Structural Operational Semantics for Weighted Transition Systems

Bartek Klin*

Warsaw University, University of Cambridge

Abstract. Weighted transition systems are defined, parametrized by a commutative monoid of weights. These systems are further understood as coalgebras for functors of a specific form. A general rule format for the SOS specification of weighted systems is obtained via the coalgebraic approach of Turi and Plotkin. Previously known formats for labelled transition systems (GSOS) and stochastic systems (SGSOS) appear as special cases.

1 Introduction

In its simplest and most well-studied form [1], Structural Operational Semantics (SOS) is a framework for inductive definition of transition systems labeled with some entities that have no internal structure, such as channel names in the process algebra CCS [2]. A rich theory of such SOS specifications have been developed including, among many other results, rule formats such as GSOS [3]. These guarantee good properties of bisimilarity, the canonical notion of equivalence on labeled transition systems.

However, already from the original paper that introduced SOS [4, 5] it is apparent that for all but the simplest applications, one needs to consider systems where transitions carry more information than simple, unstructured labels. To model various aspects of computation, one endows transitions with information on fresh and bound names [6], transition probabilities [7] or durations [8], memory states, environments and so on. Crucially, the additional structure put on transitions influences the corresponding notion of process equivalence, so the simple theory of SOS and bisimilarity cannot be directly applied to these extended specification frameworks.

An important advantage of studying the structure of labels and transitions is that in a description of a rule format, or in a concrete specification of a system, one can specify that structure only partially and leave inessential details for further stages of specification. This allows for modularity in operational specifications, where different parts of a language can be specified independently, and extending an already specified language requires only an instantiation of the partially specified structure of transitions and labels, rather than its redefinition. This idea was promoted by Peter Mosses in his Modular Structural Operational

* This work was supported by EPSRC grant EP/F042337/1.

Semantics (MSOS, [9, 10]), where transition labels are thought of as arrows in categories, while the structure of these categories models various relevant aspects of computation and can be partially abstracted away in concrete specifications. MSOS achieves excellent modularity of SOS specifications that involve environments of various sorts, memory stores and communication channels. As a theory of process equivalence for MSOS specifications is missing, it is less clear how to meaningfully apply the framework for some aspects of computation such as probability or time.

A more radical abstraction step is the approach of *universal coalgebra* [11]. There, the only assumption on the structure of transitions and labels is that it corresponds to an endofunctor on some category (most often, the category **Set** of sets and functions). In particular, the notions of a transition or label are abstracted away in the coalgebraic framework. This approach provides a useful theory of process equivalence based on coalgebraic bisimilarity. Moreover, in [12] a general coalgebra-based theory of SOS specifications was developed. However, that *abstract GSOS* theory is rather detached from the usual transition-and-label presentations of SOS specifications, and it typically takes a considerable effort to understand its instances for particular classes of systems (see [13–15]). Also, the theory of modularity for abstract GSOS specification does not exist, apart from some initial results of [16, 17].

This paper attempts to be a modest step in between the two approaches: the coalgebraic framework is applied to a class of systems where one can meaningfully speak of transitions and labels exhibiting a certain structure. More specifically, we study *weighted transition systems*, where every labeled transition is associated with a weight drawn from a commutative monoid $\mathfrak{W}$. The monoid structure determines the way in which weights of alternative transitions combine. A uniform coalgebraic treatment of weighted transition systems is provided, including a concrete definition of $\mathfrak{W}$-weighted bisimulation, and a general well-behaved rule format called $\mathfrak{W}$-GSOS, parametrized by the underlying monoid of weights, is defined. Weighted transition systems generalize both ordinary LTS and stochastic transition systems of [15], as well as other potentially useful kinds of systems.

