Abstract

General frameworks have been recently proposed as unifying theories for processes combining non-determinism with quantitative aspects (such as probabilistic or stochastically timed executions), aiming to provide general results and tools. This paper provides two contributions in this respect. First, we present a general GSOS specification format and a corresponding notion of bisimulation for non-deterministic processes with quantitative aspects. These specifications define labelled transition systems according to the ULTraS model, an extension of the usual LTSs where the transition relation associates any source state and transition label with state reachability weight functions (like, e.g., probability distributions). This format, hence called Weight Function GSOS (WF-GSOS), covers many known systems and their bisimulations (e.g. PEPA, TIPP, PCSP) and GSOS formats (e.g. GSOS, Weighted GSOS, Segala-GSOS, among others).

The second contribution is a characterization of these systems as coalgebras of a class of functors, parametric on the weight structure. This result allows us to prove soundness and completeness of the WF-GSOS specification format, and that bisimilarities induced by these specifications are always congruences.

1 Introduction

Process calculi and labelled transition systems have proved very successful for modelling and analysing concurrent, non-deterministic systems. This success has led to many extensions dealing with quantitative aspects, by adding further informations to the transition relation like probability rates or stochastic rates; see [5, 6, 15, 16, 22, 30] among others. These calculi are very effective in modeling and analysing quantitative aspects, like performance analysis of computer networks, model checking of time-critical systems, simulation of biological systems, probabilistic analysis of security and safety properties, etc.

Each of these calculi is tailored to a specific quantitative aspect and for each of them we have to develop a quite complex theory almost from scratch. This is a daunting and error-prone task, as it embraces the definition of syntax, semantics, transition rules, various behavioural equivalences, logics, proof systems; the proof of important properties like congruence of behavioural equivalences; the development of algorithms and tools for simulations, model checking, etc. This situation would naturally benefit from general frameworks for LTS with quantitative aspects, i.e., mathematical meta-models offering general methodologies, results, and tools, which can be uniformly instantiated to a wide range of specific calculi and models. In recent years, some of these theories have been proposed; we mention Segala systems [31], Functional Transition Systems (FuTS) [24], weighted labelled transition systems (WLTSs) [13, 22, 32], and Uniform Labelled Transition Systems (ULTraS), introduced by Bernardo, De Nicola and Loreti specifically as “a uniform setting for modelling non-deterministic, probabilistic, stochastic or mixed processes and their behavioural equivalences” [5].

This work is partially supported by MIUR PRIN project 2010LHT4KM, CINA.
A common feature of most of these meta-models is that their labelled transition relations do not yield simple states (e.g., processes), but some mathematical object representing quantitative informations about “how” each state can be reached. In particular, transitions in ULTraS systems have the form \( P \xrightarrow{a} \rho \) where \( \rho \) is a state reachability weight function, i.e., a function assigning a weight to each possible state. By suitably choosing the set of weights, and how these functions can be combined, we can recover ordinary non-deterministic LTSs, probabilistic transition systems, stochastic transition systems, etc. As convincingly argued in [5], the use of weight functions in place of plain processes simplifies the combination of non-determinism with quantitative aspects, like in the case of EMPA or PEPA. Moreover, it paves the way for general definitions and results, an important example being the notion of \( M \)-bisimulation [5].

Albeit quite effective, these meta-models are at their dawn, with many results and techniques still to be developed. An important example of these missing notions is a specification format, like the well-known GSOS format for non-deterministic labelled transition systems. These formats are very useful in practice, because they can be used for ensuring important properties of the system; in particular, the bisimulations induced by systems in these formats is guaranteed to be a congruence (which is crucial for compositional reasoning). From a more foundational point of view, these frameworks would benefit from a categorical characterization in the theory of coalgebras and bialgebras: this would allow a cross-fertilizing exchange of definitions, notions and techniques with similar contexts and theories.

In this paper, we provide two main contributions in this respect. First, we present a GSOS-style format, called Weight Function GSOS (WF-GSOS), for the specifications of non-deterministic systems with quantitative aspects. The judgement derived by rules in this style is of the form \( P \xrightarrow{a} \psi \), where \( P \) is a process and \( \psi \) is a weight function term. These terms describe weight functions, by means of an interpretation. A specification given in this format defines a ULTraS (although we could work also in other frameworks, such as FuTS). By choosing the set of weights, the language of weight function terms and their interpretation, we can readily capture many quantitative notions (probabilistic, stochastic, etc.), and different kinds of non-deterministic interactions, covering models like PEPA, TIPP, PCSP, EMPA, among others. Moreover, the WF-GSOS format supports a general definition of (strong) bisimulation, which can be readily instantiated to the various specific systems.

The second contribution is more fundamental. We provide a general categorical presentation of these non-deterministic systems with quantitative aspects. Namely, we prove that ULTraS systems are in one-to-one correspondence with coalgebras of a precise class of functors, parametric on the underlying weight structure. Using this characterization we define the abstract notion of WF-GSOS distributive law (i.e. a natural transformation of a specific shape) for these functors. We show that each WF-GSOS specification yields such a distributive law (i.e., the format is sound); taking advantage of Turi-Plotkin’s bialgebraic framework, this implies that the bisimulation induced by a WF-GSOS is always a congruence, thus allowing for compositional reasoning in quantitative settings. Additionally, we extend the results we presented in [26] proving that the WF-GSOS format is also complete: every abstract WF-GSOS distributive law for ULTraSs can be described by means of some WF-GSOS specification.

Previous works on this line are Klin’s Weighted GSOS, a rule format for WLTS [22], and Bartel’s Segala-GSOS, a rule format for Segala systems [3, §5.3]. Both WLTSs and Segala systems are subsumed by ULTraSs (in fact, WLTSs correspond to “deterministic” ULTraSs, where a process is associated to exactly one weight function for each label), and as we will show, WF-GSOS subsumes both WGSOS and Segala-GSOS formats. On a different direction, in [12] De Nicola et al. provide a “meta-calculus” for describing stochastic systems and their semantics as FuTS, showing that in several cases behavioural equivalences are congruences. This interesting approach is complementary to ours, since it provides some “syntactic-semantic basic blocks” to be assembled, instead of a general rule format.

The rest of the paper is structured as follows. In Section 2 we recall Uniform Labelled Transition Systems, and their bisimulation. In Section 3 we introduce the Weight Function GSOS specification format for the syntactic presentation of ULTraSs. In Section 4 we provide some 1The reader aware of advanced process calculi baffled will be not by the fact that targets are not processes. Well-known previous examples are the LTS abstractions/concretions for the calculus, for the applied calculus, for the ambient calculus, etc.
application examples, such as a WF-GSOS specification for PEPA and the translations of Segala-GSOS and WGSOS specifications in the WF-GSOS format. The categorical presentation of ULTraS and WF-GSOS, with the results that the format is sound and complete and bisimilarity is a congruence, are in Section 5. Final remarks and directions for future work are in Section 6.

2 Uniform Labelled Transition Systems and their bisimulation

In this Section we recall and elaborate the definition of ULTraSs, and define the corresponding notion of (coalgebraically derived) bisimulation; finally we compare it with the notion of M-bisimulation presented in [5]. Additional examples are provided in the Appendix. Although we focus on the ULTraS framework, the results and methodologies described in this paper can be ported to similar formats (like FuTS [24]), and more generally to a wide range of systems combining computational aspects in different ways.

2.1 Uniform Labelled Transition Systems

ULTraS are (non-deterministic) labelled transition systems whose transitions lead to state reachability weight functions, i.e. functions representing quantitative informations about “how” each state can be reached. Examples of weight functions include probability distributions, resource consumption levels, or stochastic rates. Under this light, ULTraS can be thought as a generalization of Segala systems [31], which stratify non-determinism over probabilism. Following the parallel with Segala systems, ULTraS transitions can be pictured as being composed by two steps:

\[ x \xrightarrow{\rho} y \]

where the first is a labelled non-deterministic (sub)transition and the second is a weighted one; from this perspective the weight function plays the rôle of the “hidden intermediate state”.

Akin to Weighted Labelled Transition Systems (WLTS) [13, 22, 32], weights are drawn from a fixed set endowed with a commutative monoid structure, where the unit is meant to be assigned to disabled transitions (i.e. those yielding unreachable states) and the monoidal addition is used to compositionally weight sets of transitions given by non-determinism.

Definition 1 (W-ULTraS). Given a commutative monoid \( W = (W, +, 0) \) a \( W \)-weighted Uniform Labelled Transition System is a triple \( (X, A, \rightarrow) \) where:

- \( X \) is a set of states (processes);
- \( A \) is a set of labels (actions);
- \( \rightarrow \subseteq X \times A \times [X \rightarrow W] \) is a transition relation where \([X \rightarrow W]\) denotes the set of weight functions from \( X \) to \( W \).

Monoidal addition does not play any rôle in the above definition but it is crucial to define the notion of bisimulation by uniformly providing a compositional way to weight sets of outgoing transitions (e.g. stochastic or probabilistic bisimulations). Since total weights are defined by summation, some guarantees on the cardinality of these sets are needed.

Definition 2 (Image boundedness). Let \( W = (W, +, 0) \) be a monoid. For a function \( \rho : X \rightarrow W \) the set \( \{ x \mid \rho(x) \neq 0 \} \) is called support of \( \rho \) (written \( \lceil \rho \rceil \)). Let \( \kappa, \kappa' \) be two ordinals; a function \( \rho : X \rightarrow W \) is \( \kappa \)-supported iff \( \lceil \rho \rceil < \kappa \). A \( W \)-ULTraS \((X, A, \rightarrow)\) is \( (\kappa, \kappa') \)-bounded iff for any state \( x \in X \) and label \( a \in A \) the set \( \{ \rho \mid x \xrightarrow{a} \rho \} \) has cardinality less than \( \kappa \) and contains only \( \kappa' \)-supported functions. (We shall drop \( \kappa' \) when it is equal to \( \kappa \).)

The notion of image boundedness guarantees that the branching transitions do not exceed the expressive power of summation of the underlying monoid in the sense that, if sum is defined for

\[ W \] is an ordered set with bottom. Actually, the order is not crucial to the basic notion of ULTraS as it is only used by some equivalences considered in that paper.
any family of cardinality lesser than \( \kappa \), then for any state in a \( \kappa \)-bounded system the total weights for sets of outgoing transitions are always defined. Henceforth, for the sake of simplicity, we will restrict ourselves to image-finite systems (i.e. \( \omega \)-bounded), but the development can be generalized throughout.

**Definition 3.** Let \( \mathfrak{M} = (W, +, 0) \) be a commutative monoid. For any set \( X \) let \( \mathcal{F}_\mathfrak{M} X \) be the set of finitely supported weight functions over \( X \), i.e., \( \mathcal{F}_\mathfrak{M} X \triangleq \{ \rho : X \to W \mid \| \rho \| < \omega \} \). For \( \rho \in \mathcal{F}_\mathfrak{M} X \) and \( Y \subseteq X \) let us define \( \rho(Y) \triangleq \sum_{x \in Y \cup \{ \rho \}} \rho(x) \). Finally the total weight of \( \rho \) is \( \| \rho \| \triangleq \rho(X) \).

### 2.2 Bisimulation

We present now the definition of bisimulation for ULTraS. As we will see in Section 5, this arises directly from their coalgebraic characterization of ULTraS.

Let \( R \subseteq X \times Y \) be a relation between two sets \( X \) and \( Y \). The **subset closure** of \( R \) is the smallest relation \( R^* \subseteq \varphi(X) \times \varphi(Y) \), such that, for \( C \subseteq X, D \subseteq Y \):

\[
(C, D) \in R^* \iff (\forall x \in C, \forall y \in Y : (x, y) \in R \Rightarrow y \in D) \land \\
(\forall x \in X, \forall y \in D : (x, y) \in R \Rightarrow x \in C)
\]

**Definition 4 (Bisimulation).** Let \( (X, \xrightarrow{\cdot} X) \) and \( (Y, \xrightarrow{\cdot} Y) \) be two image-finite \( \mathfrak{M} \)-ULTraS. A relation \( R \) between \( X \) and \( Y \) is a bisimulation if, and only if, for each pair of states \( x \in X \) and \( y \in Y \), \( (x, y) \in R \) implies that for each label \( a \in A \) the following hold:

\begin{itemize}
  \item if \( x \xrightarrow{a} X \varphi \) then there exists \( y \xrightarrow{a} Y \psi \) s.t. for all \( (C, D) \in R^* : \varphi(C) = \psi(D) \);
  \item if \( y \xrightarrow{a} Y \psi \) then there exists \( x \xrightarrow{a} X \varphi \) s.t. for all \( (C, D) \in R^* : \varphi(C) = \psi(D) \).
\end{itemize}

Processes \( x \) and \( y \) are said to be bisimilar if there exists a bisimulation relation \( R \) s.t. \( (x, y) \in R \).

