Structural Operational Semantics for Stochastic and Weighted Transition Systems

Bartek Klin*, Vladimiro Sassone$^\diamond$
*University of Warsaw $^\diamond$University of Southampton

Abstract
We introduce weighted GSOS, a general syntactic framework to specify well-behaved transition systems where transitions are equipped with weights coming from a commutative monoid. We prove that weighted bisimilarity is a congruence on systems defined by weighted GSOS specifications. We illustrate the flexibility of the framework by instantiating it to handle some special cases, most notably that of stochastic transition systems. Through examples we provide weighted-GSOS definitions for common stochastic operators in the literature.

1 Introduction

Process algebras such as CCS [19] or CSP [6] 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) [23], where labelled nondeterministic transition systems (LTSs) are defined by induction on the syntactic structure of processes. Formalisms for SOS decriptions 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.

Already from the original paper that introduced SOS [23, 24] it is apparent that for all but the simplest applications, one needs to consider systems where transitions carry more information than simple, unstructured labels. To model various aspects of computation, one endows transitions with information on fresh and bound names [29], transition probabilities [17] or durations [20], memory states, environments and so on. Crucially, the additional structure put on transitions influences the corresponding notion of process equivalence, so the simple theory of SOS and bisimilarity cannot be directly applied to these extended specification frameworks.

An important example is the theory of stochastic process algebras that have been deployed for applications in performance evaluation and in systems biology. There, 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 [10], PEPA [14], EMPA [4], and stochastic π-calculus [25]. Semantics of these calculi have been given by variants of the SOS approach. However, SOS formalisms used there were not based on any general framework for operational descriptions of stochastic processes, and indeed differed substantially from one another. This is unfortunate, as such a framework makes languages easier to understand, compare, and extend. Specifically, a format for SOS descriptions which guarantees the compositionality of stochastic bisimilarity, makes 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 [14] and the stochastic π-calculus [25]. As the operational techniques used there are hard to extend to a general format for well-behaved stochastic specifications, we resolve to adapt the bialgebraic theory of well-behaved SOS, based on category theory and developed by Turi and Plotkin [30]. The inspiration for our approach comes directly from results by F. Bartels, who designed a congruence format for probabilistic transition systems [3].

In fact, we develop SGSOS as a special case of a more general format that applies to a wide class of systems called *weighted transition systems*, where every labelled transition is associated with a weight drawn from a commutative monoid $\mathfrak{W}$. The monoid structure determines the way in which weights of alternative transitions combine. Weighted transition systems generalise both ordinary LTS and stochastic transition systems, as well as other potentially useful kinds of systems. The abstract bialgebraic approach leads us to a rather general format parameterised by the monoid $\mathfrak{W}$, the $\mathfrak{W}$-GSOS format. In order to study $\mathfrak{W}$-GSOS we first provide a uniform coalgebraic treatment of weighted transition systems, including a concrete definition of $\mathfrak{W}$-weighted bisimulation. Both SGSOS and the well-known GSOS format [5] for LTS specifications, arise as instances of $\mathfrak{W}$-GSOS.

The structure of the paper is as follows. In §2 we recall from the literature the basic notions underlying our exposition: labelled transition systems and GSOS rules, rated transition systems and stochastic bisimulation. We discuss the main existing approaches to modelling stochastic systems and their shortcomings. In §3 we briefly recall Turi and Plotkin's and Bartel's seminal work to recast GSOS rules in terms of structures which are at the same time algebraic and coalgebraic (bialgebras). Our own development starts in §4, where we introduce weighted transition systems, a notion general enough to encompass both non-deterministic and rated transition systems and support our entire theory. In §5 we introduce weighted GSOS, $\mathfrak{W}$-GSOS, an abstract, parametric rule framework, which in §6 we instantiate to several interesting cases, including SGSOS. The flexibility of the framework is illustrated in §7, where we put it at work to formalise some well-known stochastic operators. Finally, §8 illustrates a meta-theoretic application of SGSOS, as we address a basic issue in the original design of the stochastic pi-calculus, and study which rate functions are compatible with an observational equivalence approach to its semantics.

The paper is self-contained, although a general acquaintance with coalgebras and with Turi and Plotkin's work [30] is beneficial to fully appreciate §3 and §4 as well as some of our proofs.

This paper is a full and extended version of the extended abstracts appeared as [16] and [15]. With respect to *loc. cit.*, here we include all the proofs, which are not easily derivable from the short versions, enrich our set of instances and examples, and provide a more comprehensive discussion of related work and bibliography. We believe that this integrated presentation of two separated, yet related results, affords greater generality and, therefore, provides a significant added-value to the reader.

2 Transition systems and process calculi

We begin our development by presenting previously studied approaches to defining SOS for nondeterministic systems such as CCS, and for Markovian process algebras. We also explain some deficiencies of the latter, which motivate our search for a new SOS format for stochastic systems. Further, we introduce weighted transition systems and show how both nondeterministic and stochastic systems are instances of them.

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 over some algebraic signature, and transition relations are defined inductively, by means of inference rules. For example, in a well-known fragment of CCS [19], processes are terms over the grammar $P ::= \mathtt{nil} \mid a.P \mid P + P \mid P \parallel P$, and the LTS is induced from the following inference rules:

$$\begin{array}{ccc}
\frac{}{a.\mathtt{x} \overset{a}{\rightarrowtriangle} \mathtt{x}} & \frac{\mathtt{x}_1 \overset{a}{\rightarrowtriangle} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{y}} & \frac{\mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{y}} \\[2ex]
\frac{\mathtt{x}_1 \overset{a}{\rightarrowtriangle} \mathtt{y}}{\mathtt{x}_1 \parallel \mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{y} \parallel \mathtt{x}_2} & \frac{\mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{y}}{\mathtt{x}_1 \parallel \mathtt{x}_2 \overset{a}{\rightarrowtriangle} \mathtt{x}_1 \parallel \mathtt{y}} & \frac{\mathtt{x}_1 \overset{a}{\rightarrowtriangle} \mathtt{y}_1 \quad \mathtt{x}_2 \overset{\bar{a}}{\rightarrowtriangle} \mathtt{y}_2}{\mathtt{x}_1 \parallel \mathtt{x}_2 \overset{\tau}{\rightarrowtriangle} \mathtt{y}_1 \parallel \mathtt{y}_2}
\end{array} \tag{1}$$

The induced LTS consists of all transitions $\mathtt{t} \xrightarrow{a} \mathtt{s}$ such that $\mathtt{t} \overset{a}{\rightarrowtriangle} \mathtt{s}$ can be inferred from the rules. (In this context, usually one does not distinguish between the two kinds of arrows $\longrightarrow$ and $\rightarrowtriangle$, see e.g. [1]; however, as we shall see, for weighted system specifications this is rather useful.)

Plenty of operators can be defined formally by such rules. 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.

We shall use standard terminology and notation related to algebraic signatures, terms and substitutions. A *signature* Σ is a set of operation symbols (also denoted Σ) with an arity function $ar : \Sigma \to \mathbb{N}$, usually left implicit. The set of Σ-terms with variables from a set X is denoted $T_\Sigma X$; in particular, $T_\Sigma \emptyset$ denotes the set of closed terms. For a function $\sigma : X \to Y$, $\sigma[-] : T_\Sigma X \to T_\Sigma X$ denotes its extension to terms, defined by variable substitution.

Fix a countably infinite set $\Xi \ni \mathtt{x}, \mathtt{y}, \mathtt{z}, \ldots$ of variables. A *GSOS inference rule* [5] over a signature Σ and a set of labels A is an expression of the form

$$\frac{\{\mathtt{x}_i \xrightarrow{a_{ij}} \mathtt{y}_{ij}\}^{1\leq i\leq n}_{1\leq j\leq m_i} \qquad \{\mathtt{x}_i \not\xrightarrow{b_{ik}}\}^{1\leq i\leq n}_{1\leq k\leq l_i}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \xrightarrow{c} \mathtt{t}} \tag{2}$$

where $\mathtt{f} \in \Sigma$, $n = ar(\mathtt{f})$, $m_i, l_i \in \mathbb{N}$, $a_{ij}, b_{ik}, c \in A$, $\mathtt{t} \in T_\Sigma \Xi$, $\mathtt{x}_i$ and $\mathtt{y}_{ij} \in \Xi$ are all distinct and no other variables occur in the term $\mathtt{t}$. Expressions above the horizontal line in a GSOS rule are called its *premises*, and the expression below it is the *conclusion*. As defined formally below, a GSOS rule infers a transition for an $\mathtt{f}$-term with label c inductively from the existence of each of a set of a_{ij}-transitions and the non-existence of any of a set of b_{ik}-transitions for each of the argument $\mathtt{x}_i$ of $\mathtt{f}$. Sources and targets of transitions in the rule premises can only be specified parametrically via simple variables, which may or may not appear in the target of the defined transition $\mathtt{t}$. A *GSOS specification* is a set of GSOS rules; it is *image-finite* if it contains only finitely many rules for each $\mathtt{f}$ 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 (ground) term $s = \mathtt{f}(t_1, \ldots, t_n) \in T_\Sigma 0$, one adds a transition $s \xrightarrow{c} t$ for each substitution $\sigma : \Xi \to T_\Sigma 0$ such that for some rule $r \in \Lambda$ as in (2), there is $\sigma \mathtt{x}_i = t_i$, $\sigma \mathtt{t} = t$, and σ *satisfies* all premises of r, meaning that for each premise $\mathtt{x} \xrightarrow{a} \mathtt{y}$ there is $\sigma(\mathtt{x}) \xrightarrow{a} \sigma(\mathtt{s})$, and for each premise $\mathtt{y} \not\xrightarrow{a}$ there is no $t \in T_\Sigma 0$ for which $\sigma(\mathtt{y}) \xrightarrow{a} t$.

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 is expressed succinctly by saying that GSOS is a *congruence format* for bisimilarity on LTSs. Moreover, it is easy to prove by structural 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 $\rho : X \times A \times X \to \mathbb{R}^+_0$ 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. (A detailed presentation of CTMCs, the intuitions behind them, and their presentation by transition rates is beyond the scope of this paper; for for information see, e.g., [11, 14, 25].) For the sake of readability we will write $\rho(x \xrightarrow{a} y)$ for $\rho(x, a, y)$, and $x \xrightarrow{a,r} y$ will often denote $\rho(x \xrightarrow{a} y) = r$. 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

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

exists for each $x \in X$ and $a \in A$; it will be called the *apparent rate* of label a in state x. This is an important classification, as the discrete-space processes likely to be considered in practice are typically image-finite.

To avoid the problem of state space explosion in computations on Markov processes, various equivalence relations on states have been considered. Of those, the most significant is *stochastic bisimilarity* (called strong equivalence in [13], 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 a 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 here two of these variants: the *multi-transition* system approach used for the stochastic calculus PEPA [13, 14], and the *proved* SOS approach of the stochastic π-calculus [25, 26, 27].

2.2.1 Multi-transition systems

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

$$P ::= \mathtt{nil} \mid (a, r).P \mid P + P \mid P \underset{L}{\bowtie} P$$

where a ranges over a fixed set A of labels, L over subsets of A, and r over $\mathbb{R}^+$. Their semantics is defined in [13] by the inference rules:

$$\begin{gathered}
\frac{}{(a, r).\mathtt{x} \xrightarrow{a,r} \mathtt{x}} \qquad \frac{\mathtt{x}_1 \xrightarrow{a,r} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{a,r} \mathtt{y}} \qquad \frac{\mathtt{x}_2 \xrightarrow{a,r} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{a,r} \mathtt{y}} \\[2ex]
\frac{\mathtt{x}_1 \xrightarrow{a,r} \mathtt{y}}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{a,r} \mathtt{y} \underset{L}{\bowtie} \mathtt{x}_2} \qquad \frac{\mathtt{x}_2 \xrightarrow{a,r} \mathtt{y}}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{a,r} \mathtt{x}_1 \underset{L}{\bowtie} \mathtt{y}} \quad (a \notin L) \\[2ex]
\frac{\mathtt{x}_1 \xrightarrow{a,r_1} \mathtt{y}_1 \quad \mathtt{x}_2 \xrightarrow{a,r_2} \mathtt{y}_2}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{a,R} \mathtt{y}_1 \underset{L}{\bowtie} \mathtt{y}_2} \quad (a \in L)
\end{gathered} \tag{4}$$

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* $\underset{L}{\bowtie}$ for each set L of labels. These operators are based on CSP-style synchronisation [6] rather than CCS-style communication [19].

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*, according to which a process $P + P$ can perform the same transitions as P, but with twice the rates. In particular, the processes P and $P + P$ should *not* be stochastic bisimilar in general. This is why the semantics of PEPA is given as a *multi-transition* system labelled with pairs $(a, r) \in A \times \mathbb{R}^+$, which is a transition system whose transition relation is a *multi*set of triples $(x, (a, r), y)$. To define it, the rules in (4) are interpreted similarly as the GSOS rules in §2.1, but with the multiplicity of a transition determined by counting all its different derivations. To obtain an RTS (or, equivalently, a labelled Markov process) from the multi-transition system so induced by the rules, one then removes multiplicities by summing up all their rates on multiple transition in single rated transitions. This gives $P + P$ twice the rates of P. For example, the process $(a, 3).\mathtt{nil} + (a, 3).\mathtt{nil}$ in the induced multi-transition systems has two identical transitions to $\mathtt{nil}$ with label $(a, 3)$, whilst in the final RTS can make a single transition to $\mathtt{nil}$ with label a and rate 6. For more details of this construction, see [13].

The formula for calculating R based on r_1 and r_2 in the last rule of (4) is parametric and depends on the intended meaning of synchronisation. In applications to performance evaluation [13], the formula

$$R = \min(r_a(\mathtt{x}_1), r_a(\mathtt{x}_2)) \cdot \frac{r_1}{r_a(\mathtt{x}_1)} \cdot \frac{r_2}{r_a(\mathtt{x}_2)} \tag{5}$$

is the natural choice. We call it the *minimal rate law*, since in the resulting RTS, the rate of a transition $P \underset{L}{\bowtie} Q \xrightarrow{a} P' \underset{L}{\bowtie} Q'$ (with $a \in L$) is the least of the rates of $P \xrightarrow{a} P'$ and $Q \xrightarrow{a} Q'$ (i.e., the composed systems is as performant as its 'slowest' component). For applications to systems biology, a more convenient choice is instead

$$R = r_1 \cdot r_2, \tag{6}$$

which following [7] we call the *mass action law*. The rate of $P \underset{L}{\bowtie} Q \xrightarrow{a} P' \underset{L}{\bowtie} Q'$ (with $a \in L$) here is the product of the rates of $P \xrightarrow{a} P'$ and $Q \xrightarrow{a} Q'$ (which e.g. reflects the frequency of reactions in chemical solutions). For an intuitive description of these and other similar formulae, see [12].