After some algebraic preliminaries in Section 2, in Section 3 weighted transition systems are defined on a concrete level, without any use of coalgebraic techniques. The abstract coalgebraic approach to processes and SOS is recalled in Section 4, and weighted transition systems are presented as coalgebras in Section 5. Section 6 contains the main technical result of this paper: a definition of the $\mathfrak{W}$-GSOS format. In Section 7, it is shown how $\mathfrak{W}$-GSOS instantiates to the previously known formats GSOS [3] and SGSOS [15], for specific choices of the monoid $\mathfrak{W}$.

In Sections 4–6, the notions of functor and natural transformation are used. For these and other basic categorical notions, consult the first chapters or any handbook of category theory; [18] is the classical reference.

2 Preliminaries

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

Given a commutative monoid $\mathfrak{W}$, a function $\beta : W^k \to W$ is *multiadditive* if for each $i \in \{1, \ldots, k\}$:

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

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

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

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

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

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

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

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

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

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

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

3 Weighted transition systems

Consider any commutative monoid $\mathfrak{W} = (W, 0, +)$. Elements of W will be called *weights* and denoted $v, w, \ldots$.

Definition 1. A $\mathfrak{W}$*-weighted labelled transition system* ($\mathfrak{W}$-LTS in short) is a triple (X, A, ρ) where

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

To support intuitions based on classical labelled transition systems (LTSs), we shall write $\rho(x \xrightarrow{a} y)$ for $\rho(x, a, y)$, and to say that $\rho(x \xrightarrow{a} y) = w$ we shall write $x \xrightarrow{a,w} y$.

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

Definition 2. A $\mathfrak{W}$-LTSs (X, A, ρ) is *image finite* if for each $x \in X$ and $a \in A$, the set of $y \in X$ such that $\rho(x \xrightarrow{a} y) \neq 0$ is finite.

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

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

Definition 3. Given a $\mathfrak{W}$-LTS (X, A, ρ), a $\mathfrak{W}$*-bisimulation* is an equivalence relation R on X such that for each $x, x' \in X$, xRx' implies that for each $a \in A$ and each equivalence class C of R:

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

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

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

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

Example 1. Consider the monoid of logical values $\mathbb{2} = \{\mathtt{ff}, \mathtt{tt}\}$, with logical disjunction as $+$ and $\mathtt{ff}$ as the zero element. $\mathbb{2}$-LTSs are exactly ordinary (image-finite) LTSs, and $\mathbb{2}$-bisimulations are classical bisimulations (more precisely, bisimulation equivalences).

Example 2. For $\mathbb{R}_0^+$ the monoid of nonnegative real numbers under addition, $\mathbb{R}_0^+$-LTSs are exactly *rated transition systems* used in [15] to model stochastic systems, and $\mathbb{R}_0^+$-bisimilarity is stochastic bisimilarity [15], called strong equivalence in [19].

Example 3. The set $\mathbb{R}^{+\infty}$ of positive real numbers augmented with positive infinity ∞, forms a commutative monoid with the minimum operation as addition and ∞ as the zero element. $\mathbb{R}^{+\infty}$-LTSs themselves are almost the same as $\mathbb{R}_0^+$-LTSs of Example 2, with the only difference in the capability of making transitions with weight 0 or ∞. However, the different monoid structures lead to different notions of weighted bisimilarity and, as a result, to very different intuitions about the roles of weights in these systems. Indeed, while in Example 2 rates model the *capability* of a process to make a transition, with the idea that two similar capabilities add up to a stronger one, here weights might correspond to the *cost* of transitions, with the intuition that out of several similar possibilities, a process will always choose that of the lowest cost.