As ULTraSs can be seen as stacking non-determinism over other computational behaviour, Definition 4 stratifies bisimulation for non-deterministic labelled transition system over bisimulation for systems expressible as labelled transition systems weighted over commutative monoids. In fact, two processes \( x \) and \( y \) are related by some bisimulation if, and only if, whether one reaches a weight function via a non-deterministic labelled transition, the other can reach another function via a transition with the same label, where the two functions are equivalent in the sense that they assign the same total weight to the classes of states in the relation. For instance, in the case of weights being probabilities, functions are considered equivalent only when they agree on the probabilities assigned to each class of states which is precisely the intuition behind probabilistic bisimulation [23]. More examples will be discussed below or in the Appendix.

**Constrained ULTraS** Sometimes, the ULTraSs induced by a given monoid are too many, and we have to restrict to a subclass. For instance, fully-stochastic systems such as (labelled) CTMCs are a strict subclass of ULTraSs weighted over the monoid of non-negative real numbers \( \mathbb{R}_+^* \), where weights express rates of exponentially distributed continuous time transitions. In the case of fully-stochastic systems, for each label, each state is associated with precisely one weight function. This kind of “deterministic” ULTraSs are called functional in [5], because the transition relation is functional, and correspond precisely to WLTSs [13, 22, 32]. These are a well-known family of systems (especially their automata counterpart) and have an established coalgebraic understanding as long as a (coalgebraically derived) notion of weighted bisimulation which are shown to subsume several known kinds of systems such as non-deterministic, (fully) stochastic, generative and reactive probabilistic [22]. Moreover, Definition 4 coincides with weighted bisimulation on functional ULTraSs/WLTSs over the same monoid [22 Def. 4]; hence Definition 4 covers every system expressible in the framework of WLTS. (cf. Appendix A).

**Proposition 1.** Let \( \mathfrak{M} \) be a commutative monoid and \( (X, \rightarrow) \) be a \( \mathfrak{M} \)-LTS seen as a functional ULTraS on \( \mathfrak{M} \). Every bisimulation relation on \( \rightarrow \) is a \( \mathfrak{M} \)-weighted bisimulation and vice versa.

**Proof.** See Appendix A page 20.
Another constraint arises in the case of probabilistic systems, i.e., weight functions are probability distribution. Since addition is not a closed operation in the unit interval $[0,1]$, there is no monoid $\mathfrak{M}$ such that every weight function on it is also a probability distributions. We could relax Definition 1 to allow commutative partial monoid\footnote{A commutative partial monoid is a set endowed with a unit and a partial binary operation which is associative and commutative, where it is defined, and always defined on its unit.} such as the weight structure of probabilities $([0,1],+)$; Unfortunately, not every weight function on $[0,1]$ is a probability distribution. In fact, probabilistic systems (among others) can be recovered as ULTraSs over the $([0,1],+)$ (i.e. the free completion of $([0,1],+)$) and subject to suitable constraints. For instance, Segala systems\footnote{In the original presentation, $\mathcal{M}$ is required to consistently weight also sequences of transitions, in order to cover also trace equivalence; since this Section focuses on strong bisimulations only, this information will be omitted.} are precisely the strict subclass of $\mathbb{R}^+_0$-ULTraS such that every weight function $\rho$ in their transition relation is a probability distribution i.e. $\|\rho\|=1$. Moreover, bisimulation is preserved by constraints. For instance, bisimulations on the above class of (constrained) ULTraS are Segala’s (strong) bisimulations (cf. \cite[Def. 13]{Segala}), and vice versa.

**Proposition 2.** Let $(X, A, \rightarrow)$ be an image-finite Segala system seen as a ULTraS on $([0,1],+)$.
Every bisimulation relation on $\rightarrow$ is a strong bisimulation in the sense of \cite[Def. 13]{Segala} and vice versa.

**Proof.** See Appendix\textsuperscript{B} page 21.

A similar result holds for reactive and generative (or fully) probabilistic systems and their bisimulations. In fact, these are functional ULTraSs with weight functions in $\mathcal{F}_{\mathbb{R}^+_0}$ and subject to constraints $\forall x \in X : \sum_{a \in A, y \in X} \rho(x,a,y) \in \{0,1\}$ and $\forall x \in X, a \in A : \sum_{y \in X} \rho(x,a,y) \in \{0,1\}$ respectively.

### 2.3 Comparison with $\mathcal{M}$-bisimulation

Bernardo et al. defined a notion of bisimulation for ULTraS parametrized by a function $\mathcal{M}$ which is used to weight sets of transitions\footnote{In the original presentation, $\mathcal{M}$ is required to consistently weight also sequences of transitions, in order to cover also trace equivalence; since this Section focuses on strong bisimulations only, this information will be omitted.} such as the weight structure of probabilistic systems and their bisimulations.

**Definition 5** ($\mathcal{M}$-function). Let $(\mathcal{M}, \bot)$ be a pointed set and $(X, A, \rightarrow)$ be a $\mathfrak{M}$-ULTraS. A function $\mathcal{M} : X \times A \times \mathcal{P}X \rightarrow \mathcal{M}$ is an $\mathcal{M}$-function for $(X, A, \rightarrow)$ if, and only if, it agrees with termination and class union, i.e.:

- for all $x \in X$, $a \in A$ and $C \subseteq \mathcal{P}X$, $\mathcal{M}(x,a,C) = \bot$ whenever $\rho(C) = 0$ for every $x \xrightarrow{a} \rho$
- or there is no $\rho$ at all;

- for all $x, y \in X$, $a \in A$ and $C_1, C_2 \subseteq \mathcal{P}X$, if $\mathcal{M}(x,a,C_1) = \mathcal{M}(y,a,C_1)$ and $\mathcal{M}(x,a,C_2) = \mathcal{M}(y,a,C_2)$ then $\mathcal{M}(x,a,C_1 \cup C_2) = \mathcal{M}(y,a,C_1 \cup C_2)$.

**Definition 6** ($\mathcal{M}$-bisimulation\textsuperscript{[5]}). Let $\mathcal{M}$ be an $\mathcal{M}$-function for $(X, A, \rightarrow)$. A relation $R \subseteq X \times X$ is a $\mathcal{M}$-bisimulation for $\rightarrow$ iff for each pair $(x,y) \in R$, label $a \in A$, and class $C \in (X/R)$, the following holds:

$$\mathcal{M}(x,a,C) = \mathcal{M}(y,a,C).$$

In some sense, the notion of $\mathcal{M}$-bisimulation is more general than Definition 4 since (sets of) transitions are not necessarily weighted in the same structures. For instance, stochastic rates can be considered up-to a suitable tolerance in order to account for experimental measurement errors in the model.

A further distinction between bisimulation and $\mathcal{M}$-bisimulation arises from the fact that ULTraSs come with two distinct way of terminating. A state can be seen as “terminated” either when its outgoing transitions are always the constantly zero function, or when it has no transitions at all. In the first case, the state has still associated an outcome, saying that no further state is reachable; we call these states terminal. In the second case, the LTS does not even tell us that the state cannot reach any further state; in fact, there is no “meaning” associated to the state.
In this case, we say that the state is `stack`<sup>3</sup> The bisimulation given in Definition 4 keeps these two terminations as different (i.e., they are not bisimilar), whereas $M$-bisimulation does not make this distinction (cf. [3] Def. 3.2 or, for a concrete example based on Segala systems, [4] Def. 7.2).

Finally, the two notions differ on the quantification over equivalence classes: in the case of Definition 4 quantification depends on the non-deterministic step whereas in the case of $M$-bisimulation it does not.

Under some mild assumptions, the two notions agree. In particular, let us restrict to systems with just one of the two terminations for each action $a$—i.e. if for some $x$, $\{\rho \mid x \xrightarrow{a} \rho\} = \emptyset$ then for all $y$, $\lambda z.0 \notin \{\rho \mid y \xrightarrow{a} \rho\}$, and, symmetrically, if for some $x$, $\lambda z.0 \in \{\rho \mid x \xrightarrow{a} \rho\}$ then for all $y$, $\{\rho \mid y \xrightarrow{a} \rho\} \neq \emptyset$. Then, the bisimulation given in Definition 4 corresponds to a $M$-bisimulation for a suitable choice of $M$.

**Proposition 3.** Let $(X,A,\rightarrow)$ be a $\mathfrak{M}$-ULTraS with at most one kind of termination, for each label. Every bisimulation $R$ is also an $M$-bisimulation for

$$M(x,a,C) \triangleq \{[\rho]_{\equiv_R} \mid x \xrightarrow{a} \rho \land \rho(C) \neq 0\} \sqcup \bot$$

where $\varphi \equiv_R \psi \iff \forall C \in X/R. \varphi(C) = \psi(C)$, $M = \mathcal{T}(\mathfrak{M} X/\equiv_R)$, and $\bot = \{[\lambda z.0]_{\equiv_R}\}$.

**Proof.** See Appendix C, page 21 □

### 3 WF-GSOS: A complete GSOS format for ULTraSs

In this section we introduce the Weight Function GSOS specification format for the syntactic presentation of ULTraSs. As it will be proven in Section 5.2, bisimilarity for systems given in this format is guaranteed to be a congruence with respect to the signature used for representing processes.

The format is parametric in the weight monoid $\mathfrak{M}$ and, as usual, in the process signature $\Sigma$ defining the syntax of system processes. In contrast with “classic” GSOS formats [20], targets of rules are not processes but terms whose syntax is given by a different signature, called the weight signature. This syntax can be thought of as an “intermediate language” for representing weight functions along the line of viewing ULTraSs as stratified (or staged) systems. An early example of this approach can be found in [2], where targets are terms representing measures over the continuous state space. Earlier steps in this direction can be found e.g. in Bartels’ GSOS format for Segala systems (cf. [4] §5.3 and [26] §4.2) or in [5] where targets are described by meta-expressions.

**Definition 7 (WF-GSOS Rule).** Let $\mathfrak{M}$ be a commutative monoid and $A$ a set of labels. Let $\Sigma$ and $\Theta$ be the process signature and the weight signature respectively. A WF-GSOS rule over them is an expression of the form:

$$\begin{align*}
\{ x_i \xrightarrow{a} \varphi_{ij} \}_{1 \leq i \leq n, \ 1 \leq j \leq m^a_i} & \quad \{ x_i \xrightarrow{b} \}_{1 \leq i \leq n, \ 1 \leq j \leq m^b_i} & \quad \{ \varphi_{ij,k}^{a,b} = w_k \}_{1 \leq k \leq p} & \quad \{ \varphi_{ij,k}^{a,b} \ni y_k \}_{1 \leq k \leq q} \\
& \quad \{ x_1, \ldots, x_n \xrightarrow{f} \psi \}
\end{align*}$$

where:

- $f$ is an $n$-ary symbol from $\Sigma$;
- $X = \{x_i \mid 1 \leq i \leq n\}$, $Y = \{y_k \mid 1 \leq k \leq q\}$ are sets of pairwise distinct process variables;
- $\Phi = \{\varphi_{ij}^a \mid 1 \leq i \leq n, \ a \in A_i, \ 1 \leq j \leq m^a_i\}$ is a set of pairwise distinct weight function variables;
- $\{w_k \in \mathfrak{M} \mid 1 \leq k \leq p\}$ are weight constants;

---

<sup>3</sup>This is akin to sequential programs: a terminal state is when we reach the end of the program; a stuck state is when we are at executing an instruction whose meaning is undefined.
• \( \{c_k \mid 1 \leq k \leq q, w_k \in C_k \} \) is a set of clubs of \( \mathfrak{M} \), i.e. subsets of \( W \) being monoid ideals whose complements are sub-monoids of \( \mathfrak{M} \):

• \( a, b, c \in A \) are labels and \( A_i \cap B_i = \emptyset \) for \( 1 \leq i \leq n \);

• \( \psi \) is a weight term for the signature \( \Theta \) such that \( \text{var}(\psi) \subseteq X \cup Y \cup \Phi \).