2.2.2 Proved transition systems

A different approach was used to define semantics of stochastic π-calculus [25, 26]. 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:

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

where a ranges over a fixed set A of labels, and r over $\mathbb{R}^+$. For the semantics, the authors of [25] 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 need to enrich 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 the following rules:

$$\frac{}{(a,r).\mathtt{x} \xrightarrow{(a,r)} \mathtt{x}} \qquad \frac{\mathtt{x}_1 \xrightarrow{\theta} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{+_1\theta} \mathtt{y}} \qquad \frac{\mathtt{x}_2 \xrightarrow{\theta} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{+_2\theta} \mathtt{y}} \tag{7}$$

$$\frac{\mathtt{x}_1 \xrightarrow{\theta} \mathtt{y}}{\mathtt{x}_1 \| \mathtt{x}_2 \xrightarrow{\|_1\theta} \mathtt{y} \| \mathtt{x}_2} \qquad \frac{\mathtt{x}_2 \xrightarrow{\theta} \mathtt{y}}{\mathtt{x}_1 \| \mathtt{x}_2 \xrightarrow{\|_2\theta} \mathtt{x}_1 \| \mathtt{y}} \qquad \frac{\mathtt{x}_1 \xrightarrow{\theta_1(a,r_1)} \mathtt{y}_1 \quad \mathtt{x}_2 \xrightarrow{\theta_2(\bar{a},r_2)} \mathtt{y}_2}{\mathtt{x}_1 \| \mathtt{x}_2 \xrightarrow{\langle \|_1\theta_1(a,r_1), \|_1\theta_2(\bar{a},r_2)\rangle, R} \mathtt{y}_1 \| \mathtt{y}_2}$$

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

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

and where R depends on r_1 and r_2 according to the minimal rate law [25] or the mass action law [27], as in PEPA.

These rules are then used to induce an LTS, which results in relatively complex labels. To obtain an RTS (called a continuous time Markov chain, CTMC, in [25]), 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 $P = (a,3).\mathtt{nil} + (a,3).\mathtt{nil}$ in the induced LTS can make two distinct transitions $P \overset{+_1(a,3)}{\longrightarrow} \mathtt{nil}$ and $P \overset{+_2(a,3)}{\longrightarrow} \mathtt{nil}$, and in the final RTS it can make a transition to $\mathtt{nil}$ with label a and rate 6.

2.2.3 Criticism

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 $\mathtt{f}$ with semantics defined by a rule:

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

and see that, although $(a,2).\mathtt{nil} + \mathtt{nil}$ and $\mathtt{nil} + (a,2).\mathtt{nil}$ are stochastic bisimilar, they are not so when put in context $\mathtt{f}(-)$, 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 [9], 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 raised against 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

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

and see that, although processes $(a,3).\mathtt{nil} + (a,3).\mathtt{nil}$ and $(a,6).\mathtt{nil}$ are stochastic bisimilar, they are not so in the context $\mathtt{f}(-)$. 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. 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 [30]. To this end, we shall cast both nondeterministic and stochastic systems in a more general framework of weighted transition systems, defined below in §4.

2.2.4 Other approaches

Recently, interesting alternative approaches to operational specification of stochastic systems have been proposed [8, 21]. Their common theme is the use of premises of the type $x \xrightarrow{a} \mu$, where μ is a measure on the space of processes, rather than a single process. A similar idea was used earlier in the context of probabilistic systems, to specify so-called Segala automata (see [3]). With this richer kind of rule pemises and conclusions, it is possible to design elegant formats that guarantee the compositionality of stochastic bisimilarity. These formats are best suited for systems that combine stochastic aspects with nondeterminism. The price to pay for their relative simplicity is that the basic observations on systems, signified by rule premises, are more complex. Although this represents a significant innovation worth considering, in this paper we remain concerned with specifications based on very elementary process statements, where two processes are related with an observable label and, possibly, a quantitative rate or weight.

3 An abstract approach to SOS

Our approach to a weighted counterpart of the GSOS framework of §2.1 is based on a categorical generalisation of GSOS, developed by Plotkin and Turi in [30]. In this section we briefly recall that work; in the rest of the paper we develop a syntactic format for weighted SOS as an instance of the general framework. For basic categorical notions left unexplained here, see e.g. [18].

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 $h : X \to (\mathcal{P}_\omega X)^A$ (here, $\mathcal{P}_\omega$ is the finite powerset construction), along the intuitive correspondence:

$$y \in h(x)(a) \iff x \xrightarrow{a} y.$$

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 $h : X \to BX$ (the *structure*). Thus image-finite LTSs are coalgebras for the functor $(\mathcal{P}_\omega-)^A$.

A B-coalgebra *morphism* from a $h : X \to BX$ to $g : Y \to BY$ is a function $f : X \to Y$ such that the equation $g \circ f = Bf \circ h$ holds. This notion provides a general coalgebraic treatment of process equivalences: processes $x, y \in X$ are *observationally equivalent* with respect to $h : X \to BX$ if they are equated by some coalgebra morphism from h. For example, for $B = (\mathcal{P}_\omega-)^A$, observational equivalence specialises to the well-known notion of LTS bisimulation [22, 19]. For more information about the coalgebraic approach to process theory, see [28].

In the context of SOS, processes typically are closed terms over some algebraic signature, i.e., a set $\Sigma \ni \mathtt{f}, \mathtt{g}, \ldots$ of operation symbols with an arity function $ar : \Sigma \to \mathbb{N}$. Such a signature corresponds to a functor $\Sigma X = \coprod_{\mathtt{f}\in\Sigma} X^{ar(\mathtt{f})}$ 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 $g : \Sigma X \to X$ (the *structure*). An algebra morphism from $g : \Sigma X \to X$ to $h : \Sigma Y \to Y$ is a function $f : X \to Y$ such that $f \circ g = h \circ \Sigma f$.

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 $a : \Sigma T_\Sigma 0 \to T_\Sigma 0$ for the *functor* Σ corresponding to the signature. This Σ-algebra is *initial*: for any $g : \Sigma X \to X$ there is a unique algebra morphism

$$g^\sharp : T_\Sigma 0 \to X$$

from a to g. The construction T_Σ is also a functor, called the *free monad* over Σ.

In [30], 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 [3]), that image finite GSOS specifications are in an essentially one-to-one correspondence with *distributive laws*, i.e., natural transformations of the type

$$\lambda : \Sigma(\mathrm{Id} \times B) \Longrightarrow BT_\Sigma \tag{8}$$

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) maps 'structural' combinations (viz., Σ) of behaviours (viz., B) to the behaviour of terms (viz., BT_Σ), which is the *essence* of a SOS rule. Here Id accounts for subterms that stay idle in a transition. Moreover, any λ as above gives rise to a B-coalgebra structure on $T_\Sigma 0$, defined by a 'structural recursion theorem' (see [30]) as the only function h_λ that makes the following diagram commute:

$$\begin{tikzcd}
T_\Sigma 0 \arrow[d, "h_\lambda"'] & & \Sigma T_\Sigma 0 \arrow[ll, "a"'] \arrow[d, "\Sigma\langle \mathrm{id}, h_\lambda \rangle"] \\
B T_\Sigma 0 & B T_\Sigma T_\Sigma 0 \arrow[l, "Ba^\sharp"] & \Sigma(T_\Sigma 0 \times B T_\Sigma 0) \arrow[l, "\lambda_X"]
\end{tikzcd} \tag{9}$$

where a is the initial Σ-algebra. A pair (a, h_λ) for which the diagram commutes is called *λ-bialgebra*.

Note that h_λ is an LTS having the set of closed Σ-terms as carrier; this turns out to coincide with the LTS induced from the GSOS specification corresponding to λ. Moreover, 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 ([30], Cor. 7.5)**.** If B has a final coalgebra then for any λ as in (8), observational equivalence on $h_\lambda : T_\Sigma 0 \to B T_\Sigma 0$ is a congruence on $T_\Sigma 0$.

This result is the basis of our search for a congruence format for stochastic systems. We start below by defining a suitable notion of weighted transition systems general enough to underpin our subsequent development.

4 Weighted transition systems and their coalgebras

A *commutative monoid* $\mathfrak{W} = (W, +, 0)$ is a set equipped with a binary, associative, commutative operation called *addition* with a unit called *zero*. The addition operation is extended in an obvious way to summation $\sum$ on arbitrary finite (multi)sets of elements of W (in particular, the empty sum is defined to be 0).

For any set X, a function $f : X \to W$ is *finitely supported* if $f(x) \neq 0$ for only finitely many $x \in X$. Such functions can be extended to arbitrary subsets: for any $C \subseteq X$, define

$$f(C) = \sum_{x \in C} f(x).$$

The sum is well defined if f is finitely supported. This notation extends to multi-argument functions in an obvious way, i.e., $g(x, C) = \sum_{y \in C} g(x, y)$ etc.

The following definition assumes a commutative monoid $\mathfrak{W} = (W, 0, +)$. Elements of W will be called *weights* and denoted $v, w, \ldots$.

Definition 2. A *$\mathfrak{W}$-weighted labelled transition system* ($\mathfrak{W}$-LTS in short) is a triple (X, A, ρ) where

- X is a set of *states* (or *processes*),
- A is a set of *labels*,
- $\rho : X \times A \times X \to W$ is called the *weight function*.

For each $x \in X$ and $a \in A$, the *total weight* of a at x is defined as

$$\rho(x, a, X) = \sum_{y \in X} \rho(x, a, y).$$

To support intuitions based on classical labelled transition systems (LTSs), we shall write $\rho(x \xrightarrow{a} y)$ for $\rho(x, a, y)$, and to say that $\rho(x \xrightarrow{a} y) = w$ we shall write $x \xrightarrow{a,w} y$. Depending on $\mathfrak{W}$, the idea might be that w represents a frequency, probability, speed, or cost of occurrence, with 0-weighed transitions intuitively never occurring, i.e, de facto absent.

The latter notational convention suggests that weights can be understood as parts of labels. Indeed, $\mathfrak{W}$-LTSs labelled with A can be seen as ordinary LTSs labelled with $A \times W$, subject to a weight determinacy condition: for each $x, y \in X$ and $a \in A$, there is exactly one $w \in W$ for which $x \xrightarrow{a,w} y$.

Definition 3. A $\mathfrak{W}$-LTSs (X, A, ρ) is *image finite* if for each $x \in X$ and $a \in A$, the set of $y \in X$ such that $\rho(x \xrightarrow{a} y) \neq 0$ is finite.

In the following, we will restrict attention to image finite $\mathfrak{W}$-LTSs only.

In the definition of a $\mathfrak{W}$-LTSs, the monoid structure of $\mathfrak{W}$ was not used in any way. It is, however, crucial in the definition of *weighted bisimulation*:

Definition 4. Given a $\mathfrak{W}$-LTS (X, A, ρ), a $\mathfrak{W}$-*bisimulation* is an equivalence relation R on X such that for each $x, x' \in X$, $x \, R \, x'$ implies that for each $a \in A$ and each equivalence class C of R:

$$\rho(x, a, C) = \rho(x', a, C).$$

Processes $x, x' \in X$ are $\mathfrak{W}$-*bisimilar* if they are related by some $\mathfrak{W}$-bisimulation.

Note how the commutative monoid structure of $\mathfrak{W}$, together with the image finiteness assumption, ensures that the weights above are well-defined.

It is straightforward to see that $\mathfrak{W}$-weighted bisimulations are closed under (transitive closures of) arbitrary unions, hence $\mathfrak{W}$-bisimilarity on any LTS is the largest $\mathfrak{W}$-bisimulation on it.

Example 5. Consider the monoid of logical values $\mathcal{Y} = \{\mathtt{ff}, \mathtt{tt}\}$, with logical disjunction as + and $\mathtt{ff}$ as the zero element. As the special value zero singles out transitions which can never occur, $\mathcal{Y}$-LTSs are exactly ordinary (image-finite) LTSs, and $\mathcal{Y}$-bisimulations are classical bisimulations (more precisely, bisimulation equivalences).

Example 6. For $\mathbb{R}_0^+$ the monoid of nonnegative real numbers under addition, $\mathbb{R}_0^+$-LTSs are exactly RTSs defined in §2.2, and $\mathbb{R}_0^+$-bisimulations are exactly stochastic bisimulations.

Example 7. The set $\mathbb{R}^{+\infty}$ of positive real numbers augmented with positive infinity ∞, forms a commutative monoid with the minimum operation as addition and ∞ as the zero element. $\mathbb{R}^{+\infty}$-LTSs themselves are almost the same as $\mathbb{R}_0^+$-LTSs of Example 6, with the only difference in the capability of making transitions with weight ∞. However, the different monoid structures lead to different notions of weighted bisimulation and, as a result, to very different intuitions about the roles of weights in these systems. Indeed, while in Example 6 rates model the *capability* of a process to make a transition, with the idea that two similar capabilities add up to a stronger one, here weights might correspond to the *cost* of transitions, with the intuition that out of several similar possibilities, a process will always choose that of the lowest cost.

We now apply the theory of bialgebras illustrated in §3 to weighted transition systems. As before, we start with a commutative monoid $\mathfrak{W} = (W, 0, +)$. Let $\mathcal{F}_{\mathfrak{W}}X$ denote the set of all finitely supported functions from X to W. This extends to an endofunctor $\mathcal{F}_{\mathfrak{W}}$ on the category **Set** of sets and functions, with the action on functions defined by:

$$\mathcal{F}_{\mathfrak{W}}f(\phi)(y) = \sum_{x \in \overleftarrow{f}(y)} \phi(x)$$

for any $f : X \to Y$, $\phi \in \mathcal{F}_{\mathfrak{W}}X$ and $y \in Y$. (Here and in the following, we use $\overleftarrow{f}(y)$ for $\{x \in X \mid f(x) = y\}$.) It is easy to see that $\mathcal{F}_{\mathfrak{W}}f(\phi)$ is finitely supported if ϕ is so, and that $\mathcal{F}_{\mathfrak{W}}$ preserves identities and function composition.

Proposition 8. For any $\mathfrak{W}$, and any set A, coalgebras for the functor $(\mathcal{F}_{\mathfrak{W}}-)^A$ are in one-to-one correspondence with image-finite $\mathfrak{W}$-LTSs labelled with A.

Proof. Any $\mathfrak{W}$-LTS (X, A, ρ) determines a coalgebra $h : X \to (\mathcal{F}_{\mathfrak{W}}X)^A$ by $h(x)(a)(y) = \rho(x \xrightarrow{a} y)$. Image-finiteness of (X, A, ρ) means exactly that $h(x)(a)$ is finitely supported. This correspondence is bijective. □

Such coalgebraic understanding is justified by the treatment of weighted bisimilarity below.

Proposition 9. For (X, A, ρ) a $\mathfrak{W}$-LTS, an equivalence relation on X is a $\mathfrak{W}$-bisimulation if and only if it is the kernel relation of a coalgebra morphism from the corresponding $\mathcal{F}_{\mathfrak{W}}$-coalgebra. As a corollary, two processes are $\mathfrak{W}$-bisimilar if and only if they are observationally equivalent.