4 Abstract GSOS

In [12] (see [20] for a more elementary introduction), Turi and Plotkin proposed an abstract way of understanding well-behaved structural operational semantics for systems of various kinds. There, behaviour of transition systems is modeled by coalgebras, and their syntax by algebras. For example, image-finite LTSs labelled with elements of A can be understood as functions $h : X \to (\mathcal{P}_\omega X)^A$, where $\mathcal{P}_\omega$ is the finite powerset construction. More generally, for any covariant functor B on the category **Set** of sets and functions, a *B-coalgebra* is a set X endowed with a function $h : X \to BX$. A B-coalgebra *morphism* from a $h : X \to BX$ to $g : Y \to BY$ is a function $f : X \to Y$ such that $g \circ f = Bf \circ h$. The kernel relations of coalgebra morphisms are called *cocongruences* on their domains. Processes $x, y \in X$ are *observationally equivalent* with respect to $h : X \to BX$ if they are related by a cocongruence on h. For more information about the coalgebraic approach to process theory, see [11].

More traditionally, process syntax is modeled via algebras for endofunctors. Every algebraic signature Σ corresponds to a functor $\Sigma X = \coprod_{\mathtt{f} \in \Sigma} X^{ar(\mathtt{f})}$ on **Set**, in the sense that a model for the signature is exactly an *algebra* for the functor, i.e., a set X and a function $g : \Sigma X \to X$. The set of Σ-terms with variables from a set X is denoted $T_\Sigma X$. In particular, $T_\Sigma \emptyset$ is the set of closed terms over Σ; it admits an obvious algebra structure $a : \Sigma T_\Sigma \emptyset \to T_\Sigma \emptyset$ for the functor corresponding to the signature. This is the *initial* Σ-algebra. The construction T_Σ is also a functor, called the *free monad* over Σ.

In [12], Turi and Plotkin observed (a full proof was provided later by Bartels [13]), that operational LTS specifications in the well-known image finite GSOS format [3] are in an essentially one-to-one correspondence with *distributive laws*, i.e., natural transformations of the type

$$\lambda : \Sigma(\mathrm{Id} \times B) \Longrightarrow BT_\Sigma \tag{1}$$

where $B = (\mathcal{P}_\omega -)^A$ is the behaviour functor used for modeling LTSs, Σ is the functor corresponding to the given signature, and T_Σ is the free monad over Σ. Moreover, any λ as above gives rise to a B-coalgebra structure h_λ on $T_\Sigma 0$, defined by a "structural recursion theorem" (see [12] for details) as the only function $h_\lambda : T_\Sigma 0 \to BT_\Sigma 0$ such that:

$$h_\lambda \circ a = Ba^\sharp \circ \lambda_X \circ \Sigma\langle \mathrm{id}, h_\lambda \rangle, \tag{2}$$

where $a^\sharp : T_\Sigma T_\Sigma \emptyset \to T_\Sigma \emptyset$ is the inductive extension of a.

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

Theorem 1 ([12], Cor. 7.5). Assume B has a final coalgebra. For any λ as in (1), observational equivalence on $h_\lambda : T_\Sigma \emptyset \to BT_\Sigma \emptyset$ is a congruence on $T_\Sigma \emptyset$.

Based on this result, the search for congruence formats for weighted transition systems should begin from understanding them as coalgebras.

5 Weighted transition systems as coalgebras

As before, we start with a commutative monoid $\mathfrak{W} = (W, 0, +)$. For any set X, a function $\phi : X \to W$ is *finitely supported* if $\phi(x) \neq 0$ for only finitely many $x \in X$. Let $\mathcal{F}_\mathfrak{W} X$ denote the set of all finitely supported functions from X to W. This extends to an endofunctor $\mathcal{F}_\mathfrak{W}$ on the category **Set** of sets and functions, with the action on functions defined by:

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

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

Proposition 1. For any $\mathfrak{W}$, and any set A, coalgebras for the functor $(\mathcal{F}_\mathfrak{W} -)^A$ are in one-to-one correspondence with $\mathfrak{W}$-LTSs labelled with A.

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

This coalgebraic understanding is justified by the corresponding treatment of weighted bisimilarity:

Proposition 2. For any $\mathfrak{W}$-LTS (X, A, ρ), an equivalence relation on X is a $\mathfrak{W}$-bisimulation if and only if it is the kernel relation of a coalgebra morphism from the corresponding $\mathcal{F}_\mathfrak{W}$-coalgebra. As a corollary, two processes are $\mathfrak{W}$-bisimilar if and only if they are observationally equivalent.

