

Structural Operational Semantics for Stochastic Process Calculi

Bartek Klin¹ and Vladimiro Sassone²

¹ Warsaw University, University of Edinburgh

² ECS, University of Southampton

Abstract. A syntactic framework called SGSOS, for defining well-behaved Markovian stochastic transition systems, is introduced by analogy to the GSOS congruence format for nondeterministic processes. Stochastic bisimilarity is guaranteed a congruence for systems defined by SGSOS rules. Associativity of parallel composition in stochastic process algebras is also studied within the framework.

1 Introduction

Process algebras such as CCS [18] or CSP [5] are widely accepted as useful tools for compositional modeling of nondeterministic, communicating processes. Their semantics is usually described within the framework of Structural Operational Semantics (SOS) [19], where labelled nondeterministic transition systems (LTSs) are defined by induction on the syntactic structure of processes. Formalisms for SOS descriptions of nondeterministic systems have been widely studied and precisely defined (see [1] for a survey). In particular, several syntactic formats have been developed that guarantee certain desirable properties of the induced systems, most importantly that bisimulation is a congruence on them.

Stochastic process algebras have been deployed for applications in performance evaluation, and more recently in systems biology, where the underpinning of labelled continuous time Markov chains (CTMCs), and more generally stochastic processes, is required rather than simple LTSs. Examples of such algebras include TIPP [11], PEPA [15], EMPA [3], and stochastic π -calculus [20]. Semantics of these calculi have been given by variants of the SOS approach. However, in contrast with the case of nondeterministic processes, SOS formalisms used here are not based on any general framework for operational descriptions of stochastic processes, and indeed differ substantially from one another. This is unfortunate, as such a framework would make languages easier to understand, compare, and extend. Specifically, a format for SOS descriptions which guarantees the compositionality of stochastic bisimilarity, would make extending process algebras with new operators a much simpler task, liberating the designer from the challenging and time-consuming task of proving congruence results.

In this paper we define such a *congruence format*, which we call *SGSOS*. First we review existing approaches to the operational semantics of process algebras, concentrating on the examples of PEPA [15] and the stochastic π -calculus [20]. As the operational techniques used there seem hard to extend to a general format for well-behaved stochastic specifications, we resolve to adapt a general theory of well-behaved SOS, based on category theory and developed by Turi and Plotkin [24]. The inspiration for our approach comes directly from the work of F. Bartels [2], who used Turi and Plotkin's results to design a congruence format for probabilistic transition systems.

Standard operations of stochastic process algebras, as well as plenty of non-standard but potentially useful ones, fall within our format. Exceptions are recursive definitions and name-passing features of stochastic π -calculus, which we leave for future work.

Within the SGSOS framework, we also investigate the issue of *associativity of parallel composition* in stochastic process algebras, a design issue that, to our knowledge, has been overlooked in the literature. We notice in fact that in the original definition of stochastic π -calculus, parallel composition fails to be associative up to stochastic bisimilarity, and study conditions under which two forms of parallel composition, CSP-style synchronization and CCS-style communication, are associative.

The structure of the paper is as follows. In §2 we recall previously studied approaches to operational semantics of nondeterministic and stochastic systems. In §3 the bialgebraic theory of well-behaved SOS is recalled. In §4 we adapt the theory to obtain the SGSOS congruence format, with simple examples of GSOS specifications following in §5. The associativity of parallel composition is studied in §6, and in §7 we mention some directions of future work. Due to lack of space, all proofs are omitted in this extended abstract.

2 Transition systems and process calculi

We begin our development by comparing two previously studied approaches to defining SOS for Markovian process algebras with the well-known world of SOS for nondeterministic systems such as CCS.

2.1 Nondeterministic systems and GSOS

A *labelled transition system* (LTS) is a triple $(X, A, \longrightarrow)$, with X a set of *states*, A a set of *labels* and $\longrightarrow \subseteq X \times A \times X$ a labelled *transition relation*, typically written $x \xrightarrow{a} y$ for $(x, a, y) \in \longrightarrow$. An LTS is *image-finite* if for every $x \in X$ and $a \in A$ there are only finitely many $y \in X$ such that $x \xrightarrow{a} y$. In the context of Structural Operational Semantics (SOS), LTS states are terms, and transition relations are defined inductively, by means of inference rules. For example, in a fragment of CCS [18], processes are terms over the grammar $P ::= \text{nil} \mid a.P \mid P + P \mid P \parallel P$, and the LTS is induced from the following rules:

$$\begin{array}{c}
 \frac{}{a.x \xrightarrow{a} x} \quad \frac{x_1 \xrightarrow{a} y}{x_1 + x_2 \xrightarrow{a} y} \quad \frac{x_2 \xrightarrow{a} y}{x_1 + x_2 \xrightarrow{a} y} \\
 \frac{x_1 \xrightarrow{a} y}{x_1 \parallel x_2 \xrightarrow{a} y \parallel x_2} \quad \frac{x_2 \xrightarrow{a} y}{x_1 \parallel x_2 \xrightarrow{a} x_1 \parallel y} \quad \frac{x_1 \xrightarrow{a} y_1 \quad x_2 \xrightarrow{\bar{a}} y_2}{x_1 \parallel x_2 \xrightarrow{\tau} y_1 \parallel y_2}
 \end{array} \tag{1}$$

Plenty of operators can be defined formally by rules like these. Indeed, the above specification is an instance of a general framework for SOS definitions of LTSs (see e.g., [1]), called *GSOS* and defined formally as follows.

An *algebraic signature* is a set $\Sigma \ni f, g, \dots$ of *operation symbols* with an *arity* function $ar : \Sigma \rightarrow \mathbb{N}$, usually left implicit. The set of all terms over Σ with variables from set X is denoted $T_\Sigma X$. In particular, $T_\Sigma 0$ is the set of closed Σ -terms.

Fix a countably infinite set $\mathcal{E} \ni x, y, z, \dots$ of variables. A *GSOS inference rule* [4] over a signature Σ and a set of labels A is an expression of the form

$$\frac{\left\{x_{i_j} \xrightarrow{a_j} y_j\right\}_{1 \leq j \leq k} \quad \left\{x_{i_l} \not\xrightarrow{b_l}\right\}_{1 \leq l \leq m}}{f(x_1, \dots, x_n) \xrightarrow{c} t} \quad (2)$$

where $f \in \Sigma$, $n = ar(f)$, $k, m \in \mathbb{N}$, $i_j, i_l \in \{1, \dots, n\}$, $a_j, b_l, c \in A$, $t \in T_\Sigma \mathcal{E}$, x_i and $y_j \in \mathcal{E}$ are all distinct and no other variables occur in the term t . Expressions above the horizontal line in a GSOS rule are called its *premises*, and the expression below it is the *conclusion*. A *GSOS specification* is a set of GSOS rules; it is *image-finite* if it contains only finitely many rules for each f and c .