Proof. For $\mathfrak{W}$-LTSs (X, A, ρ) and (Y, A, θ), it is easy to check that a function $f : X \to Y$ is a morphism between the corresponding coalgebras if and only if, for each $x \in X$, $a \in A$ and $y \in Y$,

$$\theta(f(x), a, y) = \rho(x, a, \overleftarrow{f}(y)).$$

We show that a relation R is a $\mathfrak{W}$-bisimulation if and only if it is a kernel relation of such a morphism.

In the 'if' direction, assume $x, x' \in X$ such that $f(x) = f(x')$ for a coalgebra morphism f. Note that equivalence classes of the kernel relation $\ker(f)$ correspond bijectively to elements of Y in the image of f: for each equivalence class C there is a $y \in Y$ such that $C = \overleftarrow{f}(y)$. This implies that

$$\rho(x, a, C) = \theta(f(x), a, y) = \theta(f(x'), a, y) = \rho(x', a, C)$$

hence $\ker(f)$ is a $\mathfrak{W}$-bisimulation.

In the 'only if' direction, given a $\mathfrak{W}$-bisimulation R on (X, A, ρ), define a $\mathfrak{W}$-LTS on the set of R-equivalence classes $(X/R, A, \theta)$ by:

$$\theta(C, a, C') = \rho(x, a, C');$$

this is well-defined since, by Definition 4, $\rho(x, a, C') = \rho(y, a, C')$ for any $x, y \in C$. Furthermore, the quotient map $[-]_R : X \to X/R$ is a coalgebra morphism, since $\overleftarrow{[-]}_R(C) = C$. □

To apply the general machinery of bialgebraic operational semantics, the following technical result is needed:

Proposition 10. The functor $(\mathcal{F}_{\mathfrak{W}}-)^A$ admits a final coalgebra.

Proof. As proved in [2], it is enough to show that $\mathcal{F}_{\mathfrak{W}}$ is finitary, i.e. that for any set X and any $x \in \mathcal{F}_{\mathfrak{W}}X$ there is a finite subset $Y \subseteq X$ such that x arises as an element of $\mathcal{F}_{\mathfrak{W}}Y$. But this easily follows from the assumption that $\mathcal{F}_{\mathfrak{W}}X$ only contains finitely supported functions. □

Examples 5 and 6 can be revisited in coalgebraic terms as follows:

Example 11. For $\mathfrak{W} = y$ the monoid of logical values, $\mathcal{F}_{\mathfrak{W}}$ is naturally isomorphic to the covariant finite powerset functor $\mathcal{P}_\omega$, by mapping:

$$f : X \to y \qquad \mapsto \qquad \{x \in X \mid f(x) = \mathtt{tt}\} \subseteq X.$$

As a result, $(\mathcal{F}_{\mathfrak{W}}-)^A$ coalgebras are $(\mathcal{P}_\omega-)^A$-coalgebras, i.e., image-finite LTSs.

Example 12. For $\mathfrak{W} = \mathbb{R}_0^+$ the monoid of nonnegative real numbers under addition, $\mathcal{F}_{\mathfrak{W}}X$ is the set of all finitely supported functions from X to $\mathbb{R}_0^+$, and $(\mathcal{F}_{\mathfrak{W}}-)^A$-coalgebras are image-finite RTSs as defined in §2.2.

5 Weighted GSOS

From §3 and §4 it follows that as well-behaved compositional specifications of $\mathfrak{W}$-LTSs, one may take some syntactic entities (for example, sets of rules) that define natural transformations:

$$\lambda : \Sigma(\mathrm{Id} \times (\mathcal{F}_{\mathfrak{W}}-)^A) \Longrightarrow (\mathcal{F}_{\mathfrak{W}}T_\Sigma-)^A \tag{10}$$

where Σ is the process syntax signature endofunctor and T_Σ is the free monad over Σ. Moreover, for any such syntactic entity, weighted bisimilarity is a congruence for the WTS obtained from the corresponding λ by the inductive definition (9).

For $\mathfrak{W} = y$ (see Example 5), image-finite GSOS specifications [5] define (10), as noticed in [30] and proved in [3]. Moreover, every natural transformation of type (10) arises from a GSOS specification. We shall now generalise GSOS to a new rule format, which we call $\mathfrak{W}$-GSOS, parameterised by a commutative monoid $\mathfrak{W}$, and see how specifications that conform to the format give rise to natural transformations of type (10). Both GSOS and a new format SGSOS for specifying stochastic systems – our main motivating example – shall appear as special cases of this general format, as will be seen in §6.

In the following, fix an arbitrary commutative monoid $\mathfrak{W} = (W, 0, +)$. A function $\beta : W^k \to W$ is *multiadditive* if for each $i \in \{1, \dots, k\}$:

$$\begin{aligned} \beta(w_1, \dots, w_{i-1}, 0, w_{i+1}, \dots, w_k) &= 0, \\ \beta\Big(w_1, \dots, w_{i-1}, w_i + w_i', w_{i+1}, \dots, w_k\Big) &= \beta\Big(w_1, \dots, w_{i-1}, w_i, w_{i+1}, \dots, w_k\Big) \\ &\quad + \beta\Big(w_1, \dots, w_{i-1}, w_i', w_{i+1}, \dots, w_k\Big). \end{aligned}$$

To save space when the w_i are given by similar expressions which can be parameterised over i, we shall often write $\beta\,(w_i)_{i=1..k}$ for $\beta(w_1, \dots, w_k)$.

A good source of multiadditive functions are *semirings* (without 1), i.e. structures $(W, +, 0, \cdot)$ such that:

- $(W, +, 0)$ is a commutative monoid;
- $(W, \cdot)$ is a semigroup (i.e., $\cdot$ is associative but not necessarily commutative);
- $\cdot$ distributes over $+$:
    - $w \cdot (u + v) = (w \cdot u) + (w \cdot v)$
    - $(u + v) \cdot w = (u \cdot w) + (v \cdot w)$;
- 0 annihilates $\cdot$:
    - $0 \cdot w = w \cdot 0 = 0$.

Indeed in a semiring, for any $v \in W$, the function $\beta_v : W^k \to W$ defined by:

$$\beta_v\,(w_i)_{i=1..k} = v \cdot \prod_{i=1}^{k} w_i$$

is multiadditive, where $\prod$ is the obvious extension of $\cdot$ to finite sequences of elements of W.

Definition 13 ($\mathfrak{W}$-GSOS rule)**.** A $\mathfrak{W}$-*GSOS rule* for a signature Σ and a set A of labels is an expression of the form:

$$\frac{\Big\{\mathtt{x}_i \overset{a}{\longrightarrow\!\!\lhd} w_{a,i}\Big\}_{a \in D_i, 1 \le i \le n} \qquad \Big\langle \mathtt{x}_{i_j} \xrightarrow{b_j, u_j} \mathtt{y}_j \Big\rangle_{1 \le j \le k}}{\mathtt{f}(\mathtt{x}_1, \dots, \mathtt{x}_n) \xrightarrow{c, \beta(u_1, \dots, u_k)} \mathtt{t}} \tag{11}$$

where

- $\mathtt{f} \in \Sigma$ and $ar(\mathtt{f}) = n$, with $n, k \in \mathbb{N}$, and $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$;
- $\mathtt{x}_i$ and $\mathtt{y}_j$ are all distinct variables taken from a fixed countably infinite set Ξ, and no other variables appear in $\mathtt{t} \in T_\Sigma \Xi$; moreover, all variables $\mathtt{y}_j$ appear in $\mathtt{t}$;

- $D_i \subseteq A$;
- $w_{a,i} \in W$;
- $b_1, b_2, \dots, b_k, c \in A$;
- $u_1, \dots, u_k$ are pairwise distinct *weight variables* taken from a fixed countably infinite set Υ;
- $\beta : W^k \to W$ is a multiadditive function on $\mathfrak{W}$.

The set of variables from Ξ present in a rule R is denoted Ξ_R.

This is a complex definition; we now provide some terminology, notation and intuitions to aid the understanding of $\mathfrak{W}$-GSOS rules. The expression under the horizontal line in a rule is called the *conclusion*. The left side of the conclusion is called the *source* of a rule, and the right side is the *target*. Expressions above the horizontal line are *premises*. Each rule has premises of two kinds: *total weight* premises, depicted with $\longrightarrow\hspace{-0.6em}\triangleleft$ arrows, and *transition* premises, where $\longrightarrow\hspace{-0.6em}\triangleright$ arrows are used. Total weight premises specify the total weight of labelled transitions for each subterm required for the rule to trigger; transition premises, similarly to GSOS before, specify required behaviour for some of the subterms.

Total weight premises form a set, i.e., their order in a rule is irrelevant. The set of premises:

$$\left\{ \mathtt{x}_i \xrightarrow{a}\hspace{-0.6em}\triangleleft\, w_{a,i} \right\}_{a \in D_i, 1 \le i \le n}$$

for $D_i \subseteq A$, loosely specifies a function:

$$\theta : \{1, 2, \dots, n\} \times A \to W;$$

the latter satisfies the former if $\theta(i, a) = w_{a,i}$ for each $i = 1, \dots, n$ and $a \in D_i$. A complete specification, i.e., one where each $D_i = A$, uniquely determines θ that satisfies it. We say that a function θ *triggers* a rule R if it satisfies the total weight premises of R.

Intuitively, a total weight premise $\mathtt{x} \xrightarrow{a}\hspace{-0.6em}\triangleleft\, w$ is satisfied by a process x in a $\mathfrak{W}$-LTS, if the total weight of all a-labelled transitions at x equals w. This intuition will be made formal in Definition 15 below.

Transition premises in a rule form a sequence, i.e., their order is relevant. Note that the u_j in transition premises are not fixed weights (elements of W), but variables. The meaning of a premise $\mathtt{x} \xrightarrow{b,u}\hspace{-0.6em}\triangleright\, \mathtt{y}$ applied to a source process x and a target process y in a $\mathfrak{W}$-LTS is to assign the transition weight $\rho(x \xrightarrow{b} y)$ to the variable u, used then as an argument in the function β mentioned in the rule conclusion. This process is formally described in Definition 15 below.

Weight variables in transition premises are in fact redundant in a $\mathfrak{W}$-GSOS rule. Indeed, since each transition premise must come with a fresh weight variable, and the variables are then used only as arguments of β in the order prescribed by the order of premises, there is essentially only one way (up to renaming of weight variables) of putting them in any given rule. For brevity of notation, one can therefore omit weight variables altogether and write down $\mathfrak{W}$-GSOS rules as:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a}\!\!\triangleleft\, w_{a,i}\right\}_{a\in D_i, 1\le i\le n} \qquad \left\langle \mathtt{x}_{i_j} \xrightarrow{b_j}\!\!\triangleright\, \mathtt{y}_j \right\rangle_{1\le j\le k}}{\mathtt{f}(\mathtt{x}_1,\ldots,\mathtt{x}_n) \xrightarrow{c,\beta}\!\!\triangleright\, \mathtt{t}} \tag{12}$$

with the convention that the jth argument of β is the weight of the jth transition premise of the rule. The former full notation is useful as an intuitive reminder of where arguments of β come from; once one gets more familiar with the process of inducing $\mathfrak{W}$-LTSs from rules, the latter notation offers some welcome brevity.

We will be interested in collections of $\mathfrak{W}$-GSOS rules subject to a finiteness condition:

Definition 14 ($\mathfrak{W}$-GSOS specification)**.** Given a signature Σ and a set A of labels, a $\mathfrak{W}$-*GSOS specification* Λ is a set of $\mathfrak{W}$-GSOS rules such that for each $\mathtt{f} \in \Sigma$ and for each function $\theta : \{1,\ldots,n\} \times A \to W$, there are only finitely many rules with $\mathtt{f}$ in the source and triggered by θ.

To complete the definition of $\mathfrak{W}$-GSOS, we must show how $\mathfrak{W}$-GSOS specifications induce $\mathfrak{W}$-LTSs.

Definition 15 (Induced $\mathfrak{W}$-LTS)**.** The $\mathfrak{W}$-LTS induced by a $\mathfrak{W}$-GSOS specification Λ over a signature Σ and a set of labels A, has the set $T_\Sigma\emptyset$ of closed Σ-terms as states and A as the set of labels. The weight function $\rho : T_\Sigma\emptyset \times A \times T_\Sigma\emptyset \to W$ is defined by structural induction on the first argument. To this end, consider a process $s = \mathtt{f}(s_1,\ldots,s_n) \in T_\Sigma\emptyset$ and assume that all $\rho(s_i \xrightarrow{a} t)$ have been determined for all $a \in A$ and $t \in T_\Sigma\emptyset$. For a fixed label $c \in C$ and a process $t \in T_\Sigma\emptyset$, define $\rho(s \xrightarrow{c} t)$ as follows.

First, define an auxiliary total weight function $\theta : \{1,2,\ldots,n\} \times A \to W$ by

$$\theta_s(i,a) = \rho(s_i, a, T_\Sigma\emptyset)$$

that is, $\theta_s(i,a)$ is the total weight of a at s_i.

We shall say that a rule R as in (11) *fits* $s \xrightarrow{c} t$ if all of the following hold:

(i) the operator in the source of R is $\mathtt{f}$,

(ii) the label in the conclusion of R is c,

(iii) θ_s triggers R (i.e. total weight premises are satisfied by the s_i),

(iv) there exists a substitution $\sigma : \Xi_R \to T_\Sigma\emptyset$ such that:

- $\sigma\mathtt{x}_i = s_i$ for $i = 1,\ldots,n$, and
- $\sigma[\mathtt{t}] = t$.

It is important to note that if R fits $s \xrightarrow{c} t$, then the fitting substitution σ is unique. Indeed, the action of σ on the $\mathtt{x}_i$ is explicitly defined by $\sigma\mathtt{x}_i = s_i$, and the action on the

$\mathtt{y}_j$ is determined by the condition $\sigma[\mathtt{t}] = t$. This is easily proved by structural induction on $\mathtt{t}$, using the assumption that all variables $\mathtt{y}_j$ are present in $\mathtt{t}$.

If a rule R fits $s \xrightarrow{c} t$, its *contribution* to the weight of $s \xrightarrow{c} t$ is a value in W calculated by:

$$\gamma(R) = \beta\langle\rho(s_{i_j} \xrightarrow{b_j} \sigma\mathtt{y}_j)\rangle_{j=1..k}.$$

We then define $\rho(s \xrightarrow{c} t)$ as the sum, taken in $\mathfrak{W}$, of contributions of all rules in Λ that fit $s \xrightarrow{c} t$. The sum exists thanks to the finiteness condition in Definition 14.