A rule like above is triggered by a tuple \((C_1, \ldots, C_n)\) of enabled labels and by a tuple \((v_1, \ldots, v_p)\) of weights if, and only if, \( A_i \subseteq C_i, B_i \cap C_i = \emptyset \) and, \( w_j = v_j \) for \( 1 \leq i \leq n \) and \( 1 \leq j \leq p \).

Intuitively, the four families of premises can be grouped in two kinds: the first two families correspond to the non-deterministic (and labelled) behaviour, whereas the other two correspond to the weighting behaviour of quantitative aspects. The former are precisely the premises of GSOS rules for LTSs (up-to targets being functions), and describe the possibility to perform some labelled transitions. The latter are inspired by Bartels’ Segala-GSOS [4] §5.3 and Klin’s WGSOS [22] formats; a premise like \( \| \varphi \| = w \) constrains the variable \( \varphi \) to those functions whose total weight is exactly the constant \( w \); a premise like \( \varphi \) \( \in \mathcal{E} \) \( \exists \ y \) binds the process variable \( y \) to those elements being assigned a weight in \( C \). Clubs are substructures of commutative monoids that are “isolated” w.r.t. the monoidal operation in the sense that:

• are commutative monoid ideals, i.e. subsets \( \mathfrak{C} \) with a module structure: \( \text{img}(+_{|\mathfrak{M} \times \mathfrak{C}|}) \subseteq \mathfrak{C} \);

• their complement \( \overline{\mathfrak{C}} \) in \( \mathfrak{M} \) is a sub-monoid of \( \mathfrak{M} \): \( \text{img}(+_{|\mathfrak{M} \times \overline{\mathfrak{C}}|}) \subseteq \overline{\mathfrak{C}} \).

This isolation property means that clubs are not affected by variables substitutions (in fact, clubs form the finest topology on a commutative monoid enforcing this property).

Likewise Segala-GSOS (but not WGSOS), there are no variables denoting the weight of each \( y_k \) since this information can be readily extracted from \( \phi_{ik,jk}^{\alpha} \), e.g. by some operator from \( \Theta \) that “evaluates” \( \phi_{ik,jk}^{\alpha} \) on \( y_k \).

Targets of transitions defined by these rules are terms generated from the signature \( \Theta \). In order to characterize transition relation for ULTraSs, we need to evaluate these terms to weight functions. This is obtained by adding an interpretation for weight terms, besides a set of rules in the above format.

Before defining interpretations and specifications, we need to introduce some notation. For a signature \( S \) and a set \( X \) of variable symbols, let \( T_S X \) denote the set of terms freely generated by \( S \) over the variables \( X \) (in the following, \( S \) will be either \( \Sigma \) or \( \Theta \)). A substitution for symbols in \( X \) is any function \( \sigma : X \to Y \); its action extends to terms defining the function \( T_S(\sigma) : T_S X \to T_S Y \) (i.e. simultaneous substitution). When confusion seems unlikely we use the more evocative \( t[\sigma] \) instead of \( T_S(\sigma)(x) \).

A variable substitution \( \sigma : X \to Y \) induces also a function \( F_{\mathfrak{M}}(\sigma) : F_{\mathfrak{M}} X \to F_{\mathfrak{M}} Y \), mapping (finitely supported) weight functions over \( X \) to (finitely supported) weight functions over \( Y \), as follows:

\[
F_{\mathfrak{M}}(\sigma)(\rho) = \lambda y : Y. \sum_{x : \sigma(x) = y} \rho(x).
\]  

Consistently, we denote the action of \( \sigma \) on \( \rho \) by \( \rho[\sigma] \).

**Definition 8 (Interpretation).** Let \( \mathfrak{M} \) be a commutative monoid, \( \Sigma \) and \( \Theta \) be the process and the weight signature respectively. A weight term interpretation for them is a family of functions

\[
\llbracket \cdot \rrbracket_X : T^\Theta(X + F_{\mathfrak{M}}(X)) \to \mathcal{P} F_{\mathfrak{M}} T^\Sigma(X)
\]

indexed over sets of variable symbols, and respecting substitutions, i.e.:

\[
\forall \sigma : X \to Y, \psi \in T^\Theta(X) : \llbracket \psi \rrbracket_X[\sigma] = \llbracket \psi[\sigma] \rrbracket_Y.
\]

Differently from [22] interpretations allow one term to represent finitely many weight functions. This generalization offers more freedom in the use of the format by reducing the constrains on what can be encoded in weight function terms and simplifies the proof for completeness.

We are ready to introduce the WF-GSOS specification format. Basically, this is a set of WF-GSOS rules, subject to some finiteness conditions to ensure image-finiteness, together with an interpretation.
In this Section we provide some examples of applications of the WF-GSOS format. First, we show how a process calculus can be given a WF-GSOS specification; in particular, we consider PEPA, a well known process algebra with quantitative features. Then we show that Klin’s Weighted GSOS format for weighted systems [22] and Bartels’ Segala-GSOS format for Segala systems [4] are subsumed by our WF-GSOS format; this corresponds to the fact that ULTraSs subsume both weighted and Segala systems.

Every WF-GSOS specification induces an ULTraS over ground process terms.

**Definition 9** (WF-GSOS specification). Let $\mathcal{M}$ be a commutative monoid, $\Lambda$ the set of labels, $\Sigma$ and $\Theta$ the process and the weight signature respectively. An image-finite WF-GSOS specification over $\mathcal{M}, \Lambda, \Sigma$ and $\Theta$ is a pair $(\mathcal{R}, \llbracket-\rrbracket)$ where $\llbracket-\rrbracket$ is a weight term interpretation and $\mathcal{R}$ is a set of rules compliant with Definition 7 and such that only finitely many rules share the same operator in the source $(\ell)$, the same label in the conclusion $(c)$, and the same trigger $(A_1, \ldots, A_n)$, $\langle w_1, \ldots, w_p \rangle$.

The bisimulation on the ULTraS induced by a WF-GSOS specification $(\mathcal{R}, \llbracket-\rrbracket)$ over $\mathcal{M}, \Lambda, \Sigma$ and $\Theta$ is the ULTraS $\ultraS{\mathcal{R}}{\llbracket-\rrbracket}$ where $\ultraS{\mathcal{R}}{\llbracket-\rrbracket}$ is the smallest subset of $\ultraS{\Sigma}{\Lambda} \times \mathcal{M} \times \ultraS{\mathcal{R}}{\llbracket-\rrbracket}$ being closed under the following condition.

Let $p = \ell(a_1, \ldots, a_n) \in \ultraS{\Sigma}{\Lambda}$. Since the ground $\Sigma$-terms $p_i$ are structurally smaller than $p$ assume (by structural recursion) that the set $\{ \rho \mid p_i \llbracket a_1 \ldots \rrbracket \Rightarrow \rho \}$ and hence the trigger $\bar{A} = \langle A_1, \ldots, A_n \rangle$, $\bar{w} = \langle w_1, \ldots, w_q \rangle$ is determined for every $i \in \{1, \ldots, n\}$ and $a \in A$. For any rule $R \in \mathcal{R}$ whose conclusion is in the form $\ell(x_1, \ldots, x_n) \llbracket \psi \rrbracket \Rightarrow \psi$ and triggered by $\bar{A}$ and $\bar{w}$ let $X, Y, \Phi$ be the set of process and weight function variables involved in $R$ as per Definition 6. Then, for any substitution $\sigma : X \cup Y \rightarrow \ultraS{\Sigma}{\Lambda}$ and map $\theta : \Phi \rightarrow \ultraS{\mathcal{M}}{\Sigma}$ such that:

1. $\sigma(x_i) = p_i$ for $x_i \in X$.
2. $\theta(\varphi^x_{ij}) = \rho$ for each premise $x_i \llbracket \varphi^x_{ij} \rrbracket$ and $\| \varphi^x_{ij} \| = w_k$ of $R$ and for any $\rho$ s.t. $p_i \llbracket a \rrbracket \Rightarrow \rho$ and $\| \rho \| = w_k$.
3. $\sigma(y_k) = q_k$ for each premise $\varphi^a_{ik} \llbracket \mathcal{C}_k \rrbracket \ni y_k$ of $R$ and for any $q_k \in \ultraS{\Sigma}{\Lambda}$ s.t. $\theta(\varphi^a_{ik}) \llbracket (q_k) \rrbracket \in \mathcal{C}_k$ there is $\rho \llbracket \mathcal{C}_k \rrbracket X \cup Y \rightarrow \ultraS{\Sigma}{\Lambda}$ the instantiation of a possible interpretation of the target $\Theta$-term $\psi$.

The above definition is well-defined since it is based on structural recursion on ground $\Sigma$-terms (i.e. the process $p$ in each triple $(p, a, \rho)$); in particular, terms have finite depth and only structurally smaller terms are used by the recursion (i.e. the assumption of $p_i \llbracket a \rrbracket \Rightarrow \rho$ being defined for each $p_i$ in $p = \ell(p_1, \ldots, p_n)$). Moreover, for any trigger, operator, and conclusion label only finitely many rules have to be considered.

Finally we can state the main adequacy result for the proposed format.

**Theorem 4** (Congruence). The bisimulation on the ULTraS induced by a WF-GSOS specification is a congruence with respect to the process signature.

The proof is postponed to Section 5.2 where we will take advantage of the bialgebraic framework.

**Remark 1** (Expressing interpretations). Weight term interpretation can defined in many ways, such as structural recursion or $\lambda$-iteration [2]. For instance, every substitutions-respecting family of maps:

$$h_X : \Theta \ultraS{\mathcal{M}}{\Sigma}(X) \rightarrow \ultraS{\mathcal{M}}{\Sigma}(X) \quad b_X : X \rightarrow \ultraS{\mathcal{M}}{\Sigma}(X)$$

uniquely extends to an interpretation by structural recursion on $\Theta$-terms where $h_X$ and $b_X$ define the inductive and base cases respectively. These maps can be easily given by means of a set of equations, as in [26] §4.1.

## 4 Examples and applications of WF-GSOS specifications

In this Section we provide some examples of applications of the WF-GSOS format. First, we show how a process calculus can be given a WF-GSOS specification; in particular, we consider PEPA, a well known process algebra with quantitative features. Then we show that Klin’s Weighted GSOS format for weighted systems [22] and Bartels’ Segala-GSOS format for Segala systems [4] are subsumed by our WF-GSOS format; this corresponds to the fact that ULTraSs subsume both weighted and Segala systems. 

8
(ACT) \[ (a,r).P \xrightarrow{a,r} P \]

(CH1) \[ P_1 \xrightarrow{a,r} P_2 \xrightarrow{a,r} Q \]

(CH2) \[ P_1 \xrightarrow{a,r} P_2 \xrightarrow{a,r} Q \]

(H1) \[ P \xrightarrow{a,r} Q \]

(H2) \[ P \xrightarrow{a,r} Q \]

(COL) \[ P_1 \xrightarrow{a,r} Q_1 \parallel P_2 \xrightarrow{a,r} Q_2 \]

(COL2) \[ P_1 \xrightarrow{a,r} Q_1 \parallel P_2 \xrightarrow{a,r} Q_2 \]

Figure 1: Structural operational semantics for PEPA.

### 4.1 WF-GSOS for PEPA

In PEPA \cite{16,17}, processes are terms over the grammar:

\[ P ::= (a,r).P \mid P + P \mid P \parallel P \mid P \parallel L \]

where \( a \) ranges over a fixed set of labels \( A \), \( L \) over subsets of \( A \) and \( r \) over \( \mathbb{R}^+ \). The semantics of process terms is usually defined by the inference rules in Figure 1 where \( a \in A \), \( r, r_1, r_2, R \in \mathbb{R}^+ \) (passive rates are omitted for simplicity) and \( R \) depends only on \( r_1, r_2 \) and the intended meaning of synchronisation. For instance, in applications to performance evaluation \cite{16}, rates model times and \( R \) is defined by the **minimal rate law**:

\[ R = \frac{r_1}{r_a(P_1)} \cdot \frac{r_2}{r_a(P_2)} \cdot \min(r_a(P_1), r_a(P_2)) \]

where \( r_a \) denotes the apparent rate of \( a \). In systems biology \cite{8}, rates model molecules concentrations and \( R \) is defined by the **multiplicative law**: \( R = r_1 \cdot r_2 \).