Every GSOS specification Λ induces an LTS $(T_\Sigma 0, A, \longrightarrow)$, with the transition relation $\longrightarrow$ defined by induction of the syntactic structure of the source states. For a term $s = f(s_1, \dots, s_n) \in T_\Sigma 0$, one adds a transition $s \xrightarrow{c} t$ for each substitution $\sigma : \mathcal{E} \rightarrow T_\Sigma 0$ such that for some rule $r \in \Lambda$ as in (2), there is $\sigma x_i = s_i$, $\sigma t = t$, and σ satisfies all premises of r , meaning that for each premise $x \xrightarrow{a} y$ there is $\sigma x \xrightarrow{a} \sigma y$, and for each premise $x \not\xrightarrow{a}$ there is no $t \in T_\Sigma 0$ for which $\sigma x \xrightarrow{a} t$.

An important property of the LTS induced by Λ is that bisimilarity on it is guaranteed to be a congruence with respect to the syntactic structure of states. This means that GSOS is a *congruence format* for bisimilarity on LTSs. Moreover, it is easy to prove by induction that the LTS induced by an image-finite GSOS specification is image-finite.

2.2 Stochastic systems

Just as nondeterministic process algebras are defined using labelled transition systems, the semantics of stochastic processes is often provided by labelled continuous time Markov chains (CTMCs). These are conveniently presented in terms of what we shall call *rated transition systems* (RTSs), i.e., triples (X, A, ρ) , where X is a set of states, A a set of labels and $\rho : X \times A \times X \rightarrow \mathbb{R}_0^+$ is a rate function, equivalently presented as an A -indexed family of $\mathbb{R}_0^+$ -valued matrices. The number $\rho(x, a, y)$ is the parameter of an exponential probability distribution governing the duration of the transition of x to y with label a (for more information and intuition on CTMCs and their presentation by transition rates see e.g. [12, 15, 20]). For the sake of readability we will write $\rho(x \xrightarrow{a} y)$ instead of $\rho(x, a, y)$, and $x \xrightarrow{a, r} y$ will indicate that $\rho(x \xrightarrow{a} y) = r$. The latter notation suggests that RTSs can be seen as a special kind of $A \times \mathbb{R}_0^+$ -labelled nondeterministic transition systems; more specifically, exactly those that are “rate-deterministic,” i.e., such that for each $x, y \in X$ and $a \in A$ there exists exactly one $r \in \mathbb{R}_0^+$ for which $x \xrightarrow{a, r} y$.

In the following we will consider *image-finite* processes, i.e. such that for each $x \in X$ and $a \in A$ there are only finitely many $y \in X$ such that $\rho(x, a, y) > 0$. For such processes, the sum

$$\rho_a(x) = \sum_{y \in X} \rho(x \xrightarrow{a} y) \quad (3)$$

exists for each $x \in X$ and $a \in A$; it will be called the *apparent rate* of label a in state x . Further, $\rho(x \xrightarrow{a} y) / \rho_a(x)$ is called the *conditional probability* of the transition $x \xrightarrow{a} y$. It is the probability that x makes the transition provided that it makes some a -transition.

Various equivalence relations on states in RTSs have been considered. Of those, the most significant is *stochastic bisimilarity* (called strong equivalence in [14], and inspired by the notion of probabilistic bisimilarity from [17]), defined as follows. Given an RTS with state space X , a *stochastic bisimulation* is an equivalence relation R on X such that whenever $x R y$ then for each $a \in A$, and for each equivalence class C with respect to R ,

$$\sum_{z \in C} \rho(x \xrightarrow{a} z) = \sum_{z \in C} \rho(y \xrightarrow{a} z).$$

Two states are *bisimilar* if they are related by some bisimulation. It is easy to check that bisimilarity is itself an equivalence relation and indeed the largest bisimulation.

Due to the additional rate component present in transitions, the traditional approach to SOS recalled in §2.1 is inadequate for modeling stochastic process calculi. Instead, other variants of SOS have been used for this purpose. For a comparison with the following development, we recall two of these variants: the multi-transition system approach used for the stochastic calculus PEPA [14, 15], and the proved SOS approach of stochastic π -calculus [20–22].

In (a fragment of) PEPA, processes are terms over the grammar:

$$P ::= \text{nil} \mid (a, r).P \mid P + P \mid P \bowtie_L P$$

where a ranges over a fixed set A of labels, L over subsets of A , and r over $\mathbb{R}^+$. Their semantics is defined by inference rules:

$$\begin{array}{c} \frac{}{(a, r).x \xrightarrow{a, r} x} \quad \frac{x_1 \xrightarrow{a, r} y}{x_1 + x_2 \xrightarrow{a, r} y} \quad \frac{x_2 \xrightarrow{a, r} y}{x_1 + x_2 \xrightarrow{a, r} y} \\[10pt] \frac{x_1 \xrightarrow{a, r} y}{x_1 \bowtie_L x_2 \xrightarrow{a, r} y \bowtie_L x_2} \quad \frac{x_2 \xrightarrow{a, r} y}{x_1 \bowtie_L x_2 \xrightarrow{a, r} x_1 \bowtie_L y} \quad (a \notin L) \\[10pt] \frac{x_1 \xrightarrow{a, r_1} y_1 \quad x_2 \xrightarrow{a, r_2} y_2}{x_1 \bowtie_L x_2 \xrightarrow{a, R} y_1 \bowtie_L y_2} \quad (a \in L) \end{array} \quad (4)$$

where $a \in A$ and $r, r_1, r_2, R \in \mathbb{R}^+$ with R depending on r_1, r_2 according to an application-specific formula (see below). Note that instead of a single parallel composition operator, PEPA provides a *cooperation operator* $\bowtie_L$ for each set L of labels. These operators are based on CSP-style synchronisation [5] rather than CCS-style communication [18].

