($\overleftrightarrow{a}$) that in [18] 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 $CTMC[\{P\}, \overleftrightarrow{\mathcal{C}} \cup \{\tau\}]$.

5 Conclusions

We have introduced a variant of Rate Transition Systems and used them to define the semantics of stochastic extensions of several process algebras among which CSP, CCS and π-calculus [6]. An original feature of this variant is that the transition relation associates to each process, for each action, the set of possible futures paired with a measure indicating their rates. This feature leads to a compact, uniform and elegant definition of the operational semantics. In one case this has also lead to the proposal of an alternative semantics for stochastic CCS that enjoys associativity of the parallel composition operator. 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 as well. 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. Finally, we have applied our framework also to the definition of stochastic process calculi for service oriented computing [21,3]. Interesting future work includes the further study of the format of the RTS rules aiming at reaching similar general results on bisimulation congruence as in [18].

References

1. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.-P.: Model-Checking Algorithms for Continuous-Time Markov Chains. IEEE Transactions on Software Engineering 29(6), 524–541 (2003)
2. Bernardo, M., Gorrieri, R.: 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. Bravetti, M., Latella, D., Loreti, M., Massink, M., Zavattaro, G.: Combining timed coordination primitives and probabilistic tuple spaces. In: TGC 2008. LNCS, vol. 5474, pp. 52–68. Springer, Heidelberg (2009)

4. Brinksma, E., Hermanns, H.: Process Algebra and Markov Chains. In: Brinksma, E., Hermanns, H., Katoen, J.-P. (eds.) EEF School 2000 and FMPA 2000. LNCS, vol. 2090, pp. 183–231. Springer, Heidelberg (2001)
5. De Nicola, R., Katoen, J.-P., Latella, D., Loreti, M., Massink, M.: Model checking mobile stochastic logic. Theoretical Computer Science 382(1), 42–70 (2007)
6. De Nicola, R., Latella, D., Loreti, M., Massink, M.: Rate-based Transition Systems for Stochastic Process Calculi, `http://www.dsi.unifi.it/~loreti/`
7. Deng, Y., van Glabbeek, R., Hennessy, M., Morgan, C., Zhang, C.: Characterising testing preorders for finite probabilistic processes. In: Proceedings of LICS 2007, pp. 313–325. IEEE Computer Society, Los Alamitos (2007)
8. Glotz, N., Herzog, U., Rettelbach, M.: Multiprocessor and distributed systems design: The integration of functional specification and performance analysis using stochastic process algebras. In: Donatiello, L., Nelson, R. (eds.) PECCS 1993. LNCS, vol. 729. Springer, Heidelberg (1993)
9. Haverkort, B.R.: Markovian Models for Performance and Dependability Evaluation. In: Brinksma, E., Hermanns, H., Katoen, J.-P. (eds.) EEF School 2000 and FMPA 2000. LNCS, vol. 2090, pp. 38–83. Springer, Heidelberg (2001)
10. Hermanns, H.: Interactive Markov Chains. In: Hermanns, H. (ed.) Interactive Markov Chains. LNCS, vol. 2428, pp. 57–88. Springer, Heidelberg (2002)
11. Hermanns, H., Herzog, U., Katoen, J.-P.: Process algebra for performance evaluation. Theoret. Comput. Sci. 274(1-2), 43–87 (2002)
12. Hermanns, H., Johr, S.: Uniformity by Construction in the Analysis of Nondeterministic Stochastic Systems. In: DSN 2007, pp. 718–728. IEEE Computer Society Press, Los Alamitos (2007)
13. Hermanns, H., Siegle, M.: Bisimulation algorithms for stochastic process algebras and their BDD-based implementation. In: Katoen, J.-P. (ed.) AMAST-ARTS 1999, ARTS 1999, and AMAST-WS 1999. LNCS, vol. 1601, pp. 244–264. Springer, Heidelberg (1999)
14. Hillston, J.: A compositional approach to performance modelling. Distinguished Dissertation in Computer Science, Cambridge University Press, Cambridge (1996)
15. Hillston, J.: Process algebras for quantitative analysis. In: IEEE Symposium on Logic in Computer Science, pp. 239–248. IEEE Computer Society Press, Los Alamitos (2005)
16. Hoare, C.: Communicating Sequential Processes. Prentice-Hall, Englewood Cliffs (1985)
17. Kemeny, J., Snell, J.: Finite Markov Chains. Springer, Heidelberg (1976)
18. Klin, B., Sassone, V.: Structural operational semantics for stochastic process calculi. In: Lipp, P., Sadeghi, A.-R., Koch, K.-M. (eds.) Trust 2008. LNCS, vol. 4968. Springer, Heidelberg (2008)
19. Milner, R.: Communication and Concurrency. Prentice-Hall, Englewood Cliffs (1989)
20. Milner, R., Parrow, J., Walker, J.: A Calculus of Mobile Processes, I and II. Information and Computation 100(1), 1–77 (1992)
21. Nicola, R.D., Latella, D., Loreti, M., Massink, M.: MarCaSPiS: a markovian extension of a calculus for services. In: Proc. of SOS 2008. ENTCS, Elsevier, Amsterdam (2008)
22. Priami, C.: Stochastic π-Calculus. The Computer Journal 38(7), 578–589 (1995)
23. Puterman, M.: Markov Decision Processes (1994)