PEPA can be characterized by a specification in the WF-GSOS format where the process signature \( \Sigma \) is the same as \cite{2} and weights are drawn from the monoid of positive real numbers under addition extended with the \( +\infty \) element (only for technical reasons connected with the \( \sqcup \) and process variables—differently from other stochastic process algebras like EMPA \cite{6}). PEPA does not allow instantaneous actions, i.e. with rate \( +\infty \). The intermediate language of weight terms is expressed by the grammar:

\[ \theta ::= \perp \mid \diamond_r(\theta) \mid \theta_1 \sqcup \theta_2 \mid \theta_1 \parallel L \theta_2 \mid \xi \mid P \]

where \( r \in \mathbb{R}^+_\ast \), \( L \subseteq A \), \( \xi \) range over weight functions, and \( P \) over processes. Note that the grammar is untyped since it describes the terms freely generated by the signature \( \Theta = \{ \perp, \diamond_r : 1, \sqcup : 2, \parallel_L : 2 \} \), over weight function variables and processes. Intuitively \( \perp \) is the constantly 0 function, \( \diamond_r \) reshapes its argument to have total weight \( r \), \( \sqcup \) is the point-wise sum and \( \parallel_L \) parallel composition e.g. by \cite{3}. The formal meaning of these operators is given below by the definition (by structural recursion on \( \Theta \)-terms) of the interpretation \( \langle \cdot \rangle \) which is introduced alongside WF-GSOS rules for presentation convenience. Each operator is interpreted as a singleton (PEPA describes functional ULTraSs) and hence we will describe \( \langle \cdot \rangle \) as if a weight function is returned.

For each action \( a \in A \) and rate \( r \in \mathbb{R}^+ \), a process \( (a,r).P \) presents exactly one \( a \)-labelled transition ending in the weight function assigning \( r \) to the (sub)process denoted by the variable \( P \) and 0 to everything else. Hence, the **action axiom** is expressed as follows:

\[ (a,r).P \xrightarrow{a} \diamond_r(P) \]

\[ \langle \diamond_r(\psi) \rangle_X(t) = \begin{cases} \frac{r}{||\psi||_X} & \text{if } ||\psi||_X(t) \neq 0 \\ 0 & \text{otherwise} \end{cases} \]

where \( \diamond_r \) reshapes \( \langle P \rangle_X \) to equally distribute the weight \( r \) over its support; in particular, since process variables will be interpreted as “Dirac-like” functions \( \diamond_r(P) \) corresponds to the weight function assigning \( r \) to \( \Sigma \)-term denoted by \( P \).

\*\*Since the interpretation \( \langle \cdot \rangle \) is being defined by structural recursion and has to cover all the language freely generated from \( \Theta \), we can not use the (slightly more intuitive) “Dirac” operator \( \delta_r(P) \) where \( P \) is restricted to be a process variable instead of a \( \Theta \)-term. Likewise, indexing \( \delta_r, P \) also over processes would break substitution independence i.e. naturality.\*\*
Conversely to the action axiom, \((a,r).P\) can not perform any action but \(a\):

\[
(a,r).P \xrightarrow{b} \perp \quad \mathbb{I}_X(t) = 0
\]

This rule is required to obtain a functional ULTraS and is implicit in Figure II where disabled transitions are assumed with rate 0 as in any specification in the Stochastic GSOS or Weighted GSOS formats. Without this rule, transitions would have been disabled in the non-deterministic layer i.e. \((a,r).P \xrightarrow{b}\).

Stochastic choice is resolved by the stochastic race, hence the rate of each competing transition is added point-wise as in Figure II (and in the SGOS and WGSOS formats). This passage belongs to the stochastic layer of the behaviour (hence to the interpretation, in our setting) whereas the selection of which weight functions to combine is in the non-deterministic behaviour represented by the rules and, in particular, to the labelling. Therefore, the choice rules become:

\[
P_1 \xrightarrow{a} \varphi_1 \quad P_2 \xrightarrow{a} \varphi_2 \\
P_1 + P_2 \xrightarrow{a} \varphi_1 \oplus \varphi_2
\]

\[
\langle \psi \oplus \varphi \rangle X(t) = \langle \psi \rangle X(t) + \langle \varphi \rangle X(t)
\]

Likewise, process cooperation depends on the labels to select the weight function to be combined. This is done in the next two rules: one when the two processes cooperate, and the other when one process does not interact on the channel:

\[
P_1 \xrightarrow{a} \varphi_1 \quad P_2 \xrightarrow{a} \varphi_2 \\
P_1 \parallel P_2 \xrightarrow{a} \varphi_1 \parallel \varphi_2
\]

\[
P_1 \xrightarrow{a} \varphi_1 \quad P_2 \xrightarrow{a} \varphi_2 \\
(P_1 \parallel P_2) \xrightarrow{a} (\varphi_1 \parallel \varphi_2)
\]

The combination step depends on the specific e.g. in the case of GSOS:

\[
\langle \psi \parallel \varphi \rangle X(t) = \begin{cases} \\
\langle \psi \rangle X(t) \cdot \langle \varphi \rangle X(t) & \text{if } t_1 \parallel t_2 \\
0 & \text{otherwise}
\end{cases}
\]

Each process is interpreted as a weight function over process terms. This is achieved by a Dirac-like function assigning \(+\infty\) to the \(\Sigma\)-term composed by the aforementioned variable: \(\langle P \rangle X(t) = +\infty\) if \(P = t\), 0 otherwise. The infinite rate characterizes instantaneous actions as if all the mass is concentrated in the variable; e.g., in interactions based on the minimal rate law, processes are not consumed. For the same reason, if we were dealing with concentration rates and the multiplicative law, we would assign 1 to \(P\).

The remaining rules for hiding are straightforward:

\[
P \xrightarrow{a} \varphi \\
P \xrightarrow{a} \varphi
\]

\[
P \xrightarrow{a} \varphi \\
P \xrightarrow{a} \varphi
\]

This completes the definition of \(\langle \cdot \rangle\) by structural recursion and hence the WF-GSOS specification of PEPA. It is easy to check that the induced ULTraS is functional and correspond to the stochastic system of PEPA processes; that bisimulations on it are stochastic bisimulations (and vice versa) and that bisimilarity is a congruence with respect to the process signature.

4.2 Segala-GSOS

In [4], Bartels proposed a GSOS specification format\footnote{Segala-GSOS specifications yield distributive laws for Segala systems but, to the best of authors knowledge, it still is an open problem whether every such distributive law arises from some Segala-GSOS specification.} for Segala systems (hence Segala-GSOS), i.e. ULTraS where weight functions are exactly probability distributions. We recall Bartels’ definition, with minor notational differences.

**Definition 11** ([4] §5.3). A GSOS rule for Segala systems is a rule of the form

\[
\begin{cases} \\
x_i \xrightarrow{a} \varphi_{ij} & 1 \leq i \leq n, \ a \in A_i, \ 1 \leq j \leq m_i \\
x_i \xrightarrow{b} \varphi_k & 1 \leq i \leq n, \ b \in B_i, \ 1 \leq k \leq m_i \\
\end{cases}
\]

\[
f(x_1, \ldots, x_n) \xrightarrow{c} w_1 \cdot t_1 + \cdots + w_m \cdot t_m
\]

where:
\begin{itemize}
  \item $f$ is an $n$-ary symbol from $\Sigma$;
  \item $X = \{x_i | 1 \leq i \leq n\}$, $Y = \{y_k | 1 \leq k \leq m\}$, and $V = \{\varphi_{ij}^a | 1 \leq i \leq n, a \in A_i, 1 \leq j \leq m_i^a\}$ are pairwise distinct process and probability distribution variables respectively;
  \item $a, b, c \in A$ are labels and $A_i \cap B_i = \emptyset$ for any $i \in \{1, \ldots, n\}$;
  \item $t_1, \ldots, t_m$ are target terms on variables $X$, $Y$ and $V$; the latter are associated with colours from a finite palette to indicate different instances;
  \item $\{w_i \in (0, 1] | 1 \leq i \leq m\}$ are weights associated to the target terms and s.t. $\sum_{1 \leq i \leq m} w_i = 1$.
\end{itemize}

A rule like above is triggered by a tuple $\langle C_1, \ldots, C_n \rangle$ of enabled labels if, and only if, $A_i \subseteq C_i$ and $B_i \cap C_i = \emptyset$ for each $i \in \{1, \ldots, n\}$. A GSOS specification for Segala systems is a set of rules in the above format containing finitely many rules for any source symbol $f$, conclusion label $c$ and trigger $\hat{C}$.

Segala-GSOS specifications can be easily turned into WF-GSOS ones. The first two families of premises are translated straightforwardly to the corresponding ones in our format; the third can be turned into those of the form $[\varphi] \Rightarrow y$. Targets of transitions describe finite probability distributions and are evaluated to actual probability distributions by a fixed interpretation of a form similar to Definition 5. Some care is needed to handle copies of probability variables. In practice, duplicated variables are expressed by adding “colouring” operators to $\Theta$; their number is finite and depends only on the set of rules since multiplicities are fixed and finite for rules in the above format. Let $\hat{V}$ be the set of “coloured” variables from $V$ where the colouring is used to distinguish duplicated variables (cf. [4, \S 5.3]). Given a substitution $\nu$ from $\hat{V}$ to (finite) probability distributions over $T^\Sigma(X + Y)$, each $t_i$ is interpreted as the probability distribution:
\[
\hat{t}_i(t) = \begin{cases} 
\prod_{k=1}^{|\hat{V}\cap\nu(t_i)} \nu(\varphi_k)(t_k) & \text{if } t = t_i[\varphi_k/t_k] \text{ for } t_k \in T^\Sigma(X + Y) \\
0 & \text{otherwise}
\end{cases}
\]
and each target term $w_1 \cdot t_1 + \cdots + w_m \cdot t_m$ is interpreted as the convex combination of $\hat{t}_1, \ldots, \hat{t}_m$.

### 4.3 Weighted GSOS

In [22], Klin and Sassone proposed a GSOS format\(^*\) for Weighted LTSs that is parametric in the commutative monoid $\mathbb{W}$ and hence called $\mathbb{W}$-GSOS. The format subsumes many known formats for systems expressible as $WLTS$: for instance, Stochastic GSOS specifications are in the $\mathbb{R}_0^+$-GSOS format and GSOS for LTS are in the $\mathbb{B}$-GSOS format where $\mathbb{B} = (\{\mathsf{t}, \mathsf{f}\}, \lor, \mathsf{f})$.

**Definition 12 ([22 Def. 13]).** A $\mathbb{W}$-GSOS rule is an expression of the form:

\[
\begin{array}{l}
\begin{array}{l}
\{ x_i \xrightarrow{a_i} w_{ai} \}_{1 \leq i \leq n}, a \in A,
\{ x_{ik} \xrightarrow{b_{ik} u_{ik}} y_k \} \quad 1 \leq k \leq m
\end{array}
\end{array}
\]

\[
f(x_1, \ldots, x_n) \xrightarrow{c, (u_1, \ldots, u_m)} t
\]

where:

\begin{itemize}
  \item $f$ is an $n$-ary symbol from $\Sigma$;
  \item $X = \{x_i | 1 \leq i \leq n\}$, $Y = \{y_k | 1 \leq k \leq m\}$ and $\{w_{ai} | 1 \leq i \leq n, a \in A_i\}$ are pairwise distinct process and weight variables;
  \item $\{u_{ai} \in \mathbb{W} | 1 \leq i \leq n, a \in A_i\}$ are weight constants such that $w_{ai} \neq 0$ for $1 \leq k \leq m$;
  \item $\beta : W^m \rightarrow W$ is a multiaadditive function on $\mathbb{W}$;
  \item $a, b, c \in A$ are labels and $A_i \subseteq A$ for $1 \leq i \leq n$;
  \item $t$ is a $\Sigma$-term such that $Y \subseteq \text{var}(t) \subseteq X \cup Y$;
\end{itemize}

\(^*\)Weighted GSOS specifications are proved to yield GSOS distributive laws for Weighted LTSs but it is currently an open question whether the format is also complete.
A rule is triggered by a n-tuple \( \bar{C} \) of enabled labels s.t. \( A_i \subseteq C_i \) and by a family of weights \( \{v_{ai} \mid 1 \leq i \leq n, a \in A_i \} \) s.t. \( w_{ai} = v_{ai} \). A \( \mathfrak{M} \)-GSOS specification is a set of rules in the above format such that there are only finitely many rules for the same source symbol, conclusion label and trigger.