It turns out that the standard interpretation of the above rules as described in §2.1 would (among other things) contradict the intended meaning of the operator $+$ as a *stochastic choice*, where a process $P + P$ can perform the same transitions as P , with twice the rates. In particular, the processes P and $P + P$ should not be stochastic bisimilar. This is why the semantics of PEPA is given as a *multi-transition* system labeled with pairs $(a, r) \in A \times \mathbb{R}^+$, which is a transition system whose transition relation is a *multiset* of triples $(x, (a, r), y)$. To define such a semantics for PEPA, the rules in (4) are interpreted similarly as the GSOS rules in §2.1, where the multiplicity of a transition is determined by counting all its different derivations. To obtain an RTS from the induced

multi-transition system, one then discards multiplicities by summing up all their rates in single rated transitions. Thus, for example, the process $(a, 3).\text{nil} + (a, 3).\text{nil}$ in the induced multi-transition systems has two identical transitions to nil with label $(a, 3)$, whilst in the final RTS it can make a single transition to nil with label a and rate 6. For more details of this construction, see [14].

The formula for calculating R based on r_1 and r_2 in the last rule of (4) depends on the intended meaning of synchronisation. In applications to performance evaluation [14], the formula

$$R = \min(\rho_a(\mathbf{x}_1), \rho_a(\mathbf{x}_2)) \cdot \frac{r_1}{\rho_a(\mathbf{x}_1)} \cdot \frac{r_2}{\rho_a(\mathbf{x}_2)} \quad (5)$$

is a natural choice. We shall call it the *minimal rate law*, since in the resulting RTS, the apparent rate of a in $P \bowtie_L Q$ (with $a \in L$) is the least of the apparent rates of P and Q . For applications to systems biology, where rates model concentrations of molecules, a more convenient choice is

$$R = r_1 \cdot r_2, \quad (6)$$

which following [6] we call the *mass action law*. The apparent rate of a in $P \bowtie_L Q$ (with $a \in L$) here is the product of the corresponding apparent rates of P and Q . For an intuitive motivation for these and other similar formulae, see [13].

A different approach was used to define semantics of stochastic π -calculus [20]. Since stochastic features of the calculus are independent from its name-passing aspects, for simplicity we discuss it here on a fragment of the calculus that corresponds to a stochastic version of CCS (see §2.1). Thus we consider, as processes, terms over the grammar:

$$P ::= \text{nil} \mid (a, r).P \mid P + P \mid P \parallel P$$

where a ranges over a fixed set A of labels, and r over $\mathbb{R}^+$. For the semantics, the authors of [20] decided to avoid multi-transition systems and rely on the standard process of LTS induction from inference rules. For this, to model stochastic choice and communication accurately, they enriched transition labels substantially, equipping them with encodings of derivations that lead to them. In this *proved operational semantics*, our “stochastic CCS” fragment of stochastic π -calculus would be defined by:

$$\begin{array}{c} \frac{}{(a, r).x \xrightarrow{(a, r)} x} \quad \frac{x_1 \xrightarrow{\theta} y}{x_1 + x_2 \xrightarrow{+_1 \theta} y} \quad \frac{x_2 \xrightarrow{\theta} y}{x_1 + x_2 \xrightarrow{+_2 \theta} y} \\[10pt] \frac{x_1 \xrightarrow{\theta} y}{x_1 \parallel x_2 \xrightarrow{\parallel_1 \theta} y \parallel x_2} \quad \frac{x_2 \xrightarrow{\theta} y}{x_1 \parallel x_2 \xrightarrow{\parallel_2 \theta} x_1 \parallel y} \quad \frac{x_1 \xrightarrow{\theta_1(a, r_1)} y_1 \quad x_2 \xrightarrow{\theta_2(\bar{a}, r_2)} y_2}{x_1 \parallel x_2 \xrightarrow{\langle \parallel_1 \theta_1(a, r_1), \parallel_1 \theta_2(\bar{a}, r_2) \rangle, R} y_1 \parallel y_2} \end{array} \quad (7)$$

where θ ranges over *derivation proofs*, e.g. represented by terms of the grammar:

$$\theta = (a, r) \mid +_1 \theta \mid +_2 \theta \mid \parallel_1 \theta \mid \parallel_2 \theta \mid \langle \parallel_1 \theta, \parallel_2 \theta \rangle,$$

and where R depends on r_1 and r_2 according to the minimal rate law [20] or the mass action law [22], as in PEPA.

These rules are then used to induce an LTS, which results in relatively complex labels. To obtain an RTS, one then extracts more familiar labels $a \in A$ from proofs

in the obvious way, by adding up rates of identical transitions. Thus, for example, the process $P = (a, 3).\text{nil} + (a, 3).\text{nil}$ in the induced LTS can make two distinct transitions $P \xrightarrow{+_1(a,3)} \text{nil}$ and $P \xrightarrow{+_2(a,3)} \text{nil}$, and in the final RTS it can make a transition to nil with label a and rate 6.

Although both the multi- and the proved-transition approaches work fine for the specific examples described above, it appears difficult to extend any of them to a general framework for defining operational semantics for stochastic transition systems. Consider for example the proved SOS approach of stochastic π -calculus. As in the case of GSOS for nondeterministic systems, a well-behaved semantic framework should guarantee that stochastic bisimilarity is a congruence for the induced RTS. This is the case for our CCS example above, but it is easy to write examples where it fails; for example, extend the CCS language with a unary operator f with semantics defined by a rule:

$$\frac{x \xrightarrow{+_1\theta} y}{f(x) \xrightarrow{f+1\theta} y}$$

and see that, although $(a, 2).\text{nil} + \text{nil}$ and $\text{nil} + (a, 2).\text{nil}$ are stochastic bisimilar, they are not so when put in context $f(-)$, since only the former process can make a step in this context. Clearly, this is because the structure of a proof is inspected in the premise of the rule. However, it would be wrong to forbid such inspection altogether, as it is needed, e.g. in the communication rule for stochastic π -calculus.

The source of the problem is the richness of labels in the proved approach to SOS. In [8], it is claimed that proofs as transition labels carry almost all information about processes that is ever needed. Indeed, it appears they may sometimes carry excessive information; in a well-behaved SOS framework they should only carry as much data as required for the derivation of the intended semantics (here, an RTS), not a bit more.

The same criticism, though perhaps to a lesser extent, can be moved to the multi-transition systems approach used in the semantics of PEPA, where transition multiplicities are the superfluous data. In the process of multi-transition system induction, two identical transitions of rate 3 are distinguished from a single transition of rate 6. As a result, one can write specifications such as

$$\frac{x \xrightarrow{a,r} y}{f(x) \xrightarrow{a, \max(r,5)} y}$$

and see that, although processes $(a, 3).\text{nil} + (a, 3).\text{nil}$ and $(a, 6).\text{nil}$ are stochastic bisimilar, they are not so in the context $f(-)$. On the other hand, forbidding arbitrary dependency of transition rates on subprocesses rates is hard to contemplate, since that forms the very core of PEPA.