Remark 16. It is easy to identify some classes of rules that can safely be removed from any specification without affecting the induced $\mathfrak{W}$-LTS. For an easy example, consider a rule R where the function β is constantly 0. By Definition 15, the contribution of R to any transition is also 0, therefore its presence in a $\mathfrak{W}$-GSOS specification is irrelevant as far as the induced $\mathfrak{W}$-LTS is concerned.

One might also safely require that the total weight premises in every rule R completely specify a triggering function θ, i.e., that $D_i = A$ for each $i = 1, \ldots, n$. Indeed, any rule R as in Definition 13 can be equivalently replaced by a (typically infinite) set of rules with the same transition premises and the same conclusion, one for each function θ that triggers R, with θ completely specified in the total weight premises of the corresponding rule. It is easy to see that the finiteness condition of Definition 14 is preserved by this step, and the $\mathfrak{W}$-LTS induced according to Definition 15 is not changed.

In practice, of course, one does not want to fully specify all triggering functions, especially for transition labels that do not appear elsewhere in the rule, therefore we did not require $A_i = D$ in Definition 13. Sometimes, however, it is practical to require the user to specify total weights for at least those labels that appear in transition premises.

To express these considerations formally, one may extend Definition 13 with the following additional syntactic requirements:

- β is not constantly 0,
- $b_j \in D_{i_j}$ for each $j = 1, \ldots, k$,

without affecting the expressive power of $\mathfrak{W}$-GSOS. We shall make use of these additional requirements in §6.

Theorem 17. Every $\mathfrak{W}$-GSOS specification Λ gives rise to a natural transformation λ as in (10). Moreover, the coalgebra induced from λ according to (9), coincides with the $\mathfrak{W}$-LTS induced from Λ according to Definition 15.

Proof. First we shall see how a single $\mathfrak{W}$-GSOS rule defines a natural transformation λ as in (10). For a $\mathfrak{W}$-GSOS rule R, and for any set X, consider an arbitrary $s = \mathtt{f}(x_1, \delta_1, \ldots, x_n, \delta_n) \in \Sigma(X \times (\mathcal{F}_{\mathfrak{W}}X)^A)$. To define $\lambda_X(s) \in (\mathcal{F}_{\mathfrak{W}}T_\Sigma X)^A$, pick an arbitrary $c \in A$ and $t \in T_\Sigma X$ and define $\lambda_X(s)(c)(t) \in W$ as follows.

Say that R fits s, c, t if:

(i) the operator in the source of R is $\mathtt{f}$,

(ii) the label in the conclusion of R is c,

(iii) for each total weight premise $\mathtt{x}_i \xrightarrow{a}\!\!\triangleleft\, w$ in R, there is $\sum_{y \in X} \delta_i(a)(y) = w$,

(iv) there exists a substitution $\sigma : \Xi_R \to X$ such that:

- $\sigma \mathtt{x}_i = x_i$ for $i = 1, \ldots, n$, and
- $\sigma[\mathtt{t}] = t$.

Then define:

$$\lambda_X(s)(c)(t) = \begin{cases} \beta \langle \delta_{i_j}(b_j)(\sigma \mathtt{y}_j) \rangle_{j=1..k} & \text{if } R \text{ fits } s, c, t \text{ with } \sigma, \\ 0 & \text{otherwise.} \end{cases}$$

We shall now prove that λ is natural in X. To this end, for any function $g : X \to Z$, any $s = \mathtt{f}(x_1, \delta_1, \ldots, x_n, \delta_n) \in \Sigma(X \times (\mathcal{F}_{\mathfrak{W}} X)^A))$, $c \in A$ and $t \in T_\Sigma Z$, one must check that:

$$(\mathcal{F}_{\mathfrak{W}} T_\Sigma g)^A (\lambda_X(s))(c)(t) = \lambda_Y(\Sigma(g \times (\mathcal{F}_{\mathfrak{W}} g)^A)(s))(c)(t).$$

The left side of this equation is:

$$(*) = \sum_{\substack{r \in T_\Sigma X \\ \text{s.t.} g[r] = t}} \lambda_X(s)(c)(r)$$

and the right side:

$$(**) = \begin{cases} \beta \langle (\mathcal{F}_{\mathfrak{W}} g)^A (\delta_{i_j})(b_j)(\theta \mathtt{y}_j) \rangle_{j=1..k} & \text{if } R \text{ fits } g[s], c, t \text{ with } \theta : \Xi_R \to Z, \\ 0 & \text{otherwise.} \end{cases}$$

The expression in the first clause of (**) can be further rewritten as:

$$\begin{aligned} \beta \left\langle (\mathcal{F}_{\mathfrak{W}} g)^A (\delta_{i_j})(b_j)(\theta \mathtt{y}_j) \right\rangle_{j=1..k} &= \beta \left\langle \sum_{y \in \overleftarrow{f}(\theta \mathtt{y}_j)} \delta_{i_j}(b_j)(y) \right\rangle_{j=1..k} = \\ &= \sum_{\substack{y_1, \ldots, y_k \in X \\ \text{s.t.} g y_j = \theta \mathtt{y}_j}} \beta \langle \delta_{i_j}(b_j)(y_j) \rangle_{j=1..k}; \end{aligned}$$

the second equality makes use of the multiadditivity of β.

Note now that if any of the conditions (i)-(iii) above fails, then R does not fit $g[s], c, t$, and it does not fit s, c, r for any r such that $g[r] = t$. If this is the case, both (*) and (**) equal 0 and the naturality equation holds, therefore it can be safely assumed that conditions (i)-(iii) hold. With this assumption, (**) can be rewritten as:

$$(**) = \begin{cases} \sum_{\substack{y_1, \ldots, y_k \in X \\ \text{s.t.} g y_j = \theta \mathtt{y}_j}} \beta \langle \delta_{i_j}(b_j)(y_j) \rangle_{j=1..k} & \text{if } \exists \theta : \Xi_R \to Z.\ \theta \mathtt{x}_i = g x_i, \theta[\mathtt{t}] = t \\ 0 & \text{otherwise.} \end{cases}$$

Recall that θ above, if it exists, is unique. Now if θ exists, then tuples $y_1, \ldots, y_k \in X$ such that $g y_j = \theta \mathtt{y}_j$ are in bijective correspondence with substitutions $\sigma : \Xi_R \to X$

such that $\sigma\mathtt{x}_i = x_i$ and $g[\sigma[\mathtt{t}]] = t$. Moreover, the existence of such σ implies that an appropriate θ exists (take $\theta = g \circ \sigma$). As a result, we can rewrite:

$$(**) = \sum_{\substack{\sigma:\Xi_R \to X \\ \text{s.t.} \sigma\mathtt{x}_i = x_i, \\ g[\sigma[\mathtt{t}]]=t}} \beta\langle \delta_{i_j}(b_j)(\sigma\mathtt{y}_j)\rangle_{j=1..k}$$

Obviously, a substitution σ as above yields a term $r \in T_\Sigma X$ such that $g[r] = t$ (take $r = \sigma[\mathtt{t}]$). Moreover, for every $r \in T_\Sigma X$ such that R fits s, c, r with a substitution σ, the substitution satisfies the condition in the sum above. As a result, now dropping the assumption that conditions (i)-(iii) hold, we may rewrite:

$$\begin{aligned}
(**) &= \sum_{\substack{r \in T_\Sigma X \\ \text{s.t.} g[r]=t, \\ R \text{ fits } s,c,r}} \beta\langle \delta_{i_j}(b_j)(\sigma\mathtt{y}_j)\rangle_{j=1..k} = \\
&= \sum_{\substack{r \in T_\Sigma X \\ \text{s.t.} g[r]=t}} \begin{cases} \beta\langle \delta_{i_j}(b_j)(\sigma\mathtt{y}_j)\rangle_{j=1..k} & \text{if } R \text{ fits } s,c,r \text{ with } \sigma \\ 0 & \text{otherwise} \end{cases} = \\
&= \sum_{\substack{r \in T_\Sigma X \\ \text{s.t.} g[r]=t}} \lambda_X(s)(c)(r) = (*).
\end{aligned}$$

This shows that a single $\mathfrak{W}$-GSOS rule defines an appropriate natural transformation. For an arbitrary $\mathfrak{W}$-GSOS specification λ, define

$$\lambda_X(s)(c)(t) = \sum_{R \in \Lambda} \lambda_X^R(s)(c)(t)$$

where λ^R arises from every single rule R as described above. Thanks to the finiteness condition in Definition 14, for each s and c the sum contains only finitely many non-zero summands, therefore the sum is well-defined and $\lambda_X(s)(c)$ is finitely supported. Naturality of λ follows easily.

It remains to be seen that the $\mathfrak{W}$-LTS induced from a $\mathfrak{W}$-GSOS specification coincides with the $(\mathcal{F}_{\mathfrak{W}}-)^A$-coalgebra arising from the corresponding λ according to (9). To this end, it is enough to show that the induced LTS, seen as a coalgebra, makes (9) commute. A straightforward way to prove this is to notice that the induction step in Definition 15 corresponds exactly to the definition of λ, along the correspondence between $\mathfrak{W}$-LTSs and $(\mathcal{F}_{\mathfrak{W}})^A$-coalgebras. □

Corollary 18. For any $\mathfrak{W}$-GSOS specification Λ, $\mathfrak{W}$-bisimilarity is a congruence on the $\mathfrak{W}$-LTS induced by Λ.

Proof. Use Theorems 1 and 17 with Propositions 9 and 10. □

6 Instances

6.1 GSOS as y-GSOS

To relate $\mathfrak{W}$-GSOS to a more familiar format, we shall now see what $\mathfrak{W}$-GSOS specifications look like for $\mathfrak{W} = y$ (see Example 1).

First, there are only two kinds of total weight premises to consider: one of the form $\mathtt{x} \overset{a}{\relbar\!\!\lhd} \mathtt{tt}$ that requires some a-transitions from a process corresponding to $\mathtt{x}$ to exist, and one of the form $\mathtt{x} \overset{a}{\relbar\!\!\lhd} \mathtt{ff}$ that, following Definition 15, forbids such transitions. One can rewrite the former as $\mathtt{x} \overset{a}{\relbar\!\!\rhd}$, and the latter as $\mathtt{x} \overset{a}{\not\relbar\!\!\rhd}$.

Next, it is easy to see that for any $k \in \mathbb{N}$, there are only two multiadditive functions $\beta : \{\mathtt{tt}, \mathtt{ff}\}^k \to \{\mathtt{tt}, \mathtt{ff}\}$ on the monoid $\mathcal{Y}$. Indeed, by multiadditivity axioms, β is fully determined by its value on the all-$\mathtt{tt}$ vector. Assigning $\beta(\mathtt{tt}, \ldots, \mathtt{tt}) = \mathtt{tt}$ or $\beta(\mathtt{tt}, \ldots, \mathtt{tt}) = \mathtt{ff}$ one obtains respectively the k-ary conjunction or the constantly $\mathtt{ff}$ function as β. It is easy to check that both are multiadditive. By Remark 16, one can safely restrict attention to rules with logical conjunction as β; this means that β can be left implicit in the description of each rule. Moreover, since conjunction is commutative, the order of transition premises in $\mathcal{Y}$-GSOS rules is irrelevant.

The above observations let one write $\mathcal{Y}$-GSOS rules in the form:

$$\frac{\left\{\mathtt{x}_i \overset{a}{\relbar\!\!\rhd}\right\}_{a \in E_i, 1 \le i \le n} \quad \left\{\mathtt{x}_i \overset{a}{\not\relbar\!\!\rhd}\right\}_{a \in B_i, 1 \le i \le n} \quad \left\{\mathtt{x}_i \overset{b_{ij}}{\relbar\!\!\rhd} \mathtt{y}_{ij}\right\}_{1 \le i \le n, 1 \le j \le k_i}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \overset{c}{\relbar\!\!\rhd} \mathtt{t}} \tag{13}$$

where

- $\mathtt{f} \in \Sigma$ and $ar(\mathtt{f}) = n, \quad$ with $n, k_i \in \mathbb{N}$;
- $\mathtt{x}_i$ and $\mathtt{y}_{ij}$ are all distinct variables and no other variables appear in $\mathtt{t} \in T_\Sigma \Xi$; moreover, all variables $\mathtt{y}_{ij}$ appear in $\mathtt{t}$;
- $E_i, B_i \subseteq A$ and $b_{i_j}, c \in A$.

The induction process described in Definition 15 specialises to the following procedure. For a process $s = \mathtt{f}(s_1, \ldots, s_n) \in T_\Sigma$, assume that all outgoing transitions from the s_i have been determined. For a fixed label $c \in C$ and a process $t \in T_\Sigma \emptyset$, determine whether the transition $s \xrightarrow{c} t$ is present, as follows.

A rule R as in (13) *fits* $s \xrightarrow{c} t$ if all of the following hold:

- the operator in the source of R is $\mathtt{f}$,
- the label in the conclusion of R is c,
- for each premise $\mathtt{x}_i \overset{a}{\relbar\!\!\rhd}$ in R, there is $s_i \xrightarrow{a} u$ for some u, and for each premise $\mathtt{x}_i \overset{a}{\not\relbar\!\!\rhd}$ there is no transition $s_i \xrightarrow{a} u$ for any process u,
- there exists a substitution $\sigma : \Xi_R \to T_\Sigma \emptyset$ such that:
  - $\sigma \mathtt{x}_i = s_i$ for $i = 1, \ldots, n$, and
  - $\sigma[\mathtt{t}] = t$.

If a rule R fits $s \xrightarrow{c} t$, it contributes the transition $s \xrightarrow{c} t$ to the induced system if and only if for each premise $\mathtt{x}_i \overset{b_{ij}}{\relbar\!\!\rhd} \mathtt{y}_{ij}$ in R, the transition $s_i \xrightarrow{b_{ij}} \sigma \mathtt{y}_{ij}$ is present. (The universal quantification in the previous sentence corresponds to the use of conjunction as β.) Then the transition $s \xrightarrow{c} t$ is present in the induced system if any rule contributes it. (The existential quantification here corresponds to disjunction being the operator in the underlying monoid.)

It is clear that both the format (13) and the associated induction procedure are almost exactly those GSOS format (2); the only difference in rule presentation is than in GSOS, premises $\mathtt{x} \xrightarrow{a}\!\triangleright$ are equipped with dummy target variables, and consequently the condition that all target variables of transition premises are present in $\mathtt{t}$, is dropped. It is not difficult to see that this makes no semantic difference in this case.

6.2 Stochastic GSOS

In the previous section, the simplicity of the weight monoid $\mathfrak{W} = y$ allowed us to greatly simplify the presentation of $\mathfrak{W}$-GSOS rules, without reducing their expressive power. For $\mathfrak{W} = \mathbb{R}_0^+$ this is hard to achieve, but a considerable simplification is possible at the price of a very slight (and irrelevant any example we could conceive of) reduction in expressive power.