Each rule describes the weight of \( t \) in terms of weights assigned to each \( y_k \) (i.e. \( u_k \)) occurring in it; if two rules share the same symbol, label, trigger and target then their contribute for \( t \) is added.

To turn a \( \mathfrak{M} \)-GSOS specification into WF-GSOS ones, the first step is to make weight function explicit, by means of premises like \( x_i \xrightarrow{\varphi_i} \psi_i \) (since WLTS are functional ULTraS, i.e. \( m_i^t = 1 \)). Then, each premise \( x_i \xrightarrow{\varphi_i} w_{ai} \) is translated into \( \| \varphi_i \| = w_{ai} \). If \( \mathfrak{M} \) is zerosumfree (i.e., whenever \( a + b = 0 \) then \( a = b = 0 \)) then \( \mathfrak{M} \setminus \{0\} \) is a club and the translation of a \( \mathfrak{M} \)-GSOS into a WF-GSOS is straightforward. More generally, it suffices to combine rules sharing the same source, label and trigger into a single WF-GSOS rule with the same source, label and trigger. Its target is a suitable weight term containing the functions \( \beta \) and targets \( t \) of the original rules; every occurrence of variables \( y_k \) and \( u_k \) is replaced with the corresponding function variable (i.e. \( \varphi_i^{b_k} \)). In order to deal with multiple copies of the same weight variable, we wrap each occurrence in a different “colouring” operator, like in the case of Segala-GSOS.

5 A coalgebraic presentation of ULTraS and WF-GSOS

The main aim of this Section is to prove some important results about WF-GSOS specifications. We first provide a characterization of ULTraSs as coalgebras for a specific behavioural functor (Section 5.1), and their bisimulations as cocongruences. Then, leveraging this characterization in Section 5.2 we apply Turi and Plotkin bialgebraic theory \[4\], which allows us to define the categorical notion of WF-GSOS distributive law; these laws describe the interplay between syntax and behaviour in any GSOS presentation of ULTraS. We will prove that every WF-GSOS specification yields a WF-GSOS distributive law, i.e., the format is sound. As a consequence, we obtain that the bisimilarities induced by these specifications are always congruence relations. Finally, in Section 5.3 we prove that WF-GSOS specification are also complete: every abstract WF-GSOS distributive law can be described by means of a WF-GSOS specification.

5.1 ULTraSs as coalgebras

Since ULTraSs alternate non-deterministic steps with quantitative steps, the corresponding behavioural functor can be obtained by composing the usual functor \((\mathcal{F})^A : \text{Set} \to \text{Set}\) of non-deterministic labelled transition systems with the functors capturing the quantitative computational aspects.

Let us recall that for every set \( X \) we denoted by \( \mathcal{F}_{ULTRA} X = \{ \varphi : X \to W \mid \| \varphi \| < \omega \} \) the set of finitely supported weight functions over \( X \). This extends to an endofunctor \( \mathcal{F}_{ULTRA} : \text{Set} \to \text{Set} \) whose action on morphisms is given by, for \( f : X \to Y : \mathcal{F}_{ULTRA}(f)(\varphi) \triangleq \lambda y. \sum_{x \in f^{-1}(y)} \varphi(x) \). It is easy to check that identities and compositions are preserved.

Proposition 5. For any \( \mathfrak{M} \) and any \( A \), coalgebras for \((\mathcal{F}_A)^\mathfrak{M}\) are in one-to-one correspondence with \( A \)-labelled image-finite \( \mathfrak{M} \)-ULTraSs.

Proof. Any image-finite \( \mathfrak{M} \)-ULTraS \((X, A, \rightarrow)\) determines a coalgebra \((X, h)\) where, for any \( x \in X \) and \( a \in A \) : \( h(x)(a) \triangleq \{ \rho \mid x \xrightarrow{\rho} a \} \). Image-finiteness guarantees that these sets are finite and that their elements are finitely supported weight functions from \( X \) to the carrier of \( \mathfrak{M} \). Then, it is easy to check that the correspondence is bijective.

A similar result holds also for the bisimulation given in Definition 4. Categorically, a relation between \( X \) and \( Y \) is a (jointly monic) span \( X \leftarrow R \to Y \). In our case, this span has to be subject to some conditions, as shown next.

Proposition 6. Let \((X_1, A, \rightarrow_1)\) and \((X_2, A, \rightarrow_2)\) be two image-finite ULTraSs over \( \mathfrak{M} \); let \((X_1, h_1)\), \((X_2, h_2)\) be the corresponding coalgebras according Proposition 5. A relation between
\( X_1 \) and \( X_2 \) is a bisimulation iff there exists a coalgebra \((Y,g)\) and two coalgebra morphisms \( f_1 : (X_1,h_1) \to (Y,g) \) and \( f_2 : (X_2,h_2) \to (Y,g) \) such that \( f_1, f_2 \) are jointly epic and \( R \) is their pullback, i.e. the diagram below commutes.

\[
\begin{array}{ccc}
X_1 & \xrightarrow{f_1} & Y \\
\downarrow{h_1} & & \downarrow{f_2} \\
X_2 & \xrightarrow{f_2} & Y \\
\end{array}
\]

\( (\eta_{\mathcal{F}_{\text{wpp}}} X_i)^A \)

\( (\eta_{\mathcal{F}_{\text{wpp}}} f_i)^A \)

\( (\eta_{\mathcal{F}_{\text{wpp}}} Y)^A \)

\( (\eta_{\mathcal{F}_{\text{wpp}}} f_2)^A \)

\( (\eta_{\mathcal{F}_{\text{wpp}}} X_2)^A \)

Proof. See Appendix [C] page 21.

Intuitively, the system \((Y,g)\) “subsumes” both \((X_1,h_1)\) and \((X_2,h_2)\) via \( f_1, f_2 \); then, \( R \) relates the states which are mapped to the same behaviour in \( Y \) \((x_1, x_2) \in R \text{ if } f_1(x_1) = f_2(x_2)\)). This kind of behavioural equivalence is also called kernel bisimulation (or cocongruence).

**Coalgebraic bisimulation** In Concurrency theory also Aczel-Medler’s coalgebraic bisimulation [1] is widely used —arguably because its formulation corresponds precisely to the (strong) bisimulations of most calculi (CCS, \(\pi\)-calculus, etc.). In fact, it is known that kernel bisimulations and coalgebraic bisimulations coincide if the behavioural functor is weak pullback preserving (wp). This is the case for many behavioural functors, but not for \(\mathcal{F}_{\text{wpp}}\) in general [22]. Actually, the fact that \(\mathcal{F}_{\text{wpp}}\) (and \((\eta_{\mathcal{F}_{\text{wpp}}})^A\)) preserves weak pullbacks depends on the underlying monoid only:

**Lemma 7.** Coalgebraic bisimulation and behavioural equivalence on ULTraSs coincide if for any two vectors \((w_1)_{i=1\ldots n}, (v_1)_{i=1\ldots m}\) s.t. \(\sum_{i=1}^{n} w_i = \sum_{i=1}^{m} v_i = s\) there exists a \((u_{ij})_{i=1\ldots n,j=1\ldots m}\) s.t. \(\sum_{i=1}^{n} u_{ij} = v_j\) for each \(j = 1\ldots m\) and \(\sum_{j=1}^{m} u_{ij} = w_i\) for each \(i = 1\ldots n\), as in the matrix below.

\[
\begin{array}{cccc}
& w_1 & & \\
u_{1,1} & u_{1,2} & \cdots & u_{1,n}\\
u_{2,1} & u_{2,2} & \cdots & u_{2,n}\\
\vdots & \vdots & \ddots & \vdots \\
u_{m,1} & u_{m,2} & \cdots & u_{m,n}\\
\end{array}
\]

\[
\begin{array}{c}
v_1 \\
v_2 \\
\vdots \\
v_m
\end{array}
\]

\[
\begin{array}{c}
w_1 \\
w_2 \\
\vdots \\
w_m
\end{array}
\]

Proof. \((\eta)^A\) is weak wpp, and under the proposition hypothesis also \(\mathcal{F}_{\text{wpp}}\) is wpp, by a straightforward extension of [23 Th. 3.6] as noted in [19] Prop. 4. Therefore, \((\eta_{\mathcal{F}_{\text{wpp}}})^A\) is wpp, hence every behavioural equivalence is a coalgebraic bisimulation on \((\eta_{\mathcal{F}_{\text{wpp}}})^A\)-coalgebras. We conclude by Proposition [3].

This condition, called row-column property, can be easily verified and in fact holds for most monoids of interest. A simple counter example is \(\{(0,a,b,1),+,0\}\) where \(x + y = 1\) whenever \(x \neq 0 \neq y\) (cf. \(\overline{a} = \langle a,a \rangle\) and \(\overline{b} = \langle b,b \rangle\)).

### 5.2 WF-GSOS specifications are WF-GSOS distributive laws

In this subsection we put the WF-GSOS format within the bialgebraic framework [33]. As a consequence, we obtain that the bisimilarity induced by the ULTraS defined by this specification is a congruence. We refer the interested reader to [20] for an introduction to the general theory of abstract GSOS.

In particular, we prove that every WF-GSOS specification represents a distributive law of the signature over the ULTraS behavioural functor, i.e., a natural transformation of the form

\[
\lambda : \Sigma(\text{Id} \times (\eta_{\mathcal{F}_{\text{wpp}}})^A) \Rightarrow (\eta_{\mathcal{F}_{\text{wpp}}} T^\Sigma)^A
\]
where \( A \) is the set of labels, \( \mathcal{M} \) is the commutative monoid of weights, \( \Sigma = \coprod_{i \in I} H^{\mathcal{M}(i)} \) is the syntactic endofunctor induced by the process signature \( \Sigma \), and \( T^\Sigma \) is the free monad for \( \Sigma \). We will call natural transformations of this type WF-GSOS distributive laws.

Before stating the soundness theorem, we note that every natural transformation \( \lambda \) as above induces a \( (\mathcal{P}, F_{\mathcal{M}})^A \)-coalgebra structure over ground \( \Sigma \)-terms. Namely, this is the only function \( h_\lambda : T^\Sigma \emptyset \to (\mathcal{P}, F_{\mathcal{M}}(T^\Sigma \emptyset))^A \) such that:

\[
h_\lambda \circ a = (\mathcal{P}, F_{\mathcal{M}} T^\Sigma (a^\#))^A \circ \lambda_X \circ \Sigma(id, h_\lambda)
\]

(5)

where \( a^\# : T^\Sigma T^\Sigma \emptyset \to T^\Sigma \emptyset \) is the inductive extension of \( a \).

We can now provide the soundness result for WF-GSOS specifications with respect to WF-GSOS distributive laws, and between systems and coalgebras they induce over ground \( \Sigma \)-terms.

**Theorem 8** (Soundness). A specification \( (\mathcal{R}, [\cdot]) \) yields a natural transformation \( \lambda \) as in (5) such that \( h_\lambda \) and the ULTraS induced by \( (\mathcal{R}, [\cdot]) \) coincide.