It may be possible to determine the exact range of constructs and formulas that must be forbidden in the proved- or in the multi-transition approach in order to guarantee that stochastic bisimilarity is compositional. Indeed, this approach has been used with success in the related framework of probabilistic processes [16], where a well-behaved version of the proved semantics is developed. In this paper, however, we take a more principled approach and derive a formalism for stochastic operational semantics from an abstract theory of congruence formats developed in [24] and applied to the case of probabilistic transition systems in [2].

3 An abstract approach to SOS

Our approach to a stochastic counterpart of the GSOS framework of §2.1 is based on a categorical generalisation of GSOS, developed by Plotkin and Turi in [24]. In this section we briefly recall that work; in the rest of the paper we develop a syntactic format for stochastic SOS as an instance of the general framework.

3.1 Transition systems as coalgebras

The abstract study of well-behaved structural operational semantics is based on modeling the behaviour of processes via coalgebras, and their syntax via algebras. The original motivating example is that of LTSs: for a fixed set A of labels, image-finite LTSs can be seen as functions $h : X \rightarrow (\mathcal{P}_\omega X)^A$ (here, $\mathcal{P}_\omega$ is the finite powerset construction), along the correspondence $y \in h(x)(a)$ if and only if $x \xrightarrow{a} y$. More generally, for any covariant functor B on the category **Set** of sets and functions, a B -coalgebra is a set X (the *carrier*) and a function $h : X \rightarrow BX$ (the *structure*). Thus image-finite LTSs are coalgebras for the functor $(\mathcal{P}_\omega -)^A$.

A B -coalgebra *morphism* from a $h : X \rightarrow BX$ to $g : Y \rightarrow BY$ is a function $f : X \rightarrow Y$ such that the equation $g \circ f = Bf \circ h$ holds. This notion provides a general coalgebraic treatment of process equivalences: a *bisimulation* on a coalgebra $h : X \rightarrow BX$ is a binary relation $Q \subseteq X \times X$ such that for some coalgebra structure $q : Q \rightarrow BQ$ the projections $\pi_1, \pi_2 : Q \rightarrow X$ extend to a span of coalgebra morphisms from q to h . For example, for $B = (\mathcal{P}_\omega -)^A$, this *span bisimulation* specializes to the well-known notion of LTS bisimulation [18]. For more information about the coalgebraic approach to process theory, see [23].

We now show how to view RTSs as coalgebras for a suitable functor on **Set**. Call a function $f : X \rightarrow \mathbb{R}_0^+$ *finitely supported* if the set $\{x \in X \mid f(x) > 0\}$ is finite. For any set X , let $\mathcal{R}_\omega X$ be the set of all finitely supported functions from X to $\mathbb{R}_0^+$. This extends to a functor $\mathcal{R}_\omega$ on **Set**, with the action $\mathcal{R}_\omega f$ on function $f : X \rightarrow Y$ defined by

$$\mathcal{R}_\omega f(g)(y) = \sum_{f(x)=y} g(x),$$

for $g \in \mathcal{R}_\omega X$ and $y \in Y$. Since g is finitely supported the sum exists and $\mathcal{R}_\omega f(g)$ is finitely supported too. Functoriality of $\mathcal{R}_\omega$ is then easy to check.

Fix an arbitrary set A of labels. Coalgebras for the functor

$$BX = (\mathcal{R}_\omega X)^A$$

are exactly image-finite rated transition systems as defined in §2.2. Indeed, a coalgebra $h : X \rightarrow BX$ is an image-finite RTS with states X along the correspondence:

$$x \xrightarrow{a,r} y \quad \text{if and only if} \quad r = h(x)(a)(y).$$

This coalgebraic treatment of RTSs is justified by the following statement.

Proposition 1. Span bisimulations on $(\mathcal{R}_\omega -)^A$ -coalgebras, when restricted to equivalence relations, are exactly stochastic bisimulations as defined in §2.2.

In the following, a technical property of the functor $(\mathcal{R}_\omega -)^A$ will be useful:

Proposition 2. $(\mathcal{R}_\omega -)^A$ preserves weak pullbacks.

To prove the above two results, proceed exactly as in [7] for the case of probabilistic bisimulation and the corresponding behaviour functor.

3.2 Process syntax via algebras

In the context of SOS, processes typically are closed terms over some algebraic signature, i.e., a set $\Sigma \ni f, g, \dots$ of operation symbols with an arity function $ar : \Sigma \rightarrow \mathbb{N}$. Such a signature corresponds to a functor $\Sigma X = \coprod_{f \in \Sigma} X^{ar(f)}$ on **Set**, in the sense that a model for the signature is exactly an *algebra* for the functor, i.e., a set X (the *carrier*) and a function $g : \Sigma X \rightarrow X$ (the *structure*).

The set of terms over a signature Σ and a set X of variables is denoted by $T_\Sigma X$; in particular, $T_\Sigma 0$ is the set of closed terms over Σ and it admits an obvious algebra structure $a : \Sigma T_\Sigma 0 \rightarrow T_\Sigma 0$ for the *functor* Σ corresponding to the signature. This is the *initial* Σ -algebra. The construction T_Σ is also a functor, called the *free monad* over Σ .

3.3 SOS rules, distributive laws, bialgebras

In [24], Turi and Plotkin proposed an elegant treatment of well-behaved SOS at the level of algebras and coalgebras. Their main motivating application was GSOS (see §2.1). Turi and Plotkin observed (full proof provided later by Bartels [2]), that image finite GSOS specifications are in an essentially one-to-one correspondence with *distributive laws*, i.e., natural transformations of the type

$$\lambda : \Sigma(\text{Id} \times B) \Longrightarrow BT_\Sigma \quad (8)$$

where $B = (\mathcal{P}_\omega -)^A$ is the behaviour functor used for modeling LTSs, Σ is the functor corresponding to the given signature, and T_Σ is the free monad over Σ . Informally, (8) says that ‘structural’ combinations (Σ) of behaviours (B) are mapped to the behaviour of terms (BT_Σ), which is the essence of a SOS rule, with Id accounting for subterms that stay idle in a transition. Moreover, any λ as above gives rise to a B -coalgebra structure h_λ on $T_\Sigma 0$, defined by a “structural recursion theorem” (see [24] for details) as the only function $h_\lambda : T_\Sigma 0 \rightarrow BT_\Sigma 0$ such that:

$$h_\lambda \circ a = Ba^\sharp \circ \lambda_X \circ \Sigma\langle \text{id}, h_\lambda \rangle. \quad (9)$$

The fact that bisimilarity on LTSs induced from GSOS specifications is guaranteed to be a congruence, can be proved at the level of coalgebras and distributive laws:

Theorem 1 ([24], Cor. 7.5). If a functor B on **Set** preserves weak pullbacks, then for any λ as in (8), span bisimilarity on $h_\lambda : T_\Sigma 0 \rightarrow BT_\Sigma 0$ is a congruence on $T_\Sigma 0$.