Consider a multiadditive function $\beta : (\mathbb{R}_0^+)^k \to \mathbb{R}_0^+$. Using multiadditivity axioms, it is easy to prove by induction that for $n_1, \ldots, n_k \in \mathbb{N}$ there is

$$\beta(n_1, \ldots, n_k) = \beta(1, \ldots, 1) \cdot \prod_{i=1}^{k} n_i.$$

Further, for any $q_1, \ldots, q_n \in \mathbb{Q}$,

$$\beta(q_1, \ldots, q_k) = \beta(1, \ldots, 1) \cdot \prod_{i=1}^{k} q_i.$$

As a result, any multiadditive function, restricted to rational numbers, must be of the form

$$\beta(x_1, \ldots, x_n) = w \cdot \prod_{i=1}^{k} x_i, \tag{14}$$

for some constant $w \in \mathbb{R}_0^+$, and may be represented simply by the number w. For a simplified presentation of $\mathbb{R}_0^+$-rules, we restrict attention to functions β that are of this form on the whole domain $\mathbb{R}_0^+$. It should be immediately apparent that this restriction seems inessential in practice: indeed, the existence proof of multiadditive functions that are not of the form (14) is highly *nonconstructive*, and it is hard to imagine a practical example of an $\mathbb{R}_0^+$-GSOS rule that would exploit non-continuous dependence on specific irrational transition rates. In fact, the restriction to functions as in (14) does not change the expressive power of $\mathbb{R}_0^+$-GSOS at all, as we shall show in Theorem 22.

With this restriction, $\mathbb{R}_0^+$-GSOS rules as defined in (11) and (12) can be simplified considerably. First, all β's given by (14) are commutative, so as in §6.1, transition premises in rules can be seen as sets rather than sequences. Moreover, a function β as in (14) is determined, and may be presented in a rule, simply by the constant $w \in \mathbb{R}_0^+$. We obtain rules of the form:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a}\!\triangleleft\; w_{a,i}\right\}_{a \in D_i, 1 \leq i \leq n} \qquad \left\{\mathtt{x}_{i_j} \xrightarrow{b_j}\!\triangleright\; \mathtt{y}_j\right\}_{1 \leq j \leq k}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \xrightarrow{c,w}\!\triangleright\; \mathtt{t}}, \tag{15}$$

subject to conditions similar to those in Definition 13.

By Remark 16, we may additionally require that

- $w > 0$,
- $b_j \in D_{i_j}$ for each $j = 1, \dots, k$.

Moreover, thanks to special properties of $\mathfrak{W} = \mathbb{R}_0^+$, further restrictions can be safely imposed. For example, consider a rule:

$$\frac{\mathtt{x} \overset{a}{\relbar\!\!\lhd} 0 \quad \mathtt{x} \overset{a}{\relbar\!\!\rhd} \mathtt{y}}{\mathtt{f}(\mathtt{x}) \overset{c,w}{\relbar\!\!\relbar\!\!\rhd} \mathtt{f}(\mathtt{y})}$$

From Definition 15 it is easy to see that the contribution of this rule to the weight of any transition is 0. Indeed, this follows from the fact that in a $\mathbb{R}_0^+$-LTS if $\rho(x, a, X) = 0$, then $\rho(x, a, y) = 0$ for any $y \in X$, and from multilinearity of β. As a result, we may safely remove such rules from any specification, thus introducing another requirement:

- $w_{b_j,i_j} > 0$ for each $j = 1, \dots, k$.

This kind of syntactic restrictions may seem inessential, but in fact they let us present $\mathbb{R}_0^+$-GSOS rules in a much more intuitive fashion. Typically, in practical examples, the weight w used in the conclusion of an $\mathbb{R}_0^+$-GSOS rule depends multiplicatively on weights used in total weight premises, i.e., it is of the form

$$w = \frac{r}{\prod_{j=1}^{k} w_{b_j,i_j}} \tag{16}$$

for some simple $r \in \mathbb{R}^+$. It therefore makes sense to mention r instead of w in a rule, and wire the dependence (16) into the LTS induction procedure. To avoid confusion between the two similar representations of $\mathbb{R}_0^+$-GSOS rules, in the latter case we shall write $c@r$ instead of c, w in the conclusion label.

Under the condition that all $w_{b_j,i_j} > 0$ all this is merely syntactic sugar, but it allows for more intuitive presentation of inference rules, as we shall see in §7.

It is time to gather all the above requirements and simplifications in a single definition:

Definition 19. An *SGSOS rule* for a signature Σ and a set A of labels is an expression of the form:

$$\frac{\left\{\mathtt{x}_i \overset{a}{\relbar\!\!\lhd} w_{a,i}\right\}_{a \in D_i, 1 \le i \le n} \quad \left\{\mathtt{x}_{i_j} \overset{b_j}{\relbar\!\!\rhd} \mathtt{y}_j\right\}_{1 \le j \le k}}{\mathtt{f}(\mathtt{x}_1, \dots, \mathtt{x}_n) \overset{c@r}{\relbar\!\!\relbar\!\!\rhd} \mathtt{t}} \tag{17}$$

where

- $\mathtt{f} \in \Sigma$ and $ar(\mathtt{f}) = n$, with $n, k \in \mathbb{N}$, and $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$;
- $\mathtt{x}_i$ and $\mathtt{y}_j$ are all distinct variables and no other variables appear in $\mathtt{t} \in T_\Sigma \Xi$; moreover, all variables $\mathtt{y}_j$ appear in $\mathtt{t}$;
- $D_i \subseteq A$, $c \in A$ and $b_j \in D_{i_j}$

- $r \in \mathbb{R}^+, \quad w_{a,i} \in \mathbb{R}_0^+, \quad$ and moreover $w_{b_j,i_j} > 0$, for $j = 1, \ldots, k$.

A rule as the above is *triggered* by a function $\theta : \{1, 2, \ldots, n\} \times A \to \mathbb{R}_0^+$ if $\theta(i, a) = w_{a,i}$ for all $1 \le i \le n$ and all $a \in D_i$. A collection of rules is called an *SGSOS specification* if for every $\mathtt{f} \in \Sigma$, $c \in A$ and every function θ, only finitely many rules with $\mathtt{f}$ and c in the conclusion are triggered by θ.

Convention: If in a rule as in (17), for some $\mathtt{x}_i$ and $a \in D_i$ there is exactly one j for which $i_j = i$ and $b_j = a$, instead of the two premises $\mathtt{x}_i \overset{a}{\longrightarrow\!\!\lhd} w$ and $\mathtt{x}_i \overset{a}{\longrightarrow\!\!\rhd} \mathtt{y}_j$ we shall write $\mathtt{x}_i \overset{a \lhd w}{\longrightarrow\!\!\rhd} \mathtt{y}_j$ (meaning: '$\mathtt{x}_i$ makes an a-labelled transition to $\mathtt{y}_i$, and the total weight of a-labelled transitions from $\mathtt{x}_i$ is w,' and *not* '$\mathtt{x}_i$ makes an a-labelled transition to $\mathtt{y}_i$, with transition weight w').

The induction procedure of a stochastic transition system from an SGSOS specification is similar to that given in Definition 15. The notion of a fitting rule is exactly the same, but the contribution of a rule is defined a bit differently:

$$\gamma(R) = r \cdot \prod_{j=1}^{k} \frac{\rho(s_{i_j} \xrightarrow{b_j} \sigma\mathtt{y}_j)}{w_{b_j,i_j}}. \tag{18}$$

To formally relate (17) with (11), any SGSOS rule can be encoded as a $\mathbb{R}_0^+$-GSOS rule:

$$\frac{\left\{\mathtt{x}_i \overset{a}{\longrightarrow\!\!\lhd} w_{a,i}\right\}_{a \in D_i, 1 \le i \le n} \qquad \left\langle \mathtt{x}_{i_j} \overset{b_j}{\longrightarrow\!\!\rhd} \mathtt{y}_j \right\rangle_{1 \le j \le k}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \overset{c,\beta}{\longrightarrow\!\!\rhd} \mathtt{t}} \tag{19}$$

with transition premises arranged in an arbitrary order, and with $\beta : (\mathbb{R}_0^+)^k \to \mathbb{R}_0^+$ defined by:

$$\beta(w_1, \ldots, w_k) = \frac{r}{\prod_{j=1}^{k} w_{b_j,i_j}} \cdot \prod_{j=1}^{k} w_j.$$

The numbers w_{b_j,i_j} are fixed, and the division is well-defined, by the last requirement on SGSOS rules above. Since β is commutative, the order of transition premises can be chosen arbitrarily.

Some illustrative examples of SGSOS specifications will be given in §7. For now, let us restate the congruence result:

Theorem 20. Every SGSOS specification Λ gives rise to a natural transformation

$$\lambda : \Sigma(\mathrm{Id} \times (\mathcal{F}_{\mathbb{R}_0^+})^A) \Longrightarrow (\mathcal{F}_{\mathbb{R}_0^+} T_\Sigma -)^A \tag{20}$$

Moreover, the coalgebra induced from λ according to (9), coincides with the RTS induced from Λ according to Definition 15 with formula (18).

Proof. This is simply an instance of Theorem 17 for Example 12. □

Corollary 21. For any SGSOS specification Λ, stochastic bisimilarity is a congruence on the RTS induced by Λ.

In the case of $\mathfrak{W} = \mathbb{R}^+_0$, Theorem 20 can be reversed, giving a completeness result of the SGSOS format with respect to the distributive law approach to stochastic system specification:

Theorem 22. Any distributive law (20) arises from an SGSOS specification according to Definition 15 with formula (18).

Proof. The proof closely follows an analogous result of Bartels [3] for probabilistic transition systems, and is reported in Appendix A. □

We do not know whether the completeness result holds for all monoids $\mathfrak{W}$.

6.3 $\mathbb{R}^{+\infty}$-GSOS

The appearance of $\mathfrak{W}$-GSOS rules does not depend on the monoid structure of $\mathfrak{W}$ apart from the choice of functions β, so $\mathbb{R}^{+\infty}$-GSOS and $\mathbb{R}^+_0$-GSOS specifications look almost the same. Note that for the monoid $\mathbb{R}^{+\infty}$ (see Example 7), where the minimum operation is taken as addition, a function β is multiadditive if and only if it is monotonic and preserves ∞, i.e. $\beta(w_1, \ldots, w_n) = \infty$ whenever some $w_i = \infty$. As a result, $\mathbb{R}^{+\infty}$-GSOS rules look as in (11), with the requirement that all $w_{a,i} \in \mathbb{R}^{+\infty}$ and β is a monotonic, ∞-preserving function.

This rule format allows for SOS definition of several interesting operators aimed at compositional specification of cost-oriented transition systems. For example, a unary prefixing operator and a binary nondeterministic choice operator, with syntax given by:

$$P ::= \mathtt{nil} \mid (a,w).P \mid P + P \qquad (a \in A, w \in \mathbb{R}^+)$$

can be defined by rules:

$$\frac{\quad}{(a,w).\mathtt{x} \xrightarrow{a,w} \mathtt{x}} \qquad \frac{\mathtt{x} \xrightarrow{a,u} \mathtt{x}'}{\mathtt{x} + \mathtt{y} \xrightarrow{a,u} \mathtt{x}'} \qquad \frac{\mathtt{y} \xrightarrow{a,u} \mathtt{y}'}{\mathtt{x} + \mathtt{y} \xrightarrow{a,u} \mathtt{y}'}$$

where in the first rule w represents the function constant at w, and in the other two rules the identity function is taken for β. By Definition 15 applied to $\mathfrak{W} = \mathbb{R}^{+\infty}$, contributions of different rules to single transitions are combined using the minimum operation. As a result, for example, the process $(a.2).\mathtt{nil} + (a,3).\mathtt{nil}$ is $\mathbb{R}^{+\infty}$-bisimilar to $(a.2).\mathtt{nil}$, which corresponds to the intuition that nondeterministic processes always choose the lowest possible cost of transition.

Other versions of nondeterministic composition, where the process of resolving a nondeterministic choice is associated with its own internal cost, can also be modeled. For example, a binary operator ${}_3{+}_5$ is defined by rules:

$$\frac{\mathtt{x} \xrightarrow{a,u} \mathtt{x}'}{\mathtt{x} \,{}_3{+}_5\, \mathtt{y} \xrightarrow{a,u+3} \mathtt{x}'} \qquad \frac{\mathtt{y} \xrightarrow{a,u} \mathtt{y}'}{\mathtt{x} \,{}_3{+}_5\, \mathtt{y} \xrightarrow{a,u+5} \mathtt{y}'}$$

Here, choosing the left summand of nondeterministic choice incurs a lower cost.

Various ways of synchronisation are also possible. For example, one can add cost information to the well-known CCS communication rule for a binary parallel composition operator $\|$, as in:

$$\frac{\mathtt{x} \xrightarrow{a,u} \mathtt{x}' \quad \mathtt{y} \xrightarrow{a,v} \mathtt{y}'}{\mathtt{x}\|\mathtt{y} \xrightarrow{\tau,u+v} \mathtt{x}'\|\mathtt{y}'}$$

which can model a situation where two processes use a common resource during synchronisation, so that their costs of single transitions are added together. Another option is:

$$\frac{\mathtt{x} \xrightarrow{a,u} \mathtt{x}' \quad \mathtt{y} \xrightarrow{a,v} \mathtt{y}'}{\mathtt{x}\|\mathtt{y} \xrightarrow{\tau,\max(u,v)} \mathtt{x}'\|\mathtt{y}'}$$

where, intuitively, the two processes do not compete for a shared resource. Any operation can be used for weight combination here, as long as it is monotonic and preserves ∞.

Additional flexibility is provided by total weight premises of $\mathfrak{W}$-GSOS, which can be used here to check the minimal weight among all transition originating in a given process. For example, for a weighted version of a priority operator, one might define a unary operator ∂_{ab} by taking rules:

$$\frac{\mathtt{x} \overset{a}{\longrightarrow\!\!\!\lhd} w \quad \mathtt{x} \overset{b}{\longrightarrow\!\!\!\lhd} v \quad \mathtt{x} \xrightarrow{a,u} \mathtt{x}'}{\partial_{ab}(\mathtt{x}) \xrightarrow{a,u} \partial_{ab}(\mathtt{x}')} \qquad \frac{\mathtt{x} \overset{a}{\longrightarrow\!\!\!\lhd} v \quad \mathtt{x} \overset{b}{\longrightarrow\!\!\!\lhd} w \quad \mathtt{x} \xrightarrow{b,u} \mathtt{x}'}{\partial_{ab}(\mathtt{x}) \xrightarrow{b,u} \partial_{ab}(\mathtt{x}')}$$

for each $w \leq v \in \mathbb{R}^{+\infty}$. The resulting set of rules is uncountable, but it satisfies the finiteness condition of Definition 14. The operator defined by these rules preserves all a-labelled transitions if the minimal weight of an a-labelled outgoing transition is not bigger than the minimal weight of a b-labelled one, and vice versa.

All these operators conform to the $\mathbb{R}^{+\infty}$-GSOS format, so compositionality of $\mathbb{R}^{+\infty}$-weighted bisimilarity is immediately guaranteed for them.