**Proof.** For any set \( X \), define the function \( \lambda_X \) as the composite:

\[
\Sigma(X \times (\mathcal{P}, F_{\mathcal{M}} X)^A) \xrightarrow{[\mathcal{R}]_X} (\mathcal{P}, F_{\mathcal{M}}(X + F_{\mathcal{M}} X))^A \xrightarrow{(\mu \circ \mathcal{P} \circ \mu)^A} (\mathcal{P}, F_{\mathcal{M}} T^\Sigma X)^A
\]

where \( [\mathcal{R}]_X \) is defined as follows: for all \( \psi' \in T^\Sigma(X + F_{\mathcal{M}} X) \), \( \mathcal{X} \in \Sigma \), \( c \in A \), trigger \( \mathcal{A} = \langle a_1, \ldots, a_n \rangle \), \( \mathcal{w} = (w_1, \ldots, w_p) \), \( y_k \in X \) and \( \Phi_i(a) = \{ \phi^{a_i}_{ij} \in F_{\mathcal{M}} X \mid 1 \leq j \leq m^{a_i}_n \} \) for \( n = \mathit{nr}(\mathcal{X}) \) and \( i \in \{1, \ldots, n\} \), let

\[
\psi' \in [\mathcal{R}]_X(\mathcal{F}(\langle x'_1, \Phi_1 \rangle, \ldots, \langle x'_n, \Phi_n \rangle))
\]

if, and only if, there exists in \( \mathcal{R} \) a (possibly renamed) rule

\[
\left\{ x_i \xrightarrow{\psi} \phi_j^{a_j} \right\}_{1 \leq i \leq n, 1 \leq j \leq m^{a_i}_n} \quad \left\{ x_i \xrightarrow{\mathcal{A}} \right\}_{1 \leq i \leq n, 1 \leq k \leq p} \quad \left\{ w_k \right\}_{1 \leq k \leq p} \quad \left\{ \psi_{ik}^{a_k, \sigma_{ik}} \right\}_{1 \leq k \leq q} \quad \psi
\]

such that \( m^{a_i}_n \neq 0 \) if \( a \in A \) and there exists a substitution \( \sigma \) such that \( \psi' = \sigma[\psi], \sigma x_i = x'_i, \sigma y_k = y'_k, \sigma \phi^{a_i}_{ij} = \phi^{a_i}_{ij}, [\phi^{a_k}_{ik} w_k] = w_k \) and \( \phi^{a_k}_{ik} (\sigma y_k) \in \mathcal{C}_k \). Then, naturality can be proved separately for the two components: the former can be tackled as in \([33, Th. 1.1]\) and the latter readily follows from Definition 8.

Correspondence of \( T^\Sigma \emptyset, h_\lambda \) with the induced ULTraS follows by noting that the latter is given by structural recursion on \( \Sigma \)-terms by applying precisely the distributive law \( \lambda \) as given above (cf. \([33]\) and Definition 10).

Now, by general results from the bialgebraic framework, every behavioural equivalence on the coalgebra induced on ground process terms \( T^\Sigma \emptyset, h_\lambda \) is also a congruence on its carrier \( T^\Sigma \emptyset \) with respect to the process signature \( \Sigma \) (i.e. w.r.t. their \( \Sigma \)-algebra structure). In order to obtain this result we need the following (simple yet important) property.

**Proposition 9.** The category of \( (\mathcal{P}, F_{\mathcal{M}})^A \)-coalgebras has a final object.

**Proof.** By \([3]\) every finitary Set endofunctor admits a final coalgebra. By definition \( F_{\mathcal{M}} \) is finitary for every abelian monoid \( \mathcal{M} \). The thesis follows from \( \mathcal{P} \cong F_2 \) and from finitarity being preserved by functor composition.

**Corollary 10** (Congruence). Behavioural equivalence on the coalgebra over \( T^\Sigma \emptyset \) induced by \( (\mathcal{R}, [\cdot]) \) is a congruence with respect to the process signature \( \Sigma \).

**Proof.** The syntactic endofunctor \( \Sigma \) admits an initial algebra and, by Proposition 9 the behavioural endofunctor \( (\mathcal{P}, F_{\mathcal{M}})^A \) admits a final coalgebra. The same holds for their free monad and cofree copointed functor respectively. The specification \( (\mathcal{R}, [\cdot]) \) defines, by Theorem 8, a distributive law which uniquely extends to a distributive law distributing the free monad over the cofree copointed functor; then the thesis follows from \([33, Cor. 7.3]\).
5.3 WF-GSOS distributive laws are WF-GSOS specifications

In this subsection we give the important result that the WF-GSOS format is also complete with respect to distributive laws of the form \( \lambda \).

**Theorem 11** (Completeness). Every WF-GSOS distributive law \( \lambda \) arises from some WF-GSOS specification \( \langle R, \theta \rangle \).

The proof of this Theorem follows the methodology introduced by Bartels for proving adequacy of Bloom’s GSOS specification format \([4, \S 3.3.1]\). The (rather technical) proof will take the rest of this subsection, so for sake of conciseness we omit to recall some results which can be found in loc. cit.

The thesis follows by proving that, for every \( \lambda \), there exists an image-finite set of WF-SOS rules \( R \) (and suitable interpretations \( \theta \) and \( \xi \)) making the diagram in Figure 2 commute. The lower part of the diagram defines the interpretation \( \{\cdot\} \) out of \( \xi \) and \( \theta \) completing the WF-GSOS specification for \( \lambda \). The middle and right parts of the diagram trivially commute.

The upper part of the diagram commutes because of the following lemma which states that every WF-GSOS distributive law arises from an interpretation and a natural transformation having the same type of those defined by image-finite sets of WF-GSOS rules.

**Lemma 12.** Let \( \Sigma, \ A \) and \( \mathcal{M} \) be a signature, a set of labels and a commutative monoid respectively. Let \( \lambda \) be a WF-GSOS distributive law as in \([4]\). There exist \( \Theta \) and an interpretation \( \theta \) factorizing \( \lambda \) i.e. there exist \( \rho : \Sigma(\text{Id} \times (\bar{\mathcal{R}}, \mathcal{A})^A) \to (\bar{\mathcal{R}}, \mathcal{T}^\Theta(\text{Id} + \mathcal{F}_{\mathcal{M}}))^A \) s.t. \( \lambda = (\mu \circ \bar{\mathcal{R}} \theta)^A \circ \rho \).

*Proof (sketch).* In \textbf{Set} it is easy to encode finitely supported functions as terms. For instance let \( \Theta \) extend \( \Sigma \) with operators for describing collections and weight assignments (e.g. \( \langle \cdot \rangle \mapsto w \) where \( w \in \mathcal{M} \setminus \{0\} \)). Then, we can turn \( \lambda \) into \( \rho \) by simply encoding its codomain. Then \( \theta \) simply evaluates these terms back to weight functions everything else to the \( \emptyset \).

The left part of the diagram commutes by reducing \( \rho \) to simpler, but equivalent, families of natural transformations following Bartels’ methodology and eventually deriving a syntactical specification which is then shown to be equivalent to a image-finite set of WF-GSOS rules and an intermediate interpretation \( \xi \). The use of another weight signature \( \Xi \) besides \( \Theta \) gives us an extra degree of freedom and simplifies the proof. In particular, it allows us to encode natural transformations of type \( \mathcal{F}_{\mathcal{M}} \Rightarrow \bar{\mathcal{R}} \mathcal{F}_{\mathcal{M}} \) (yielded by the aforementioned reduction) in \( \xi \) and handle them downstream to the interpretation \( \{\cdot\} \). This expressiveness gain is one of the reasons for the introduction of non-determinism in Definition \( 8 \).

First, note that, by \([4, \text{Lem. A.1.1}]\), \( \rho \) as above is equivalent to:

\[
\bar{\rho} : \Sigma(\text{Id} \times (\bar{\mathcal{R}}, \mathcal{A})^A) \times A \Rightarrow \bar{\mathcal{R}} T^\Theta(\text{Id} + \mathcal{F}_{\mathcal{M}})
\]
which is equivalent to a family of natural transformations
\[ \alpha_{\tau,c} : (\text{Id} \times (\mathcal{P} F_{\Theta})^A)^N \Longrightarrow \mathcal{P} T^\Theta(\text{Id} + F_{\Theta}) \] indexed by \( f \in \Sigma \) and \( c \in A \) and where \( N = \{1, \ldots, \text{ar}(f)\} \). In fact, \( \Sigma \) is a polynomial functor and \( \text{Id} \times A \cong A \cdot \text{Id} \) is an \(|A|\)-fold coproduct.

By [4] Lem. A.1.7, each \( \alpha_{\tau,c} \) is equivalent to a natural transformation
\[ \alpha_{\tau,c} : (\mathcal{P} F_{\Theta})^{A \times N} \Longrightarrow \mathcal{P} T^\Theta(N + \text{Id} + F_{\Theta}) \] and, by the natural isomorphism
\[ (\mathcal{P} F_{\Theta})^{A \times N} \cong (\mathcal{P} F_{\Theta}^+)^{A \times N} \cong \prod_{E \subseteq A \times N} (\mathcal{P} F_{\Theta}^+)^E \]
each \( \alpha_{\tau,c} \) is equivalent to a family of natural transformations
\[ \beta_{\tau,c,E} : (\mathcal{P} F_{\Theta}^+)^E \Longrightarrow \mathcal{P} T^\Theta(N + \text{Id} + F_{\Theta}) \] where the added index corresponds to the vector of sets of labels \( (E_1, \ldots, E_{\text{ar}(f)}) \) composing the trigger of a WF-GSOS rule. By the natural isomorphism
\[ \mathcal{P} F_{\Theta}^+ \cong \mathcal{P} F_{\Theta}^+ \prod_{v \in V} \mathcal{F}^\psi_{\Theta} \cong \prod_{v \in \mathcal{P} F_{\Theta}^+} \prod_{v \in V} \mathcal{P} F_{\Theta}^+ \mathcal{F}^\psi_{\Theta} \]
\[ \text{where } \mathcal{F}^\psi_{\Theta} X \triangleq \{ \varphi \in \mathcal{F} X \mid \| \varphi \| = v \} \], each \( \beta_{\tau,c,E} \) is equivalent to a family of natural transformations
\[ \gamma_{\tau,c,E,w} : \prod_{c \in E} \prod_{v \in w(c)} \mathcal{P} F_{\Theta}^+ \Longrightarrow \mathcal{P} T^\Theta(N + \text{Id} + F_{\Theta}) \] where \( w : E \rightarrow \mathcal{P} F_{\Theta}^+ \). Since total weight premises associate pairs from \( E \) to weights, maps like \( w \) can be seen as families of triggering weights.

By [4] Lem. A.1.3 and by the natural isomorphism
\[ T^\Theta \cong \prod_{|\psi|_*} \text{Id}^{[\psi]_*} \]
where \(|\psi|_*\) denotes the number of occurrences of \(* \in \{1 \) in the \( \Theta \)-term \( \psi \) (cf. [4] Lem. A.1.5]) each \( \gamma_{\tau,c,E,w} \) corresponds to a family of natural transformations
\[ \delta_{\tau,c,E,w,|\psi|_*} : \prod_{c \in E} \prod_{v \in w(c)} \mathcal{P} F_{\Theta}^+ \Longrightarrow \mathcal{P} T^\Theta(\text{Id} + F_{\Theta})^{[\psi]_*} \] where the added index \( |\psi|_* \) ranges over some subset of \( T^\Theta(1 + N) \) (cf. target terms of WF-GSOS rules).

Then, following [4] §3.3.1, Cor. A.2.8 it is easy to check that each \( \delta_{\tau,c,E,w,|\psi|_*} \) describes a non-empty, finite set of derivation rules of the form of
\[ \phi_{j,vj} \in \pi_{\psi,v} (\Phi_{c_j}) \quad y_i \in \epsilon_{j,vj} (\phi_{j,vj}) \\
(z_1, \ldots, z_{|\psi|_*}) \in \delta_{\tau,c,E,w,|\psi|_*} (\Phi_{c_j} v \in \text{Id}) \]
\[ \text{where } p, q \in \mathbb{N}, e_j \in E, 1 \leq j \leq p, 1 \leq i \leq q, v_j \in w(e_j), \text{ each } z_k \in \{ y_i \mid 1 \leq i \leq q \} \text{ for } 1 \leq k \leq |\psi|_* \text{ and each } \epsilon_{j,vj} \text{ is a natural transformation:} \]
\[ \epsilon_{j,vj} : \mathcal{F}^{\psi}_{\Theta} \Longrightarrow \mathcal{P} F_{\Theta}^+ (\text{Id} + F_{\Theta}) \].

Natural transformations of this type can be easily encoded in the term \( \psi \) by suitable extensions of \( \Theta \) and therefore each \( \delta_{\tau,c,E,w,|\psi|_*} \) can be shown to be equivalent to a \( \delta \)-specification i.e. a non-empty finite set of derivation rules as above except for each \( z_k \) being a term wrapping \( \phi_{j,vj} \) with the symbol denoting \( \epsilon_{j,vj} \). These terms are then evaluated by the interpretation \( \xi^3 \) as expected.