This result, together with Propositions 1 and 2, is the basis of our search for a congruence format for stochastic systems.

4 Stochastic GSOS

We now proceed to the main technical contribution of this paper: a complete characterisation of distributive laws (8) for stochastic systems in terms of inference rules. To find the characterisation, we closely follow the technique used by Bartels [2] for the case of probabilistic transition systems.

Definition 1. An *SGSOS rule* for a signature Σ and a set A of labels is an expression of the form:

$$\frac{\left\{ \mathbf{x}_i \xrightarrow{a @ r_{ai}} \right\}_{a \in D_i, 1 \leq i \leq n} \quad \left\{ \mathbf{x}_{i_j} \xrightarrow{b_j} \mathbf{y}_j \right\}_{1 \leq j \leq k}}{\mathbf{f}(\mathbf{x}_1, \dots, \mathbf{x}_n) \xrightarrow{c @ W} \mathbf{t}} \quad (10)$$

where

- $\mathbf{f} \in \Sigma$ and $ar(\mathbf{f}) = n$, with $n, k \in \mathbb{N}$, and $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$;
- $\mathbf{x}_i$ and $\mathbf{y}_j$ are all distinct variables and no other variables appear in $\mathbf{t} \in T_\Sigma \mathcal{E}$; moreover, all variables $\mathbf{y}_j$ appear in $\mathbf{t}$;
- $D_i \subseteq A$, $c \in A$ and $b_j \in D_{i_j}$,
- $W \in \mathbb{R}^+$, $r_{ai} \in \mathbb{R}_0^+$, and moreover $r_{b_j i_j} > 0$, for $j = 1, \dots, k$.

A rule is *triggered* by a tuple of real values $(v_{ai})_{a \in A, 1 \leq i \leq n}$ if $v_{ai} = r_{ai}$ for all $1 \leq i \leq n$ and all $a \in D_i$. A collection of rules is called an *SGSOS specification* if for every $\mathbf{f} \in \Sigma$, $c \in A$, every tuple (v_{ai}) triggers only finitely many rules with $\mathbf{f}$ and c in the conclusion.

In order to complete the definition of SGSOS we need describe how to derive an RTS from an SGSOS specification. Intuitively, a rule as in (10) contributes to the rate of a c -labeled transition from $\mathbf{f}(s_1, \dots, s_n)$ if the *apparent* a -rates (see Eqn. (3)) of the s_i match the corresponding r_{ai} ; its contribution depends on W and on *conditional probabilities* of a selection $s_{i_j} \xrightarrow{b_j} u_j$ of transitions from the s_i . Formally:

Definition 2. Every SGSOS specification Λ *induces* a rated transition system $(T_\Sigma 0, A, \rho)$, with the rate function ρ defined by induction on the first argument as follows. For a term $s = \mathbf{f}(s_1, \dots, s_n) \in T_\Sigma 0$, assume that $\rho(s_i \xrightarrow{a} u)$ has been defined for all $i = 1, \dots, n$, all $a \in A$ and all $u \in T_\Sigma 0$; then, for any $c \in A$ and $t \in T_\Sigma 0$, define $\rho(s \xrightarrow{c} t)$ as below.

Let $\Lambda_c \subseteq \Lambda$ be the set of all those rules with $\mathbf{f}$ and c in the conclusion that are triggered by the tuple of apparent rates $v_{ai} = \rho_a(s_i)$ – cf. (3). Note that Λ_c is *finite*. To calculate the value of $\rho(s \xrightarrow{c} t)$, look at each rule $L \in \Lambda_c$ in turn and check whether there exists a substitution $\sigma : \mathcal{E} \rightarrow T_\Sigma 0$ such that $\sigma \mathbf{t} = t$ and $\sigma \mathbf{x}_i = s_i$ for $i = 1, \dots, n$. Note that although many such σ may exist, their values on each $\mathbf{y}_j$ coincide, since all $\mathbf{y}_j$ appear in $\mathbf{t}$. If σ exists, calculate the *contribution* $\gamma_L \in \mathbb{R}_0^+$ of L to $\rho(s \xrightarrow{c} t)$ according to the formula:

$$\gamma_L = W \cdot \prod_{j=1}^k \frac{\rho(s_{i_j} \xrightarrow{b_j} \sigma \mathbf{y}_j)}{\rho_{b_j}(s_{i_j})}$$

where W , k , i_j , b_j and $\mathbf{y}_j$ are determined by the shape of rule L – cf. rule format (10). Note that the quotient is well defined since $\rho_{b_j}(s_{i_j}) = r_{b_j i_j} > 0$. If no suitable σ exist, take $\gamma_L = 0$. Finally, define $\rho(s \xrightarrow{c} t) = \sum_{L \in \Lambda_c} \gamma_L$; the sum exists since Λ_c is finite.

Notation 1. If in a rule as in (10), for some $\mathbf{x}_i$ and $a \in D_i$ there is exactly one j for which $i_j = i$ and $b_j = a$, instead of the two premises $\mathbf{x}_i \xrightarrow{a@r_{ai}}$ and $\mathbf{x}_i \xrightarrow{a} \mathbf{y}_j$ we shall write simply $\mathbf{x}_i \xrightarrow{a@r_{ai}} \mathbf{y}_j$. Note that, unlike in the frameworks recalled in §2.2, such a premise does *not* require that a transition $\mathbf{x}_i \xrightarrow{a} \mathbf{y}_j$ has rate r_{ai} ; instead, r_{ai} refers to the *apparent* rate $\rho_a(\mathbf{x}_i)$. To avoid this confusion, @ is used in (10) instead of a comma.