7 Examples of SGSOS

We shall now concentrate on SGSOS, perhaps the most interesting instance of $\mathfrak{W}$-GSOS. To illustrate the form of SGSOS specifications, we 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 23 (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:

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

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

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

Thus, according to Definition 15 and the formula (18), the process

$$P = (a, 2).(b, 3).\mathtt{nil}$$

can make a unique transition $P \xrightarrow{a,2} (b, 3).\mathtt{nil}$ in the RTS induced by the rules.

Example 24 (stochastic choice)**.** Consider an extension of the language from Example 23 with a binary operator $P + P$ with semantics defined by rules:

$$\frac{\mathtt{x}_1 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{a@r} \mathtt{y}} \qquad \frac{\mathtt{x}_2 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 + \mathtt{x}_2 \xrightarrow{a@r} \mathtt{y}}$$

(note the use of the notational convention from Definition 19), 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 function $f : A \to \mathbb{R}_0^+$: those where $r = f(a)$. The rules define + to be the stochastic choice operator, as present e.g. in PEPA and stochastic π-calculus. In particular, according to Definition 15 with formula (18), in the stochastic transition system induced by the rules, the process $P = (a, 2).\mathtt{nil} + (a, 2).(b, 1).\mathtt{nil} + (c, 3).\mathtt{nil}$ can make three transitions $P \xrightarrow{a,2} \mathtt{nil}$, $P \xrightarrow{a,2} (b, 1).\mathtt{nil}$ and $P \xrightarrow{b,3} \mathtt{nil}$.Note, however, that the process $Q = (a, 2).\mathtt{nil} + (a, 3).\mathtt{nil}$ can only make one transition $Q \xrightarrow{a,5} \mathtt{nil}$, and this is because both the left rule with $f(a) = 2$ and the right rule with $f(a) = 3$ contribute to the weight of the transition. In particular, processes $(a, 2).\mathtt{nil} + (a, 3).\mathtt{nil}$ and $(a, 5).\mathtt{nil}$ are not only stochastic bisimilar, but can actually make exactly the same outgoing transitions.

We believe it is important to remark here 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 25 (PEPA-style synchronisation)**.** Extend the language from Example 24 with a binary synchronisation operator:

$$P ::= \ldots \mid P \underset{L}{\bowtie} P$$

for each $L \subseteq A$, with semantics defined by a family of rules:

$$\frac{\mathtt{x}_1 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{a@r} \mathtt{y} \underset{L}{\bowtie} \mathtt{x}_2} \qquad \frac{\mathtt{x}_2 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{a@r} \mathtt{x}_1 \underset{L}{\bowtie} \mathtt{y}} \tag{21}$$

$$\frac{\mathtt{x}_1 \xrightarrow{b \triangleleft r_1} \mathtt{y}_1 \quad \mathtt{x}_2 \xrightarrow{b \triangleleft r_2} \mathtt{y}_2}{\mathtt{x}_1 \underset{L}{\bowtie} \mathtt{x}_2 \xrightarrow{b@R} \mathtt{y}_1 \underset{L}{\bowtie} \mathtt{y}_2} \tag{22}$$

for each $a \in A \setminus L$, $\quad b \in L$, $\quad$ and $r, r_1, r_2, R \in \mathbb{R}^+$ such that $R = \min(r_1, r_2)$. It is not difficult to see that this, according to Definition 15 with formula (18), is the synchronisation operator of PEPA where the minimal rate law (5) is used. As an example, consider processes

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

where P_1, P_2 and Q_1 are all distinct. Then the process $P \underset{\{b\}}{\bowtie} Q$, where $b \neq a$, can make the following transitions:

$$P \underset{\{b\}}{\bowtie} Q \xrightarrow{a,1} P_1 \underset{\{b\}}{\bowtie} Q \qquad P \underset{\{b\}}{\bowtie} Q \xrightarrow{a,3} P_2 \underset{\{b\}}{\bowtie} Q \qquad P \underset{\{b\}}{\bowtie} Q \xrightarrow{a,2} P \underset{\{b\}}{\bowtie} Q_1.$$

On the other hand, the outgoing transitions from $P \underset{\{a\}}{\bowtie} Q$ are:

$$P \underset{\{a\}}{\bowtie} Q \xrightarrow{a,\frac{1}{2}} P_1 \underset{\{a\}}{\bowtie} Q_1 \qquad P \underset{\{a\}}{\bowtie} Q \xrightarrow{a,\frac{3}{2}} P_2 \underset{\{a\}}{\bowtie} Q_1.$$

Example 26 (CCS-style communication)**.** Similarly, one can extend the language from Example 24 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:

$$P ::= \ldots \mid P \parallel P$$

with semantics defined by rules:

$$\frac{\mathtt{x}_1 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 \parallel \mathtt{x}_2 \xrightarrow{a@r} \mathtt{y} \parallel \mathtt{x}_2} \qquad \frac{\mathtt{x}_2 \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{x}_1 \parallel \mathtt{x}_2 \xrightarrow{a@r} \mathtt{x}_1 \parallel \mathtt{y}} \tag{23}$$

$$\frac{\mathtt{x}_1 \xrightarrow{a \triangleleft r_1} \mathtt{y}_1 \quad \mathtt{x}_2 \xrightarrow{\bar{a} \triangleleft r_2} \mathtt{y}_2}{\mathtt{x}_1 \parallel \mathtt{x}_2 \xrightarrow{\tau@R} \mathtt{y}_1 \parallel \mathtt{y}_2} \tag{24}$$

for each $a \in A$ and for each $r, r_1, r_2, R \in \mathbb{R}^+$ such that $R = \min(r_1, r_2)$. This, according to Definition 19, is the communication operator of the original definition of stochastic π-calculus [25], with the minimal rate law (5) used. For example, consider processes:

$$P = (a, 1).P_1 + (a, 3).P_2 \qquad\qquad Q = (\bar{a}, 2).Q_1$$

where P_1, P_2 and Q_1 are all distinct. Then the process $P \parallel Q$ can make the following transitions:

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

Alternatively, one could use the same rules with $R = r_1 \cdot r_2$. This would correspond to the semantics of parallel composition in the biological stochastic π-calculus [27], with the mass action law (6) used. For example, the process $P \parallel Q$ above can then make the following transitions:

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

Example 27. Several non-standard, yet meaningful syntactic constructors of stochastic transition systems can be defined within the SGSOS format. For example, one can extend the languages described above with unary '*catalyst*' and '*inhibitor*' operators

$\mathtt{cat}_a$ and $\mathtt{inh}_a$ for each $a \in A$, which influence rates of process transitions; they can be seen as stochastic counterparts of the label hiding operator ν of CCS. Their semantics is defined by the rules:

$$\frac{\mathtt{x} \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{cat}_a(\mathtt{x}) \xrightarrow{a @ 2r} \mathtt{cat}_a(\mathtt{y})} \qquad \frac{\mathtt{x} \xrightarrow{a \triangleleft r} \mathtt{y}}{\mathtt{inh}_a(\mathtt{x}) \xrightarrow{a @ r/2} \mathtt{inh}_a(\mathtt{y})}$$

$$\frac{\mathtt{x} \xrightarrow{b \triangleleft r} \mathtt{y}}{\mathtt{cat}_a(\mathtt{x}) \xrightarrow{b @ r} \mathtt{cat}_a(\mathtt{y})} \qquad \frac{\mathtt{x} \xrightarrow{b \triangleleft r} \mathtt{y}}{\mathtt{inh}_a(\mathtt{x}) \xrightarrow{b @ r} \mathtt{inh}_a(\mathtt{y})}$$

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 $\mathtt{cat}_a((a, 2).\mathtt{nil}) \xrightarrow{a,4} \mathtt{nil}$. Since the above rules conform to the SGSOS format, it is immediate that operators $\mathtt{cat}_a$ and $\mathtt{inh}_a$ preserve stochastic bisimilarity.

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

$$\frac{\mathtt{x}_1 \xrightarrow{a \triangleleft r_1} \mathtt{y} \quad \mathtt{x}_2 \xrightarrow{a} \triangleleft\, r_2}{\mathtt{x}_1 \mathbin{!!} \mathtt{x}_2 \xrightarrow{a @ r_1} \mathtt{y} \mathbin{!!} \mathtt{x}_2} \qquad \frac{\mathtt{x}_1 \xrightarrow{a} \triangleleft\, r_2 \quad \mathtt{x}_2 \xrightarrow{a \triangleleft r_1} \mathtt{y}}{\mathtt{x}_1 \mathbin{!!} \mathtt{x}_2 \xrightarrow{a @ r_1} \mathtt{x}_1 \mathbin{!!} \mathtt{y}}$$

for each $a \in A$ and $r_1, r_2 \in \mathbb{R}_0^+$ such that $r_1 > r_2$. For example, the process $P = ((a, 2).Q) \mathbin{!!} ((a, 3).T)$ has only one outgoing transition $P \xrightarrow{a,3} ((a, 2).Q) \mathbin{!!} T$. Again, stochastic bisimilarity is immediately compositional with respect to !!.

This last example illustrates the fact that in the semantics of stochastic operators defined by SGSOS specifications, total rates of subprocesses can be tested, compared and used in an arbitrary fashion (recall that the total rate for a process P and a label a is the sum of rates of a-transitions from P). Note, however, that rates of *single* transitions of subprocesses cannot be used with the same degree of freedom. For example, within the SGSOS framework, it is not possible to write the semantics of a hypothetical unary operator $\mathtt{even}(\_)$ that would propagate only transitions with even rates, suppressing those with odd rates. Indeed, stochastic bisimilarity would not be preserved by this operator. On the other hand, it is possible to write an SGSOS operator that propagates only transitions with labels whose total rates are even.

8 Associative parallel composition for stochastic systems

As a small side application of the theory of $\mathfrak{W}$-GSOS and SGSOS, we address an issue in the original design of the stochastic π-calculus [25]: if the minimal rate law (5) is used in the definition (7), then the CCS-style communication operator $\|$ is not associative up to stochastic bisimilarity. Indeed, consider processes

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

Note that $r_a(P_1) = r$, $r_a(P_1 \parallel P_1) = 2r$, and $r_{\bar{a}}(P_2) = r_{\bar{a}}(P_1 \parallel P_2) = r$. With reference to the construction via proved transition of [25] recalled in §2.2.2, this means that in the derived proved-transitions semantics

$$Q_1 \xrightarrow{\langle \|_1(a,r),(\bar{a},r)\rangle,R_1} (\mathtt{nil} \parallel P_1) \parallel \mathtt{nil}$$
$$Q_2 \xrightarrow{\langle (a,r),\|_2(\bar{a},r)\rangle,R_2} \mathtt{nil} \parallel (P_1 \parallel \mathtt{nil}),$$

one has

$$R_1 = \min(2r, r) \cdot \frac{r}{2r} \cdot \frac{r}{r} = \frac{r}{2} \qquad\qquad R_2 = \min(r, r) \cdot \frac{r}{r} \cdot \frac{r}{r} = r.$$

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 [27], 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 operators within the framework of GSOS and characterise those rate formulas β 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.

8.1 CCS-style communication

Consider the language of Example 26, extending those of Examples 23 and 24. Two versions of the language were mentioned there, depending on the choice of the family of rules of type (24) used in the semantics: one where $R = \min(r_1, r_2)$ (the *minimal rate* law) and one where $R = r_1 \cdot r_2$ (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 (24) in our semantics for each label a, and that, moreover, the number R in the conclusion of the rules does not depend on a; we can thus treat the R's as a function $R : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$. Formally, we then look for a characterisation of those rate functions R for which the operator $\parallel$ is associative up to stochastic bisimilarity.

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

Theorem 28. In the situation described above, $\parallel$ is associative up to stochastic bisimilarity if and only if $R(r_1, r_2) = c \cdot r_1 \cdot r_2$ for some constant $c \in \mathbb{R}^+$.

Proof. First we prove that if $\parallel$ is associative up to bisimilarity then R is of the form above. Fix arbitrary numbers $t_1, t_2, t_3 \in \mathbb{R}^+$ and consider processes

$$P_1 = (a, t_1).\mathtt{nil} \qquad P_2 = (\bar{a}, t_2).\mathtt{nil} \qquad P_3 = (\bar{a}, t_3).\mathtt{nil}.$$

Applying Definition 15 with formula (18) to the rules in Example 26, it is straightforward to derive the following transitions in the induced transition system:

$$P_1 \xrightarrow{a,t_1} \mathtt{nil} \qquad P_2 \xrightarrow{\bar{a},t_2} \mathtt{nil} \qquad P_3 \xrightarrow{\bar{a},t_3} \mathtt{nil}$$

$$P_1 \parallel P_2 \xrightarrow{\tau,R(t_1,t_2)} \mathtt{nil} \parallel \mathtt{nil} \qquad P_2 \parallel P_3 \xrightarrow{\bar{a},t_2} \mathtt{nil} \parallel P_3$$

$$(P_1 \parallel P_2) \parallel P_3 \xrightarrow{\tau,R(t_1,t_2)} (\mathtt{nil} \parallel \mathtt{nil}) \parallel P_3$$

$$P_1 \parallel (P_2 \parallel P_3) \xrightarrow{\tau,R(t_1,t_2+t_3)\cdot t_2/(t_2+t_3)} \mathtt{nil} \parallel (\mathtt{nil} \parallel P_3).$$

Note that in the derivation of the above transition for $P_1 \parallel P_2$, an instance of (24) with $r_1 = t_1, r_2 = t_2$ is used; on the other hand, the correct instance for the above transition from $P_1 \parallel (P_2 \parallel P_3)$ is with $r_1 = t_1$ and $r_2 = t_2 + t_3$.

By our assumption on $\parallel$, processes $(P_1 \parallel P_2) \parallel P_3$ and $P_1 \parallel (P_2 \parallel P_3)$ are stochastic bisimilar. Moreover, it is easy to see that they cannot make any transitions to any process bisimilar to $(\mathtt{nil} \parallel \mathtt{nil}) \parallel P_3$ other than those derived above. This means that

$$R(t_1, t_2 + t_3) \cdot \frac{t_2}{t_2 + t_3} = R(t_1, t_2).$$

Since t_2 and t_3 were arbitrary positive real numbers, it follows that for any $t_1, t_2 \in \mathbb{R}^+$ and $p > 1$ there is

$$R(t_1, p \cdot t_2) = p \cdot R(t_1, t_2)$$

(take $t_3 = (p - 1) \cdot t_2$). In other words, the function R is linear in the second argument. A very similar argument shows that R is linear is the first argument as well; this means that $R(t_1, t_2) = c \cdot t_1 \cdot t_2$ for some constant c.