This proof points out the trade-off that has to be made in presence of specifications with interpretation such as WF-GSOS or MGSOS [2]. In fact, clubs were not mentioned in the above reduction of \( \rho \) since \( \epsilon \) were handled by the interpretation \( \xi \). However, the following result shows that clubs (hence, premises like \( \psi \in \mathcal{E} \supseteq y \)) characterize natural transformations of type \( \mathcal{F}^\psi_{\Theta} \Rightarrow \mathcal{P} F_{\Theta}^+ \).
Lemma 13. For any natural transformation \( \nu : \mathcal{F}^w_{\mathcal{M}} \Rightarrow \mathcal{P} \) there exists a club \( \mathfrak{C}_\nu \) characterizing it: \( x \in \nu \chi(\varphi) \iff \phi(x) \in \mathfrak{C}_\nu \).

Proof (sketch). Intuitively, natural transformations of this type are “selecting a finite subset from each weight function domain” and it is easy to check that elements can be only singled out by their weight. Likewise, finiteness and naturality prevent the selection of anything outside function supports. Then, the problem readily translates into finding the finest topology on the weight monoid that “plays well” with \( \mathcal{F}^w_{\mathcal{M}} \) i.e. such that monoidal addition, seen as a continuous map from the product topology, preserves opens (i.e. any admissible selection). Clubs form this topology since, by definition, these are the only substructures isolated w.r.t. \( \mathcal{F}^w_{\mathcal{M}} \)-action. Hence selections made by \( \nu \) are completely characterized by a single club \( \mathfrak{C}_\nu \).

Finally, we have to translate the set of rules we got so far into the WF-GSOS format; we do it by reversing the chain that led us from \( \rho \) to \( \delta \) and \( \delta \)-specification. By Lemma 13 every \( \delta \)-specification is equivalent to a \( \gamma \)-specification

\[
\left\{ \begin{array}{l}
\phi_j \in \Phi_{\nu_j} \\
\| \phi_j \| = v_j \\
\psi(z_1, \ldots, z_{|\psi|}) \in \gamma_{x_1, x_2, \ldots, x_{|\psi|}}(\Phi_{\nu_j})
\end{array} \right\}
\]

where \( \phi_j[\zeta_j] \) is a term build with the \( \Xi \)-operator denoting the natural transformation \( \zeta_j : \mathcal{F}^w_{\mathcal{M}} \Rightarrow \mathcal{P} \mathcal{F}^w_{\mathcal{M}} \) and \( \xi^d \) acts as \( \xi^d \) on these terms, as the identity on those generated from \( \Theta \) (distributing the powerset as expected) and maps everything else to \( \emptyset \). A \( \gamma \)-specification defines a natural transformation as in \( \mathcal{F}^w_{\mathcal{M}} \) and every family of \( \gamma \)-specifications characterizing a natural transformation as in \( \mathcal{F}^w_{\mathcal{M}} \) is equivalent to a \( \beta \)-specification i.e. a set of derivation rules

\[
\left\{ \begin{array}{l}
\phi_j \in \Phi_{\nu_j} \\
\| \phi_j \| = v_j \\
\psi(z_1, \ldots, z_{|\psi|}) \in \beta_{x_1, x_2, \ldots, x_{|\psi|}}(\Phi_{\nu_j})
\end{array} \right\}
\]

finite up to vectors of total weights \( \vec{v} = (v_1, \ldots, v_p) \). Since \( E \subseteq A \times N \), every family of \( \beta \)-specifications describing a natural transformation as in \( \mathcal{F}^w_{\mathcal{M}} \) is equivalent to a set

\[
\left\{ \begin{array}{l}
\Phi_{\nu_{b_n}} = \emptyset \\
\phi_j \in \Phi_{\nu_{b_n}} \\
\| \phi_j \| = v_j \\
\psi(z_1, \ldots, z_{|\psi|}) \in \alpha_{x_1, x_2, \ldots, x_{|\psi|}}(\Phi_{\nu_{b_n}})
\end{array} \right\}
\]

containing finitely many rules for every \( E \) and \( \vec{v} \). This set corresponds to an \( \alpha \)-specification i.e. an image-finite set like the following:

\[
\left\{ \begin{array}{l}
\Phi_{\alpha_{b_n}}(\Phi_{\nu_{b_n}}) = \emptyset \\
\phi_j \in \Phi_{\alpha_{b_n}} \\
\| \phi_j \| = v_j \\
\psi(z_1, \ldots, z_{|\psi|}) \in \alpha_{x_1, x_2, \ldots, x_{|\psi|}}(\Phi_{\alpha_{b_n}}(\Phi_{\nu_{b_n}}))
\end{array} \right\}
\]

Finally, every family of \( \alpha \)-specifications equivalent to a natural transformation as \( \rho \) corresponds to an image-finite set of WF-GSOS rules and an interpretation. Therefore we conclude that for any \( \rho \) there exist \( \mathcal{R} \) and \( \xi \) as in Figure 2.

6 Conclusions and future work

In this paper we have presented WF-GSOS, a GSOS-style format for specifying non-deterministic systems with quantitative aspects. A WF-GSOS specification is composed by a set of rules for the derivation of judgements of the form \( P \xrightarrow{\omega} \psi \), where \( \psi \) is a term of a specific signature, and an interpretation for these terms as weight functions. We have shown that a specification in this format defines an ULTraS, and it is expressive enough to subsume other more specific formats such as WGSOS and Segala-GSOS. WF-GSOS induces naturally a notion of (strong) bisimulation, which we have compared with \( \mathcal{M} \)-bisimulation used in ULTraS. We have also provided a general categorical presentation of ULTraSs as coalgebras of a precise class of functors, parametric on
the underlying weight structure. This presentation allows us to define categorically the notion of abstract GSOS for ULTraS, i.e., natural transformations of a precise type. We have proved that WF-GSOS specification format is adequate (i.e., sound and complete) with respect to this notion. Taking advantage of Turi-Plotkin’s bialgebraic framework, we have proved that the bisimulation induced by a WF-GSOS is always a congruence; hence our specifications can be used for compositional and modular reasoning in quantitative settings (e.g., for ensuring performance properties). Moreover, the format is at least as expressive as every GSOS specification format for systems subsumed by ULTraS.

In [10] Gebler et al. proposed a ntµfν/ntµxν rule format for describing Segala Systems and such that the induced bisimulation is always a congruence. It would be interesting to investigate the possibility to extend their work to the wider range of behaviours covered by ULTraSs since, in general, GSOS and tree-rule formats have different expressive powers.

Although in this paper we have taken ULTraS systems as a reference, WF-GSOS can be interpreted in other meta-models, such as FuTS [24]. Like ULTraS, FuTS have state-to-function transitions, but admit several distinct domains for weight functions and hence can be read as “composing in parallel” distinct behaviours. The results in this paper readily extend to FuTS since these systems can be seen as coalgebras for functors with a specific shape: \((\bigotimes_{W \in \mathcal{W}} F_W)^A\) for \(\mathcal{W}\) being the set of admitted weight domains. In this context it is easy to formulate compositionality results also for the framework for stochastic calculi proposed in [12]. A coalgebraic understanding of FuTS is presented in [24] but covers only the deterministic case (i.e. \((\bigotimes_{W \in \mathcal{W}} F_W)^A\)), while ours is non-deterministic.

For sake of simplicity, we have characterized ULTraS systems using the functor \(F_W\). However, the results and definition presented here can be further generalized by taking generic behavioural functors in place of \(F_W\), thus considering systems that are coalgebras for functors of the form of \((\bigotimes B)^A\). This would affect mostly the evaluation \([\cdot]\), while only minor changes to the rule format may be required in order to capture interactions between \(\bigotimes\) and \(B\) (like e.g. the total weight premises). This fact suggests to investigate systems with stratified (or “staged”) behaviours via “stratified” specifications. We can develop general results at the abstract level of bialgebraic structural operational semantics, aiming to provide some modularity to the format. This line of research can be seen as complementary to Mosses’ Modular SOS [29] and recent developments towards a GSOS equivalent [27] (which still are more “syntax bound” since the behavioural functor is not very changed by these compositions).

The categorical characterization of ULTraS systems paves the way for further interesting lines of research. One is to develop Hennessy-Milner style modal logics for quantitative systems at the generality level of the ULTraS framework. In fact, Klin has shown in [18] that HML and CCS are connected by a (contravariant) adjunction. A promising direction is to follow this connection taking advantage of the bialgebraic presentation of ULTraSs provided in this paper. Another is to explore the implications of the recent developments in the coalgebraic understanding of silent actions [2][14][25] in the context of this work.

Acknowledgements We thank Rocco De Nicola, Daniel Gebler and the QAPL’14 participants for useful discussions on the conference version of this paper.

References

[1] P. Aczel and N. Mendler. A final coalgebra theorem. In D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigne, editors, Pro. CTCS, volume 389 of Lecture Notes in Computer Science, pages 357–365, Berlin, 1989. Springer-Verlag.

[2] G. Bacci and M. Miculan. Structural operational semantics for continuous state probabilistic processes. In D. Pattinson and L. Schröder, editors, CMCS, volume 7399 of Lecture Notes in Computer Science, pages 71–89. Springer, 2012.

[3] M. Barr. Terminal coalgebras in well-founded set theory. Theoretical Computer Science, 114(2):299–315, 1993.

[4] F. Bartels. On Generalised Coinduction and Probabilistic Specification Formats: Distributive Laws in Coalgebraic Modelling. PhD thesis, CWI, Amsterdam, 2004.

[5] M. Bernardo, R. De Nicola, and M. Loreti. A uniform framework for modeling nondeterministic, probabilistic, stochastic, or mixed processes and their behavioral equivalences. Information and Computation, 225:29–82, 2013.
[6] 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.

[7] T. Brengos. On coalgebras with internal moves. In M. M. Bonsangue, editor, *Coalgebraic Methods in Computer Science*, Lecture Notes in Computer Science, pages 75–97. Springer, 2014.

[8] M. Calder, S. Gilmore, and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways. In G. Plotkin, editor, *Proc. CMSB*, pages 204–215, 2005.

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

[10] M. David Lee, D. Gebler, and P. R. D’Argenio. Tree rules in probabilistic transition system specifications with negative and quantitative premises. In B. Luttik and M. A. Reniers, editors, *EXPRESS/SOS*, volume 89 of *EPTCS*, pages 115–130, 2012.

[11] R. De Nicola, D. Latella, M. Loreti, and M. Massink. Rate-based transition systems for stochastic process calculi. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part II*, ICALP ’09, pages 435–446. Springer-Verlag, 2009.

[12] R. De Nicola, D. Latella, M. Loreti, and M. Massink. A uniform definition of stochastic process calculi. *ACM Computing Surveys*, 46(1):5, 2013.

[13] M. Droste, W. Kuich, and H. Vogler. *Handbook of weighted automata*. Springer, 2009.

[14] S. Goncharov and D. Pattinson. Coalgebraic weak bisimulation from recursive equations over monads. In J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, editors, *Proc. ICALP*, volume 8573 of *Lecture Notes in Computer Science*, pages 196–207. Springer, 2014.

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

[16] J. Hillston. A compositional approach to performance modelling. Cambridge University Press, 1996.

[17] J. Hillston. Process algebras for quantitative analysis. In *LICS*, pages 239–248. IEEE Computer Society, 2005.

[18] B. Klin. Bialgebraic methods and modal logic in structural operational semantics. *Information and Computation*, 207(2):237–257, 2009.

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

[20] B. Klin. Bialgebras for structural operational semantics: An introduction. *Theoretical Computer Science*, 412(38):5043–5069, 2011.

[21] B. Klin and V. Sassone. Structural operational semantics for stochastic process calculi. In R. M. Amadio, editor, *FoSSaCS*, volume 4962 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2008.

[22] B. Klin and V. Sassone. Structural operational semantics for stochastic and weighted transition systems. *Information and Computation*, 2013.

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

[24] D. Latella, M. Massink, and E. P. de Vink. Bisimulation of labeled state-to-function transition systems of stochastic process languages. In U. Golas and T. Soboll, editors, *Proc. ACCAT*, volume 93 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–43, 2012.

[25] M. Miculan and M. Peressotti. Weak bisimulations for labelled transition systems weighted over semirings. *CoRR*, abs/1310.4106, 2013.