It turns out that SGSOS specifications correspond to distributive laws (8) for the behaviour functor used for modeling stochastic systems:

Theorem 2. For all signatures Σ and label sets A , every SGSOS specification Λ for Σ and A determines a distributive law $\lambda : \Sigma(\text{Id} \times (\mathcal{R}_\omega)^A) \Longrightarrow (\mathcal{R}_\omega T_\Sigma)^A$ such that $h_\lambda : T_\Sigma 0 \rightarrow (\mathcal{R}_\omega T_\Sigma 0)^A$ defined as in (9) coincides with the RTS induced by Λ . Moreover, every such distributive law Λ is defined by an SGSOS specification.

Corollary 1. Stochastic bisimilarity on RTSs induced by SGSOS specifications is always a congruence. (*Proof:* Combine Theorems 2 and 1 with Propositions 1 and 2.)

Although technically more involved, the correspondence between SGSOS and RTSs is a perfect match for that for GSOS and LTSs, and lifts the benefits of congruence formats to the equally more involved semantics of stochastic models. In §5 below we shall illustrate that the format affords expressiveness, conciseness and elegance.

5 Examples of SGSOS

To illustrate the form of SGSOS specifications, we now present a few simple examples, including operators present in stochastic π -calculus or in PEPA, as well as some other operators of potential interest.

Example 1 (atomic actions). A basic ingredient of most process calculi is prefixing composition with atomic actions. To model stochastic systems, these actions are equipped with basic rates. For the simplest nontrivial example of SGSOS, fix a set A of labels and consider a language with syntax defined by the grammar:

$$P ::= \text{nil} \mid (a, r).P$$

where a ranges over A and r over $\mathbb{R}^+$. The semantics of nil is defined by the empty set of rules, and the semantics of a unary operator $(a, r)._$ is defined by a single rule:

$$\frac{}{(a, r).x \xrightarrow{a@r} x}$$

Thus, according to Definition 2, the process $P = (a, 2).(b, 3).\text{nil}$ can make a unique transition $P \xrightarrow{a, 2} (b, 3).\text{nil}$ in the transition system induced by the rules.

Example 2 (stochastic choice). Consider an extension of the language from Example 1 with a binary operator $P + P$ with semantics defined by rules:

$$\frac{x_1 \xrightarrow{a@r} y}{x_1 + x_2 \xrightarrow{a@r} y} \quad \frac{x_2 \xrightarrow{a@r} y}{x_1 + x_2 \xrightarrow{a@r} y}$$

(note the use of Notation 1), for each $a \in A$ and $r \in \mathbb{R}^+$. Note that this is a well-defined SGSOS specification. Although it contains uncountably many rules, for every $a \in A$ exactly two rules are triggered by every tuple of apparent rates. The rules define $+$ to be the stochastic choice operator, as present e.g. in PEPA and stochastic π -calculus. In particular, according to Definition 2, in the stochastic transition system induced by the rules, the process $P = ((a, 2).\text{nil} + (a, 2).(b, 1).\text{nil}) + (c, 3).\text{nil}$ can make three transitions $P \xrightarrow{a,2} \text{nil}$, $P \xrightarrow{a,2} (b, 1).\text{nil}$ and $P \xrightarrow{c,3} \text{nil}$. Note, however, that the process $Q = (a, 2).\text{nil} + (a, 3).\text{nil}$ can only make one transition $Q \xrightarrow{a,5} \text{nil}$. In particular, processes $(a, 2).\text{nil} + (a, 3).\text{nil}$ and $(a, 5).\text{nil}$ are not only stochastic bisimilar, but can actually make exactly the same outgoing transitions.

We remark that when compared to the existing literature, in all our examples the expected semantics of the operators arises naturally from intuitive and elementary specifications, witness of the flexibility of the SGSOS format.

Example 3 (PEPA-style synchronisation). Extend the language from Example 2 with a binary synchronisation operator $\boxtimes_L$ for each $L \subseteq A$, with semantics defined by a family of rules:

$$\frac{x_1 \xrightarrow{a@r} y}{x_1 \boxtimes_L x_2 \xrightarrow{a@r} y \boxtimes_L x_2} \quad \frac{x_2 \xrightarrow{a@r} y}{x_1 \boxtimes_L x_2 \xrightarrow{a@r} x_1 \boxtimes_L y} \quad (11)$$

$$\frac{x_1 \xrightarrow{b@r_1} y_1 \quad x_2 \xrightarrow{b@r_2} y_2}{x_1 \boxtimes_L x_2 \xrightarrow{b@W} y_1 \boxtimes_L y_2} \quad (12)$$

for each $a \in A \setminus L$, $b \in L$, and $r, r_1, r_2, W \in \mathbb{R}^+$ such that $W = \min(r_1, r_2)$. It is not difficult to see that this, according to Definition 2, is the synchronisation operator of PEPA where the minimal rate law (5) is used. As an example, consider processes:

$$P = (a, 1).P_1 + (a, 3).P_2 \quad Q = (a, 2).Q_1 \quad (13)$$

where $P_1 \neq P_2$. Then the process $P \boxtimes_{[b]} Q$, where $b \neq a$, can make the transitions:

$$P \boxtimes_{[b]} Q \xrightarrow{a,1} P_1 \boxtimes_{[b]} Q \quad P \boxtimes_{[b]} Q \xrightarrow{a,3} P_2 \boxtimes_{[b]} Q \quad P \boxtimes_{[b]} Q \xrightarrow{a,2} P \boxtimes_{[b]} Q_1.$$

On the other hand, the outgoing transitions from $P \boxtimes_{[a]} Q$ are:

$$P \boxtimes_{[a]} Q \xrightarrow{a,\frac{1}{2}} P_1 \boxtimes_{[a]} Q_1 \quad P \boxtimes_{[a]} Q \xrightarrow{a,\frac{3}{2}} P_2 \boxtimes_{[a]} Q_1.$$