On the other hand, for $R(t_1, t_2) = c \cdot t_1 \cdot t_2$ the operator $\parallel$ is associative up to stochastic bisimilarity. The proof of this proceeds much the same as the analogous proof for CCS [19]. □

8.2 CSP-style synchronisation

Consider now the language of Example 25, extending those of Examples 23 and 24. Again, assume that for each pair $r_1, r_2 \in \mathbb{R}_0^+$ there is exactly one rule of the type (22) for each label a, and that the number R in the conclusion of the rules does not depend on a; thus, as before, we have a function $R : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$. It turns out that, compared to §8.1, one has considerably more freedom in choosing R so that each of the synchronisation operators $\underset{L}{\bowtie}$ is associative:

Theorem 29. In the situation described above, each $\underset{L}{\bowtie}$ is associative up to stochastic bisimilarity if and only if R is associative, i.e., $R(r_1, R(r_2, r_3)) = R(R(r_1, r_2), r_3)$ for all $r_1, r_2, r_3 \in \mathbb{R}^+$.

Proof. Fix an arbitrary nonempty $L \subseteq A$. We prove that if $\underset{L}{\bowtie}$ is associative up to bisimilarity then R is associative. Fix some $a \in L$ and arbitrary numbers $t_1, t_2, t_3 \in \mathbb{R}^+$ and consider processes

$$P_i = (a, t_i).\mathtt{nil}$$

for $i = 1, 2, 3$. Applying Definition 15 with formula (18) to the rules in Example 25, it is straightforward to derive the following transitions in the induced transition system:

$$P_1 \xrightarrow{a,t_1} \mathtt{nil} \qquad P_2 \xrightarrow{a,t_2} \mathtt{nil} \qquad P_3 \xrightarrow{a,t_3} \mathtt{nil}$$

$$P_1 \underset{L}{\bowtie} P_2 \xrightarrow{a,R(t_1,t_2)} \mathtt{nil} \underset{L}{\bowtie} \mathtt{nil} \qquad P_2 \underset{L}{\bowtie} P_3 \xrightarrow{a,R(t_2,t_3)} \mathtt{nil} \underset{L}{\bowtie} \mathtt{nil}.$$

Note also that no other transitions from $P_1 \underset{L}{\bowtie} P_2$ or $P_2 \underset{L}{\bowtie} P_3$ can be derived, therefore

$$r_a(P_1 \underset{L}{\bowtie} P_2) = R(t_1, t_2) \text{ and } r_a(P_2 \underset{L}{\bowtie} P_3) = R(t_2, t_3).$$

This means that rates s_1, s_2 of the following transitions:

$$P_1 \underset{L}{\bowtie} (P_2 \underset{L}{\bowtie} P_3) \xrightarrow{a,s_1} \mathtt{nil} \underset{L}{\bowtie} (\mathtt{nil} \underset{L}{\bowtie} \mathtt{nil})$$
$$(P_1 \underset{L}{\bowtie} P_2) \underset{L}{\bowtie} P_3 \xrightarrow{a,s_1} (\mathtt{nil} \underset{L}{\bowtie} \mathtt{nil}) \underset{L}{\bowtie} \mathtt{nil}$$

are calculated as follows:

$$s_1 = R(t_1, r_a(P_2 \underset{L}{\bowtie} P_3)) \cdot \frac{t_1}{t_1} \cdot \frac{R(t_2, t_3)}{r_a(P_2 \underset{L}{\bowtie} P_3)} = R(t_1, R(t_2, t_3))$$
$$s_2 = R(r_a(P_1 \underset{L}{\bowtie} P_2), t_3) \cdot \frac{R(t_1, t_2)}{r_a(P_1 \underset{L}{\bowtie} P_2)} \cdot \frac{t_3}{t_3} = R(R(t_1, t_2), t_3)$$

By our assumption on $\|$, processes $(P_1 \underset{L}{\bowtie} P_2) \underset{L}{\bowtie} P_3$ and $P_1 \underset{L}{\bowtie} (P_2 \underset{L}{\bowtie} P_3)$ are stochastic bisimilar. Moreover, it is easy to see that they cannot make any transitions to any process bisimilar to $(\mathtt{nil} \underset{L}{\bowtie} \mathtt{nil}) \underset{L}{\bowtie} \mathtt{nil}$ other than those derived above. This means that

$$R(t_1, R(t_2, t_3)) = R(R(t_1, t_2), t_3)$$

and since t_i were arbitrary, it follows that R is associative.

On the other hand, for associative R the operator $\|$ is associative up to stochastic bisimilarity. The proof of this proceeds much the same as the analogous proof for CCS [19]. □

Note that the function $R(r_1, r_2) = c \cdot r_1 \cdot r_2$ of Theorem 28 (the mass action law) is associative. However, there is plenty of other associative functions such as $R(r_1, r_2) = \min(r_1, r_2)$ (the minimal rate law); this means that it is easier to obtain associative parallel composition within the framework of CSP-style synchronisation, as in PEPA, than with CCS-style communication, as in stochastic π-calculus.

9 Conclusion

We defined SGSOS, the first congruence format for the specification of stochastic transition systems which guarantees the compositionally of stochastic bisimilarity. SGSOS allows intuitive specifications based on simple transition labels, their (total) apparent rates, and the (multi-additive) combination of actual transition rates. Even though some of the inherit complexity of the matter is transferred to the construction of the induced stochastic labelled transition system, we believe that the advantage of a natural specification framework make of SGSOS a significant contribution. To illustrate the point, we presented several examples of the SGSOS specification of stochastic operators commonly encountered in the literature.

Our proof technique applies Turi and Plotkin's bialgebraic treatment of operational semantics, and Bartels' work on probabilistic systems. The theoretical development uses a general notion of weighted transition system and defines a parametric format system $\mathfrak{W}$-GSOS, on which our results are proved. $\mathfrak{W}$-GSOS can be instantiated to several interesting cases, SGSOS chief among them.

Frameworks like this also provide the opportunity to focus on meta-theoretical questions. We analysed one such issue by using SGSOS to characterise which parallel composition operators are compatible with (the observational semantics of) the stochastic pi-calculus. The interesting result is that stochastic pi as originally formulated is not as flexible a framework as it appears.

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] M. Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Science*, 114:299–315, 1993.

[3] F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. PhD dissertation, CWI, Amsterdam, 2004.

[4] M. Bernardo and R. Gorrieri. A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *Theoretical Computer Science*, 202(1-2):1–54, 1998.

[5] B. Bloom, S. Istrail, and A. Meyer. Bisimulation can't be traced. *Journal of the ACM*, 42:232–268, 1995.

[6] 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.

[7] M. Calder, S. Gilmore, and J. Hillston. Automatically deriving odes from process algebra models of signalling pathways. In *Procs. Computational Methods in Systems Biology*, pages 204–215, 2005.

[8] L. Cardelli and R. Mardare. The measurable space of stochastic processes. In *Procs. QEST'10*, pages 171–180. IEEE Computer Society, 2010.

[9] P. Degano and C. Priami. Enhanced operational semantics. *ACM Comput. Surv.*, 28(2):352–354, 1996.

[10] N. Götz, U. Herzog, and M. Rettelbach. 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.

[11] H. Hermanns, U. Herzog, and J.-P. Katoen. Process algebra for performance evaluation. *Theoretical Computer Science*, 274(1-2):43–87, 2002.

[12] J. Hillston. On the nature of synchronisation. In *Procs. PAPM'94*, pages 51–70, 1994.

[13] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[14] J. Hillston. Process algebras for quantitative analysis. In *Procs. LiCS'05*, pages 239–248. IEEE Computer Society Press, 2005.

[15] B. Klin. Structural operational semantics for weighted transition systems. In *Semantics and Algebraic Specification*, volume 5700 of *Lecture Notes in Computer Science*, pages 121–139. Springer, 2009.

[16] B. Klin and V. Sassone. Structural operational semantic for stochastic systems. In *Proc. FOSSACS'08*, volume 4962 of *LNCS*, pages 428–442, 2008.

[17] K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[18] S. Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

[19] R. Milner. A calculus of communicating systems. *Journal of the ACM*, 1980.

[20] F. Moller and C. Tofts. A temporal calculus of communicating systems. In *Proc. CONCUR'90*, volume 458 of *LNCS*, pages 401–415, 1990.

[21] Rocco De Nicola, Diego Latella, Michele Loreti, and Mieke Massink. Rate-based transition systems for stochastic process calculi. In *Procs. ICALP'09*, volume 5556 of *Lecture Notes in Computer Science*. Springer, 2009.

[22] D. M. Park. Concurrency and automata on infinite sequences. *LNCS*, 140:195–219, 1981.

[23] G. D. Plotkin. A structural approach to operational semantics. DAIMI Report FN-19, Computer Science Department, Aarhus University, 1981.

[24] G. D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.

[25] C. Priami. Stochastic π-calculus. *Computer Journal*, 38(7):578–589, 1995.

[26] C. Priami. Language-based performance prediction for distributed and mobile systems. *Information and Computation*, 175(2):119–145, 2002.

[27] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In *Proc. Pacific Symposium of Biocomputing*, 2001.

[28] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[29] D. Sangiorgi and D. Walker. *The π-Calculus: a Theory of Mobile Processes*. Cambridge University Press, 2003.

[30] D. Turi and G. D. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS'97*, pages 280–291. IEEE Computer Society Press, 1997.

A Proof of Theorem 22

To characterise natural transformations of the type

$$\lambda : \Sigma(\mathrm{Id} \times (\mathcal{F}_{\mathbb{R}_0^+})^A) \Longrightarrow (\mathcal{F}_{\mathbb{R}_0^+} T_\Sigma)^A \tag{25}$$

in terms of inference rules, we modify Bartels's analogous characterisation [3] for the case of probabilistic transition systems. On the way, we will use a few lemmas proved in [3].

First, note that λ as above is equivalent to a natural transformation:

$$\bar{\lambda} : \Sigma(\mathrm{Id} \times (\mathcal{F}_{\mathbb{R}_0^+})^A) \times A \Longrightarrow \mathcal{F}_{\mathbb{R}_0^+} T_\Sigma$$

(see [3, Lemma A.1.1]). Recall that Σ, a polynomial functor, is a coproduct. Also the functor $- \times A$ can be seen as an $|A|$-fold coproduct; this means that $\bar{\lambda}$ is equivalent to a family of natural transformations:

$$\rho_{\mathtt{f},c} : (\mathrm{Id} \times (\mathcal{F}_{\mathbb{R}_0^+})^A)^N \Longrightarrow \mathcal{F}_{\mathbb{R}_0^+} T_\Sigma \tag{26}$$

indexed by $\mathtt{f} \in \Sigma$ and $c \in A$, where $N = \{1, \ldots, ar(\mathtt{f})\}$. Each $\rho_{\mathtt{f},c}$ is equivalent to

$$\bar{\rho}_{\mathtt{f},c} : (\mathcal{F}_{\mathbb{R}_0^+})^{A \times N} \Longrightarrow \mathcal{F}_{\mathbb{R}_0^+} T_\Sigma(\mathrm{Id} + N) \tag{27}$$

(see [3, Lemma A.1.7]).

Note the natural isomorphism

$$(\mathcal{F}_{\mathbb{R}_0^+})^{A \times N} \cong (\mathcal{F}^+_{\mathbb{R}_0^+} + 1)^{A \times N} \cong \coprod_{E \subseteq A \times N} (\mathcal{F}^+_{\mathbb{R}_0^+})^E,$$

where $\mathcal{F}^+_{\mathbb{R}_0^+} X$ is the set $\mathcal{F}_{\mathbb{R}_0^+} X$ restricted to functions that are not totally zero. This means that each $\bar{\rho}_{\mathtt{f},c}$ is equivalent to a family of natural transformations:

$$\nu_{\mathtt{f},c,E} : (\mathcal{F}^+_{\mathbb{R}_0^+})^E \Longrightarrow \mathcal{F}_{\mathbb{R}_0^+} T_\Sigma(\mathrm{Id} + N) \tag{28}$$

indexed by $E \subseteq A \times N$. Further, by the natural isomorphism $\mathcal{F}^+_{\mathbb{R}_0^+} \cong \mathbb{R}^+ \times \mathcal{D}_\omega$, each $\nu_{\mathtt{f},c,E}$ is equivalent to a family of natural transformations:

$$\delta_{\mathtt{f},c,E,r} : (\mathcal{D}_\omega)^E \Longrightarrow \mathcal{F}_{\mathbb{R}_0^+} T_\Sigma(\mathrm{Id} + N) \tag{29}$$

indexed by $r : E \to \mathbb{R}^+$.

Note the natural isomorphism

$$T_\Sigma \cong \coprod_{t \in T_\Sigma 1} \mathrm{Id}^{|t|_*}$$

where $|t|_*$ is the number of occurrences of $* \in 1$ in t. Moreover, for functors $G_i : C \to \mathbf{Set}$ ($i \in I$), there is a natural isomorphism

$$\mathcal{F}_{\mathbb{R}_0^+}\left(\coprod_{i \in I} G_i\right) \cong \coprod_{J \subseteq_{\mathrm{fin}} I} \prod_{j \in J} \mathcal{F}^+_{\mathbb{R}_0^+} G_j;$$

indeed, a finitely supported function from a coproduct of a family of sets is equivalent to a finite subfamily (of those sets where the function is nonzero) together with a finitely supported, nonzero function from each set in the subfamily. Therefore, we can rewrite $\delta_{\mathtt{f},c,E,r}$ as:

$$\delta_{\mathtt{f},c,E,r} : (\mathcal{D}_\omega)^E \Longrightarrow \coprod_{J \subseteq_{\mathrm{fin}} T_\Sigma(1+N)} \prod_{t \in J} \mathcal{F}^+_{\mathbb{R}_0^+}(\mathrm{Id}^{|t|_*}). \tag{30}$$

Note that the functor $\mathcal{D}_\omega$, and therefore also $(\mathcal{D}_\omega)^E$, preserve final objects. This means, by [3, Lemma A.1.3], that $\delta_{\mathtt{f},c,E,r}$ factors through one of components of the coproduct in its codomain; in other words, it is equivalent to a finite set $J \subseteq T_\Sigma(1 + N)$ together with a family of natural transformations

$$\zeta_{\mathtt{f},c,E,r,t} : (\mathcal{D}_\omega)^E \Longrightarrow \mathcal{F}^+_{\mathbb{R}_0^+}(\mathrm{Id}^{|t|_*}) \tag{31}$$

indexed by $t \in J$. Now we can again use the natural isomorphism $\mathcal{F}^+_{\mathbb{R}_0^+} \cong \mathbb{R}_0^+ \times \mathcal{D}_\omega$ to present $\zeta_{\mathtt{f},c,E,r,t}$ as a pair of natural transformations. One is of the type $(\mathcal{D}_\omega)^E \Longrightarrow \mathbb{R}^+$; it is straigthforward to see that is is equivalent to a number $W \in \mathbb{R}^+$, since a transformation (in **Set**) from a functor preserving finality to a constant functor must be constant. The second component is

$$\xi_{\mathtt{f},c,E,r,t} : (\mathcal{D}_\omega)^E \Longrightarrow \mathcal{D}_\omega(\mathrm{Id}^{|t|_*}). \tag{32}$$

This completes the process of decomposition of λ into natural transformations of simpler types. We will now construct an inference rule representation of λ, proceeding from the simplest transformations to more complex ones.