[26] M. Miculan and M. Peressotti. Gsos for non-deterministic processes with quantitative aspects. In N. Bertrand and L. Bortolussi, editors, *Proc. QAPL*, volume 154 of *Electronic Proceedings in Theoretical Computer Science*, pages 17–33. Open Publishing Association, 2014.

[27] S. Milch, L. S. Moss, and D. Schwencke. Abstract gsos rules and a modular treatment of recursive definitions. *Logical Methods in Computer Science*, 9(3), 2013.

[28] L. S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1):277–317, 1999.

[29] P. D. Mosses. Foundations of modular sos. In M. Kutylowski, L. Pacholski, and T. Wierzbicki, editors, *MFCS*, volume 1672 of *Lecture Notes in Computer Science*, pages 70–80. Springer, 1999.

[30] C. Priami. Stochastic pi-calculus. *Comput. J.*, 38(7):578–589, 1995.

[31] R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. *Nord. J. Comput.*, 2(2):250–273, 1995.

[32] C. M. N. Tofts. A synchronous calculus of relative frequency. In J. C. M. Baeten and J. W. Klop, editors, *CONCUR*, volume 458 of *Lecture Notes in Computer Science*, pages 467–480. Springer, 1990.

[33] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Proc. LICS*, pages 280–291. IEEE Computer Society Press, 1997.
A Weighted transition Systems

Weighted labelled transition systems (e.g. [13, 22, 32]) are LTS whose transition are assigned a weight drawn from a commutative monoid \( W = (W, 0, +) \). Henceforth we will write \( W \)-LTS for \( W \)-Weighted LTS or in general WLTS if no specific monoid is intended.

Definition 13 ([22, Def. 2]). Given a commutative monoid \( W = (W, 0, +) \), a \( W \)-weighted LTS is a triple \((X, A, \rho)\) where:

- \( X \) is a set of states (processes);
- \( A \) is a set of labels (actions);
- \( \rho : X \times A \times X \rightarrow W \) is a weight function, mapping each triple of \( X \times A \times X \) to a weight.

\((X, A, \rho)\) is said to be image-finite iff for each \( x \in X \) and \( a \in A \), the set \( \{ y \in X \mid \rho(x, a, y) \neq 0 \} \) is finite.

It is well-known that, for suitable choices of \( W \) and constraints, WLTS subsume several kind of systems such as:

- \((\{\mathsf{tt}, \mathsf{ff}\}, \vee, \mathsf{ff})\) for non-deterministic systems;
- \((\mathbb{R}^+_0, +, 0)\) for rated systems [11, 21] (e.g. CTMCs);
- \((\mathbb{R}^+_0, +, 0)\) and \( \forall x \in X, a \in A \sum_{y \in X} \rho(x, a, y) \in \{0, 1\} \) for generative (or fully) probabilistic systems;
- \((\mathbb{R}^+_0, +, 0)\) and \( \forall x \sum_{a \in A, y \in X} P(x, a, y) \in \{0, 1\} \) for reactive probabilistic systems;
- \((\mathbb{R}^+_0, \max, 0)\) for “capabilities” (weights denotes the capabilities of a process and similar capabilities add up to a stronger one);
- etc.

Moreover, Klin defined in [22] a notion for WLTS (based on cocongruences) which uniformly instantiates to known bisimulations for systems expressible in the WLTS framework.

Definition 14 ([22, Def. 4]). Given a \( W \)-LTS \((X, A, \rho)\), a \( W \)-bisimulation is an equivalence relation \( R \) on \( X \) such that for each pair \((x, x')\) of elements of \( X \), \((x, x') \in R\) implies that for each label \( a \in A \) and each equivalence class \( C \) of \( R \):

\[ \sum_{y \in C} \rho(x, a, y) = \sum_{y \in C} \rho(x', a, y). \]

WLTS are precisely functional ULTraS and, as stated in Proposition 1, every weighted bisimulation for a WLTS is a bisimulation for the corresponding functional ULTraS and vice versa.

Proof of Proposition 1. Trivially, there is a 1-1 correspondence between \( W \)-LTS and functional \( W \)-ULTraS. Then, Proposition 1 readily follows by observing that, for any given \( W \)-LTS/ULTraS \((X, A, \rightarrow)\), Definition 4 degenerates in Definition 13 because for any \( x \stackrel{a}{\rightarrow} \varphi \) there is exactly one \( y \stackrel{a}{\rightarrow} \psi \).

B Segala Systems

In their general format, Segala systems [31] are state machines (originally introduced as automata) whose transitions can be pictured as being made of two steps belonging to two different behavioural aspects: the first sub-step is non-deterministic and the second one is probabilistic. The following definitions are taken from [31] with minor notational differences and by restricting to finite probability distributions (whereas the original definition is given to discrete at most countable probability spaces) for conciseness and uniformity with the restriction to image-finite systems made in the paper.

Definition 15. A Segala system is a triple \((X, A, \rightarrow)\) where:
X is a set of states (processes);
A is a set of labels (actions);
→ ⊆ X × D(A × X) a transition relation between states and discrete probability spaces over pairs of labels and states.

Before stating what is a strong bisimulation for these systems, we need to recall the notion of R-equivalence between probability distributions [31, Def. 13] which, intuitively, says that two distributions are R-equivalent if they assign the same probability to each class in the equivalence relation R.

**Definition 16.** Let R be an equivalence relation over a set X. Two probability distributions ϕ, ψ over it are said to be R-equivalent (written ϕ ≡R ψ) iff for each [x] ∈ [P]/R there exists [y] ∈ [P′]/R such that xRy (vice versa for [y]), and for each [x] ∈ [P]/R and [y] ∈ [P]/R xRy implies that

\[ \sum_{z \in [P]} \phi(z) = \sum_{z \in [P']} \psi(z) \cdot \mathbb{P}(z). \]

Strong bisimulation [31, Def. 14] is defined in terms of simple-labelled steps i.e. transitions of the form of x → ⊳ P which intuitively means to see labels as inputs ((DX)A) instead of outputs (D(A × X)).

**Definition 17** (31, Def. 14). Given a Segala system (X, A, →), a bisimulation is an equivalence relation R on X such that for each pair (x, x') of elements of X, (x, x') ∈ R implies that for each label a ∈ A if x → ⊳ P then x' → ⊳ P' such that P ′ ≡R P and symmetrically for x'.

**Proof of Proposition 2.** Clearly DX ⊆ Fg X and hence Segala systems are constrained ULTraS. Then, Proposition 2 readily follows by observing that the two notions coincide on the non-deterministic part and that R-equivalence means that the two distributions assign the same probability to each equivalence class of R i.e. the same total weight: P(C) = \( \sum_{c \in C} P(c) = P'(C) \).

C Omitted proofs

**Proof of Proposition 2.** The function M is well-defined because \( M(x, a, C) = \perp \) whenever x → ⊳ ρ or, for each x → ⊳ ρ, ρ(C) = 0, and \( M(x, a, C_1) = M(y, a, C_1) \) and \( M(x, a, C_2) = M(y, a, C_2) \) implies \( M(x, a, C_1 \cup C_2) = M(y, a, C_1 \cup C_2) \) by definition of ≡R.

By Definition 2 whenever x → ⊳ ϕ then y → ⊳ ψ s.t. ϕ(C) = ψ(C) for each C ∈ XR i.e. ϕ ≡R ψ and the symmetric case for y. Therefore (x, y) ∈ R implies that \( \Phi_{x, a} \triangleq \{ [\phi]_{=R} \mid x \rightarrow \phi \} \) and \( \Phi_{y, a} \triangleq \{ [\psi]_{=R} \mid y \rightarrow \psi \} \) are equal for each a ∈ A. We can safely add ⊥ to both \( \Phi_{x, a} \) and \( \Phi_{y, a} \) since, whenever both x and y terminate, they are either both stuck or both terminal. In fact, equality and inequality are preserved while adding ⊥ since \( \Phi_{x, a} = \emptyset \implies \bot \notin \Phi_{y, a} \) (and vice versa) by hypothesis. For each C ∈ XR (x, y ∈ X and a ∈ A) let \( \Psi_{x, a, C} \triangleq \{ [\rho]_{=R} \mid \rho(C) = 0 \} \) \( \cup \{ \bot \} \). Clearly \( \Phi_{x, a} \cup \Psi_{x, a, C} \) and \( \Phi_{y, a} \cup \Psi_{y, a, C} \) are equal. Complementarily, if (x, y) ∉ R then there exists some ψ ∈ \( \Phi_{x, a} \) s.t. for no ϕ ∈ \( \Phi_{y, a} \) ϕ ≡R ψ or vice versa; w.l.o.g. assume the former. Hence there exists C ∈ XR such that ϕ(C) ≠ ψ(C) whence \( \Psi_{x, a, C} \neq \Psi_{y, a, C} \). Finally, we conclude by \( M(x, a, C) = \Psi_{x, a, C} \) for each x ∈ X, a ∈ A and C ∈ XR.

**Proof of Proposition 2.** Let (X, A, →X), (Y, A, →Y), (X, α) and (Y, β) be two ULTraS over a given commutative monoid M and their corresponding coalgebras (cf. Proposition 5). Recall that a function f : X → Y is a also coalgebra morphism f : α ↠ β iff, for each x ∈ X, and a ∈ A:

\[ f(x) \rightarrow \alpha \psi \iff x \rightarrow \alpha \varphi \land \psi = \varphi[f] \]

where \( \varphi[f] \) denotes the action of f on \( \varphi \) (i.e. the function \( \lambda_y : Y. \sum_{x \in f^{-1}(y)} \varphi(x) \)) and function equality is defined point-wise as usual.

Firstly, we prove that if R is a kernel relation of some jointly epic cospan of coalgebra morphism from α and β then it is a bisimulation in the sense of Definition 2. Let the aforementioned cospan
be \((X, \alpha) \xrightarrow{f} (Z, \gamma) \xrightarrow{g} (Y, \beta)\), \((Z, A, \rightarrow_Z)\) the ULTraS for \(\gamma\) and assume \(x\) and \(y\) such that \(f(x) = g(y)\). Since \(f\) is coalgebra homomorphism, \(f(x) = z\) implies:

\[
x \xrightarrow{a} x \varphi \iff z \xrightarrow{\alpha} z \rho = \varphi[f] = \lambda c : Z. \sum_{x \in f^{-1}(c)} \varphi(x).
\]

Likewise \(g(y) = z\) implies:

\[
y \xrightarrow{a} y \psi \iff z \xrightarrow{\alpha} z \rho = \psi[g] = \lambda c : Z. \sum_{y \in g^{-1}(c)} \psi(y).
\]

Therefore \(f(x) = g(y)\) implies:

\[
x \xrightarrow{a} x \varphi \implies y \xrightarrow{a} y \psi \land \forall \ C \in Z. \sum_{x \in f^{-1}(c)} \varphi(x) = \sum_{y \in g^{-1}(c)} \psi(y)
\]

We conclude by noting that if \(R\) is the kernel of \(f, g\) there is a bijective correspondence between its equivalence classes and elements in \(Z\) since every class is in the image of \(f\) or \(g\) by the epic sink assumption.

For the converse, given a bisimulation \(R\) for \((X, A, \rightarrow_X)\) \((Y, A, \rightarrow_Y)\) let \(Z\) be the set of the equivalence classes in \(R\) and consider the ULTraS \((Z, A, \rightarrow_Z)\) defined as follows:

\[
C \xrightarrow{a} \lambda D : Z. \sum_{x_\in D} \varphi(x') \iff x \xrightarrow{a} X \varphi \land \forall \ C \in Z. \sum_{x \in f^{-1}(c)} \varphi(x) = \sum_{y \in g^{-1}(c)} \psi(y)
\]

The two statements are redundant since \(x, y \in C \iff xRy\) and hence iff for every \(x \xrightarrow{a} X \varphi\) there is \(y \xrightarrow{a} Y \psi\) s.t. \(\varphi \equiv_R \psi\) and vice versa. Finally, class membership defines a jointly epic coalgebra cospan from the coalgebras associated to \((X, A, \rightarrow_X)\) and \((Y, A, \rightarrow_Y)\) to the one associated to \((Z, A, \rightarrow_Z)\) by simply mapping each \(x \in X\) and each \(y \in Y\) to its class. \(\square\)