Example 4 (CCS-style communication). Similarly, one can extend the language from Example 2 with a CCS-style communication operator. Assume $A = A_0 \cup \{\bar{a} \mid a \in A_0\} \cup \{\tau\}$ (denote $\bar{\bar{a}} = a$) and extend the language with a single binary operator $\parallel$, with semantics defined by rules:

$$\frac{x_1 \xrightarrow{a@r} y}{x_1 \parallel x_2 \xrightarrow{a@r} y \parallel x_2} \quad \frac{x_2 \xrightarrow{a@r} y}{x_1 \parallel x_2 \xrightarrow{a@r} x_1 \parallel y} \quad (14)$$

$$\frac{x_1 \xrightarrow{a@r_1} y_1 \quad x_2 \xrightarrow{\bar{a}@r_2} y_2}{x_1 \parallel x_2 \xrightarrow{\tau@W} y_1 \parallel y_2} \quad (15)$$

for each $a \in A$ and for each $r, r_1, r_2, W \in \mathbb{R}^+$ such that $W = \min(r_1, r_2)$. This, according to Definition 2, is the communication operator of the original definition of stochastic π -calculus [20], with the minimal rate law (5) used. For example, consider processes P, Q as in (13). The process $P \parallel Q$ can make the following transitions:

$$\begin{aligned} P \parallel Q &\xrightarrow{a,1} P_1 \parallel Q & P \parallel Q &\xrightarrow{a,3} P_2 \parallel Q & P \parallel Q &\xrightarrow{\bar{a},2} P \parallel Q_1 \\ &P \parallel Q \xrightarrow{\tau, \frac{1}{2}} P_1 \parallel Q_1 & & P \parallel Q \xrightarrow{\tau, \frac{3}{2}} P_2 \parallel Q_1. \end{aligned}$$

Alternatively, one could use the same rules with $W = r_1 \cdot r_2$. This would correspond to the semantics of parallel composition in the biological stochastic π -calculus [22], with the mass action law (6) used. For example, the process $P \parallel Q$ above can then make the following transitions:

$$\begin{aligned} P \parallel Q &\xrightarrow{a,1} P_1 \parallel Q & P \parallel Q &\xrightarrow{a,3} P_2 \parallel Q & P \parallel Q &\xrightarrow{\bar{a},2} P \parallel Q_1 \\ &P \parallel Q \xrightarrow{\tau,2} P_1 \parallel Q_1 & & P \parallel Q \xrightarrow{\tau,6} P_2 \parallel Q_1. \end{aligned}$$

Example 5. Several non-standard, yet meaningful stochastic operators can be defined within the SGSOS format. For example, consider unary “*catalyst*” and “*inhibitor*” operators cat_a and inh_a for each $a \in A$, which influence rates of process transitions; they can be seen as stochastic counterparts of the restriction operator of CCS. Their semantics is defined by the rules:

$$\begin{array}{c} \frac{x \xrightarrow{a@r} y}{\text{cat}_a(x) \xrightarrow{a@2r} \text{cat}_a(y)} \qquad \frac{x \xrightarrow{a@r} y}{\text{inh}_a(x) \xrightarrow{a@r/2} \text{inh}_a(y)} \\[1em] \frac{x \xrightarrow{b@r} y}{\text{cat}_a(x) \xrightarrow{b@r} \text{cat}_a(y)} \qquad \frac{x \xrightarrow{b@r} y}{\text{inh}_a(x) \xrightarrow{b@r} \text{inh}_a(y)} \end{array}$$

for each $r \in \mathbb{R}^+$ and $a, b \in A$ such that $b \neq a$. For example, in the derived stochastic transition system we find the transition $\text{cat}_a((a, 2).\text{nil}) \xrightarrow{a,4} \text{cat}_a(\text{nil})$. Since the above rules conform to the SGSOS format, it is immediate that operators cat_a and inh_a preserve stochastic bisimilarity.

Another example is a binary operator $!!$ of “unfair race parallel composition,” which only allows transitions from processes with higher apparent rates than their competitors. Formally, its semantics is defined by rules

$$\frac{x_1 \xrightarrow{a@r_1} y \quad x_2 \xrightarrow{a@r_2} y}{x_1 !! x_2 \xrightarrow{a@r_1} y !! x_2} \qquad \frac{x_1 \xrightarrow{a@r_2} y \quad x_2 \xrightarrow{a@r_1} y}{x_1 !! x_2 \xrightarrow{a@r_1} x_1 !! y}$$

for each $a \in A$ and $r_1, r_2 \in \mathbb{R}_0^+$ such that $r_1 > r_2$. For example, the process $P = ((a, 2).Q) !! ((a, 3).T)$ has only one outgoing transition $P \xrightarrow{a,3} ((a, 2).Q) !! T$. Again, stochastic bisimilarity is immediately compositional with respect to $!!$. This example illustrates the fact that in the semantics of SGSOS operators, apparent rates (3) of sub-processes can be tested, compared and used in an arbitrary fashion. This is in contrast

with formats defined for probabilistic systems [2, 16], where probabilities of transitions can be used in a very restricted manner. Note however that in SGSOS rates of *single* transitions of subprocesses cannot be used entirely freely. For example, it is not possible to write SGSOS semantics of a hypothetical unary operator $\text{even}(\cdot)$ that would propagate transitions with even rates and suppress those with odd rates. Indeed, such an operator would not preserve stochastic bisimilarity. However, one can define an SGSOS operator that propagates only transitions with labels whose apparent rates are even.

6 Associative parallel composition for stochastic systems

In this section we address an issue in the original design of the stochastic π -calculus [20], which to our knowledge has not yet been addressed in the literature. Namely, if the minimal rate law (5) is used in the definition (7), then the CCS-style communication operator $\parallel$ is not associative up to stochastic bisimilarity. Indeed, consider processes

$$\begin{aligned} P_1 &= (a, r).\text{nil} & P_2 &= (\bar{a}, r).\text{nil} \\ Q_1 &= (P_1 \parallel P_1) \parallel P_2 & Q_2 &= P_1 \parallel (P_1 \parallel P_2). \end{aligned}$$

Note that $r_a(P_1) = r$, $r_a(P_1 \parallel P_1) = 2r$, and $r_{\bar{a}}(P_2) = r_{\bar{a}}(P_1 \parallel P_2) = r$. This means that, in the derived proved-transitions

$$\begin{aligned} Q_1 &\xrightarrow{\langle \parallel_1(a, r), (\bar{a}, r) \rangle, R_1} (\text{nil} \parallel P_1) \parallel \text{nil} \\ Q_2 &\xrightarrow{\langle (a, r), \parallel_2(\bar{a}, r) \rangle, R_2} \text{nil} \parallel (P_1 \parallel \text{nil}), \end{aligned}$$

one has $R_1 = \min(2r, r) \cdot \frac{r}{2r} \cdot \frac{r}{r} = \frac{r}{2}$ and $R_2 = \min(r, r) \cdot \frac{r}{r} \cdot \frac{r}{r} = r$, hence in the resulting RTS, processes Q_1 and Q_2 do the corresponding τ -transitions respectively with rates $r/2$ and r . As a result, they are not stochastic bisimilar. On the other hand, the same operator $\parallel$ with the rate calculation formula changed to the law of mass action (6), as in [22], is associative. Moreover, CSP-style synchronisation as used in PEPA is associative for both minimal rate and mass action laws.