ξ-specifications. A complete characterisation of transformations of type (32) was given by Bartels in [3, Corollary A.3.10]. We recall it here with some syntactic sugar removed or modified. Fix an infinite, countable set Ξ of variables. Given a set E and a number $m = |t|_*$, a *ξ-rule* is an expression of the form

$$\frac{\left\{e_j \triangleright \mathsf{y}_j\right\}_{1 \leq j \leq k}}{w \triangleright \mathsf{y}_{o_1}, \ldots, \mathsf{y}_{o_m}} \tag{33}$$

for some $k \in \mathbb{N}$ and $w \in \mathbb{R}^+$ (w is called the *weight* of the rule), where $e_j \in E$, $1 \leq o_i \leq k$ and $\mathsf{y}_1, \ldots, \mathsf{y}_k$ are pairwise distinct variables; moreover, for each $j = 1, \ldots, k$

we require an $i \in \{1, \ldots, m\}$ to exist such that $o_i = j$. A *ξ-specification* is a finite set of ξ-rules whose weights sum up to 1 (note that this implies that a ξ-specification is nonempty). Any ξ-specification determines a transformation ξ as in (32) as follows: given a set X and probability distributions $\mu_1, \ldots, \mu_{|E|} : X \to [0, 1]$, for each $(x_1, \ldots, x_m) \in X^m$, define $\xi_X((\mu_e)_{e \in E})(x_1, \ldots, x_m)$ to be the sum of all contributions inferred from all rules. To determine the contribution of a rule as in (33), check whether the substitution $\sigma : \Xi \to X$ that maps each $\mathtt{y}_{o_i}$ to x_i is well-defined (i.e., whether $o_i = o_j$ implies $x_i = x_j$). If this is the case, define the contribution of the rule to be

$$w \cdot \prod_{j=1}^{k} \mu_{e_j}(\sigma(\mathtt{y}_j)); \tag{34}$$

otherwise let it be 0.

In [3] it is proved that with this definition $\xi_X((\mu_e)_{e \in E})$ is a probability distribution on X^m, that ξ is a natural transformation, and that every transformation as in (32) is defined by a ξ-specification.

ζ-specifications. Stated simply, a ζ-specification is a number $W \in \mathbb{R}^+$ and a ξ-specification; it should be clear how these define natural transformations as in (31). However, for our purposes it will be useful to rephrase ζ-specifications a little. Note that in a ξ-specification, the sum of all rule weights must be one. Therefore a set of rules with a different sum of weights may be seen as a number (the sum of weights) together with a ξ-specification, obtained from these rules by normalisation of weights. Hence we say that for given E and $m = |t|_*$, a *ζ-specification* is any finite, nonempty set of ξ-rules. Any such specification determines a transformation ζ as in (31) just as in the case of ξ-specifications, except that now $\zeta_X((\mu_e)_{e \in E})$ need not be a probability distribution, but is just a finitely supported function from X^m that is not totally zero.

δ-specifications. By (30), a specification of $\delta_{\mathtt{f},c,E,r}$ in (29) for given E and $N = \{1, \ldots, ar(\mathtt{f})\}$ is a finite set $J \subseteq T_\Sigma(1 + N)$ together with a ζ-specification for each $t \in J$. These can be written as a single collection of rules, provided that each ξ-rule rule is tagged with its corresponding term t. Syntactically, this can be done by replacing the vector $\mathtt{y}_{o_1}, \ldots, \mathtt{y}_{o_m}$ with the term $\mathtt{t} \in T_\Sigma(\Xi + N)$, obtained from t by replacing each occurrence of $*$ with the respective variable $\mathtt{y}_{o_i}$ (recall that $m = |t|_*$). Note that any term $\mathtt{t}$ that contains all variables $\mathtt{y}_j$ and no other variables from Ξ can be obtained this way. Thus a *δ-specification* is a finite collection of *δ-rules* of the form:

$$\frac{\left\{e_j \triangleright \mathtt{y}_j\right\}_{1 \leq j \leq k}}{w \triangleright \mathtt{t}} \tag{35}$$

for some $k \in \mathbb{N}$ and $w \in \mathbb{R}^+$, where $e_j \in E$, $\mathtt{y}_1, \ldots, \mathtt{y}_k$ are pairwise distinct variables, and $\mathtt{t} \in T_\Sigma(\Xi + N)$ contains all variables $\mathtt{y}_j$ and no other variables from Ξ. Note that such a collection of rules uniquely determines the set J on which it is based (this is because every ζ-specification is nonempty), therefore J may be omitted from the specification. Note also that a δ-specification may be empty; this corresponds to $J = \emptyset$.

Any δ-specification determines a transformation δ as in (29) as follows: given a set X and probability distributions $\mu_1, \dots, \mu_{|E|} : X \to [0,1]$, for each $t \in T_\Sigma(X + N)$, define $\delta_X((\mu_e)_{e \in E})(t)$ to be the sum of all contributions inferred from all rules. To determine the contribution of a rule as in (35), check whether the substitution $\sigma : \Xi \to X$ such that $\mathtt{t}\sigma = t$ exists. If this is the case, define the contribution of the rule as in (34), otherwise let it be 0. Note that the $\sigma(\mathtt{y}_j)$ in (34) are independent from the choice of σ, as long as $\mathtt{t}\sigma = t$.

ν-specifications. A specification of $\nu_{\mathtt{f},c,E}$ as in (28) for given E and $N = \{1, \dots, ar(\mathtt{f})\}$ is a family of δ-specifications indexed by $(\mathbb{R}^+)^E$. Again, this can be written as a single family of rules tagged with appropriate functions $r : E \to \mathbb{R}^+$. Syntactically, let a *ν-rule* be an expression of the form:

$$\frac{\{e \triangleleft r_e\}_{e \in E} \qquad \left\{e_j \triangleright \mathtt{y}_j\right\}_{1 \le j \le k}}{w \triangleright \mathtt{t}} \tag{36}$$

for some $k \in \mathbb{N}$ and $w \in \mathbb{R}^+$, where $r_e \in \mathbb{R}^+$, $e_j \in E$, $\mathtt{y}_1, \dots, \mathtt{y}_k$ are pairwise distinct variables, and $\mathtt{t} \in T_\Sigma(\Xi + N)$ contains all variables $\mathtt{y}_j$ and no other variables from Ξ. A *ν-specification* is a set of ν-rules such that for each $r = (r_e)_{e \in E}$, there are only finitely many rules with r represented in the premises.

Any ν-specification determines a transformation ν as in (28) as follows: given a set X and finitely supported, not totally zero functions $f_1, \dots, f_{|E|} : X \to \mathbb{R}_0^+$, for each $t \in T_\Sigma(X + N)$, define $\nu_X((f_e)_{e \in E})(t)$ to be the sum of all contributions inferred from all rules. To determine the contribution of a rule as in (36), check whether

- $\sum_{x \in X} f_e(x) = r_e$ for each $e \in E$ (the *rate condition*), and
- the substitution $\sigma : \Xi \to X$ such that $\mathtt{t}\sigma = t$ exists.

If this is the case, define the contribution of the rule to be

$$w \cdot \prod_{j=1}^{k} \frac{f_{e_j}(\sigma(\mathtt{y}_j))}{r_{e_j}}; \tag{37}$$

otherwise let it be 0. Note that the division above is well-defined since $r_{e_j} \in \mathbb{R}^+$.

$\bar{\rho}$-specifications. A specification of $\bar{\rho}_{\mathtt{f},c}$ as in (27) for given $N = \{1, \dots, ar(\mathtt{f})\}$ is a family of ν-specifications indexed by subsets E of $A \times N$, or equivalently by tuples of subsets $E_1, \dots, E_n \subseteq A$, where $n = ar(\mathtt{f})$. Note that each ν-rule determines the set E on which it is based; this means that no additional tagging is needed and one might consider rules of the form

$$\frac{\{i, a \triangleleft r_{ai}\}_{a \in E_i, 1 \le i \le n} \qquad \left\{i_j, b_j \triangleright \mathtt{y}_j\right\}_{1 \le j \le k}}{w \triangleright \mathtt{t}} \tag{38}$$

for some $E_i \subseteq A$, where $i_j \in N$, $b_j \in E_{i_j}$, with other restrictions as for ν-rules in (36). A collection of such rules defines $\bar{\rho}_{\mathtt{f},c}$ much the same as for ν-specifications, except that here the rate condition for functions $(f_{a,i} : X \to \mathbb{R}_0^+)_{a \in A, 1 \le i \le n}$ is replaced with:

$$\sum_{x \in X} f_{a,i}(x) = \begin{cases} r_{ai} & \text{if } a \in E_i \\ 0 & \text{otherwise.} \end{cases}$$

At this point we introduce some syntactic sugar to rules (38). First, we allow $r_{ai} = 0$, at the same time forcing *all* tuples $(a, i) \in A \times N$ to be present in the premises of every rule. Clearly, this does not change the expressive power of rules at all.

Second, we again remove the just-introduced condition of all tuples (i, a) being present. However, at the same time, we see a single rule as a shorthand for a whole family of rules: all rules that can be obtained from it by adding arbitrary premises for tuples (a, i) absent from the rule. This step does not change the expressive power of collections of rules, either.

After introducing these conventions, we rearrange the syntax of (38) a little to obtain the notion of a *$\bar{\rho}$-rule*, an expression of the form:

$$\frac{\left\{ i \xrightarrow{a}\!\!\triangleleft\; r_{ai} \right\}_{a \in E_i, 1 \le i \le n} \qquad \left\{ i_j \xrightarrow{b_j}\!\!\triangleright\; \mathtt{y}_j \right\}_{1 \le j \le k}}{w \triangleright \mathtt{t}} \tag{39}$$

for some $k \in \mathbb{N}$, $w \in \mathbb{R}^+$ and $E_1, \ldots, E_n \subseteq A$, where $n = ar(\mathtt{f})$, $i_j \in \{1, \ldots, n\}$, $b_j \in E_{i_j}$, $r_{ai} \in \mathbb{R}_0^+$, $\mathtt{y}_1, \ldots, \mathtt{y}_k$ are pairwise distinct variables, and $\mathtt{t} \in T_\Sigma(\Xi + N)$ contains all variables $\mathtt{y}_j$ and no other variables from Ξ. A tuple of functions $f_1, \ldots, f_n : A \to \mathbb{R}_0^+$ *triggers* such a rule if for each $i = 1, \ldots, n$ and for each $a \in E_i$, $f_i(a) = r_{ai}$ (and other values of f_i may be arbitrary). A *$\bar{\rho}$-specification* is a set of $\bar{\rho}$-rules such that each tuple $f_1, \ldots, f_n : A \to \mathbb{R}_0^+$ triggers only finitely many rules.

Any $\bar{\rho}$-specification determines a transformation $\bar{\rho}$ as in (27) as follows: given a set X and finitely supported functions $g_{a,i} : X \to \mathbb{R}_0^+$ for each $i \in N$ and $a \in A$, for each $t \in T_\Sigma(X + N)$ define $\bar{\rho}_X((g_{i,a})_{i \in N, a \in A})(t)$ to be the sum of all contributions inferred from all rules. To determine the contribution of a rule as in (39), check whether

- functions $f_1, \ldots, f_n : A \to \mathbb{R}_0^+$ trigger the rule, where $f_i(a) = \sum_{x \in X} g_{a,i}(x)$, and
- the substitution $\sigma : \Xi \to X$ such that $\mathtt{t}\sigma = t$ exists.

If this is the case, define the contribution of the rule to be

$$w \cdot \prod_{j=1}^{k} \frac{g_{b_j, i_j}(\sigma(\mathtt{y}_j))}{r_{b_j i_j}}; \tag{40}$$

otherwise let it be 0.

ρ-specifications. By [3, Lemma A.1.7] transformations ρ and $\bar{\rho}$ as in (26) and (27) are equivalent, hence ρ-specifications should be equivalent to $\bar{\rho}$-specifications. However, the type of (26) suggests a slightly different syntactic presentation: throughout any rule, instead of number $i \in N$, we shall use variables $\mathtt{x}_1, \ldots, \mathtt{x}_n \in \Xi$, distinct from any

$\mathtt{y}_j$. The list of variables must be then specified in the rule. As a result, let a *ρ-rule* be an expression of the form:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a}\!\!\triangleleft\; r_{ai}\right\}_{a\in E_i, 1\leq i\leq n} \qquad \left\{\mathtt{x}_{i_j} \xrightarrow{b_j}\!\!\triangleright\; \mathtt{y}_j\right\}_{1\leq j\leq k}}{(\mathtt{x}_1,\ldots,\mathtt{x}_n) \xrightarrow{w}\!\!\triangleright\; \mathtt{t}} \tag{41}$$

with $\mathtt{t} \in T_\Sigma \Xi$ such that all $\mathtt{y}_j$, some (possibly none or all) $\mathtt{x}_i$, and no other variables from Ξ occur in $\mathtt{t}$, with other restrictions, and with a definition of a ρ-specification, as for (39).

Any ρ-specification determines a transformation ρ as in (26) as follows: given a set X and a tuple $(x_i, g_i)_{1\leq i\leq n}$ where $x_i \in X$ and $g_i : A \to X \to \mathbb{R}_0^+$ such that all $g_i(a)$ are finitely supported, for each $t \in T_\Sigma X$ define $\rho_X((x_i, g_i)_{i\in N})(t)$ to be the sum of all contributions inferred from all rules. To determine the contribution of a rule as in (41), check whether

- functions $f_1, \ldots, f_n : A \to \mathbb{R}_0^+$ trigger the rule, where $f_i(a) = \sum_{x\in X} g_i(a)(x)$, and
- the substitution $\sigma : \Xi \to X$ such that $\sigma(\mathtt{x}_i) = x_i$ and $\mathtt{t}\sigma = t$ exists.

If this is the case, define the contribution of the rule to be

$$w \cdot \prod_{j=1}^{k} \frac{g_{i_j}(b_j)(\sigma(\mathtt{y}_j))}{r_{b_j i_j}}; \tag{42}$$

otherwise let it be 0.

SGSOS specifications. Finally, specifications of λ as in (10) can be presented as families of ρ-specifications inxedex by $\mathtt{f} \in \Sigma$ and $c \in A$. Tagging ρ-rules with $\mathtt{f}$ and c appropriately, one obtaines the notion of SGSOS rule and SGSOS specifications as in Definition 19, and it is straightforward to modify the above procedure for ρ, to understand how SGSOS specifications define λ's as in (10). Combining this procedure with the inductive definition (9), one obtains the procedure of inducing stochastic systems (coalgebras) from SGSOS specifications, described in Definition 15. □