In the following, we consider parallel composition within the framework of SGSOS and characterise those rate formulas for which CCS-style communication and CSP-style synchronisation operators are associative up to stochastic bisimilarity. It turns out that the CSP-style composition gives much more freedom in the choice of rate formula.

6.1 CCS-style communication

Consider the language of Example 4, extending those of Examples 1 and 2. Two versions of the language were mentioned there, depending on the choice of the family of rules of type (15) used in the semantics: one where $W = \min(r_1, r_2)$ (the *minimal rate* law) and one where $W = r_1 \cdot r_2$ (the *mass action* law). We will now characterise those “laws” that give rise to an associative operator $\parallel$. More formally, we assume that for each pair $r_1, r_2 \in \mathbb{R}_0^+$ there is exactly one rule of the type (15) in our semantics for each label a , and that, moreover, the number W in the conclusion of the rules does not depend on a ; we can thus treat the W ’s as a function $W : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$. We then look for those rate functions W for which the operator $\parallel$ is associative up to stochastic bisimilarity.

As the following theorem shows, the choice of W is severely limited: the mass action law is essentially the only choice that makes $\parallel$ associative.

Theorem 3. In the situation described above, $\parallel$ is associative up to stochastic bisimilarity if and only if $W(r_1, r_2) = c \cdot r_1 \cdot r_2$ for some constant $c \in \mathbb{R}^+$.

6.2 CSP-style synchronisation

Consider now the language of Example 3, extending those of Examples 1 and 2. Again, assume that for each pair $r_1, r_2 \in \mathbb{R}_0^+$ there is exactly one rule of the type (12) for each label a , and that the number W in the conclusion of the rules does not depend on a ; thus, as before, we have a function $W : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$. It turns out that, compared to §6.1, one has considerably more freedom in choosing W so that each of the synchronisation operators $\boxtimes_L$ is associative:

Theorem 4. In the situation described above, each $\boxtimes_L$ is associative up to stochastic bisimilarity if and only if W is associative, i.e., $W(r_1, W(r_2, r_3)) = W(W(r_1, r_2), r_3)$ for all $r_1, r_2, r_3 \in \mathbb{R}^+$.

7 Conclusions and future work

We have defined *SGSOS*, a congruence format for structural operational descriptions of discrete space, continuous time Markov chains. Stochastic bisimilarity is guaranteed to be compositional on languages defined by SGSOS rules. Standard operators of Markovian process algebras, such as prefixing, choice and various forms of synchronization, as well as plenty of non-standard, yet potentially useful operators, are definable in SGSOS. The format arises naturally from the abstract theory of well-behaved operational semantics, based on bialgebras and distributive law.

SGSOS is similar to formats for reactive probabilistic systems developed in [2, 16]. Apart from syntactic sugar, the most important difference is the treatment of apparent rates, absent in the probabilistic setting. Rates of single transitions (and their conditional probabilities) are treated in SGSOS just as probabilities of transitions are in [2, 16]. In [16], additional complication is necessary to cater for *generative* probabilistic systems. The notions of reactive and generative RTSs coincide, and the additional complexity is not needed in SGSOS.

This is only an initial study of a theory of well-behaved stochastic operational semantics, and several research directions are left open. Look-ahead premises are not allowed in SGSOS, unlike in the probabilistic formats of [16]. Recursive definitions or the name-binding features of stochastic π -calculus are not currently supported; to treat the latter correctly, one should combine SGSOS with techniques from [10]. Also, non-Markovian processes are not treated here, as a coalgebraic treatment of them is missing. Process algebra for continuous-space Markov chains [9] is another possible direction.

References

1. L. Aceto, W. J. Fokkink, and C. Verhoef. Structural operational semantics. In J. A. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, pages 197–292. Elsevier, 2002.
2. F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. PhD dissertation, CWI, Amsterdam, 2004.
3. M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theor. Comp. Sci.*, 202(1-2):1–54, 1998.
4. B. Bloom, S. Istrail, and A. Meyer. Bisimulation can’t be traced. *Journal of the ACM*, 42:232–268, 1995.
5. S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31:560–599, 1995.
6. M. Calder, S. Gilmore, and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways. In *Procs. CMSB’05*, pages 204–215, 2005.
7. E. P. de Vink and J. J. M. M. Rutten. Bisimulation for probabilistic transition systems: A coalgebraic approach. *Theoretical Computer Science*, 221(1-2):271–293, 1999.
8. P. Degano and C. Priami. Enhanced operational semantics. *ACM Comput. Surv.*, 28(2):352–354, 1996.
9. J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Information and Computation*, 179:163–193, 2002.
10. M. Fiore and S. Staton. A congruence rule format for name-passing process calculi from mathematical structural operational semantics. In *Proc. LICS’06*, pages 49–58. IEEE Computer Society Press, 2006.
11. N. Götz, U. Herzog, and M. Rettelsbach. Multiprocessor and distributed system design: The integration of functional specification and performance analysis using stochastic process algebras. In *Performance/SIGMETRICS Tutorials*, pages 121–146, 1993.
12. H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002.
13. J. Hillston. On the nature of synchronisation. In *Procs. PAPM’94*, pages 51–70, 1994.
14. J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
15. J. Hillston. Process algebras for quantitative analysis. In *Procs. LiCS’05*, pages 239–248. IEEE Computer Society Press, 2005.
16. R. Lanotte and S. Tini. Probabilistic bisimulation as a congruence. *ACM Trans. Comp. Logic*, to appear, 2008.
17. K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
18. R. Milner. A calculus of communicating systems. *Journal of the ACM*, 1980.
19. G. D. Plotkin. A structural approach to operational semantics. DAIMI Report FN-19, Computer Science Department, Aarhus University, 1981.
20. Corrado Priami. Stochastic π -calculus. *Computer Journal*, 38(7):578–589, 1995.
21. Corrado Priami. Language-based performance prediction for distributed and mobile systems. *Information and Computation*, 175(2):119–145, 2002.
22. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π -calculus process algebra. In *Proc. Pacific Symp. Biocomp.*, 2001.
23. J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.
24. D. Turi and G. D. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS’97*, pages 280–291. IEEE Computer Society Press, 1997.