Proof. See Appendix A. □

Remark 1. See Appendix B for an explanation as to why observational equivalence is used here instead of the approach of coalgebraic bisimilarity, based on spans of coalgebra morphisms.

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

Proposition 3. The functor $(\mathcal{F}_{\mathfrak{W}}-)^A$ admits a final coalgebra.

Proof. As proved in [21], it is enough to show that $\mathcal{F}_{\mathfrak{W}}$ is finitary, i.e. that for any set X and any $x \in \mathcal{F}_{\mathfrak{W}}X$ there is a finite subset $Y \subseteq X$ such that x arises as an element of $\mathcal{F}_{\mathfrak{W}}Y$. But this easily follows from the assumption that $\mathcal{F}_{\mathfrak{W}}X$ only contains finitely supported functions. □

6 Weighted GSOS

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

$$\lambda : \Sigma(\mathrm{Id} \times (\mathcal{F}_{\mathfrak{W}}-)^A) \Longrightarrow (\mathcal{F}_{\mathfrak{W}}T_\Sigma-)^A \tag{3}$$

where Σ is the process syntax signature endofunctor and T_Σ is the free monad over Σ. Moreover, for any such syntactic entity, weighted bisimilarity is a congruence for the WTS obtained by from the corresponding λ by the inductive definition (2).

For $\mathfrak{W} = 2$ (see Example 1), image-finite GSOS specifications [3] define (3), as noticed in [12] and proved in [13]. Moreover, every natural transformation of type (3) arises from a GSOS specification. A similar full characterisation result, based on earlier developments of [13], was proved in [15] for $\mathfrak{W} = \mathbb{R}_0^+$, where a format called SGSOS (Stochastic GSOS) was proposed as a way to specify rated transition systems (see Example 2) well-behaved with respect to stochastic bisimilarity.

We shall now show a rule format, which we call $\mathfrak{W}$-*GSOS*, parametrized by a commutative monoid $\mathfrak{W}$, and see how specifications that conform to the format give rise to natural transformations of type (3). Both GSOS and SGSOS shall appear as special cases of this general format, as will be seen in Section 7.

Note, however, that we do not claim that every natural transformation of type (3) can be presented by a $\mathfrak{W}$-GSOS specification for every monoid $\mathfrak{W}$. In this sense, our results are weaker than those of [13] and [15], where full characterisation results for specific monoids were proved; nevertheless, we still provide a general congruence format for transition systems weighted by an arbitrary monoid.

In the following, fix an arbitrary commutative monoid $\mathfrak{W} = (W, 0, +)$.

Definition 4 ($\mathfrak{W}$-GSOS rule). A $\mathfrak{W}$-*GSOS rule* for a signature Σ and a set A of labels is an expression of the form:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a}\!\!\!\!\blacktriangleleft\, w_{a,i}\right\}_{a\in D_i, 1\le i\le n} \quad \left\langle \mathtt{x}_{i_j} \xrightarrow{b_j,u_j}\!\!\!\!\blacktriangleright\, \mathtt{y}_j \right\rangle_{1\le j\le k}}{\mathtt{f}(\mathtt{x}_1,\ldots,\mathtt{x}_n) \xrightarrow{c,\beta(u_1,\ldots,u_k)}\!\!\!\!\blacktriangleright\, \mathtt{t}} \tag{4}$$

where

- $\mathtt{f} \in \Sigma$ and $ar(\mathtt{f}) = n$, with $n, k \in \mathbb{N}$, and $\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,n\}$;
- $\mathtt{x}_i$ and $\mathtt{y}_j$ are all distinct variables taken from a fixed countably infinite set Ξ, and no other variables appear in $\mathtt{t} \in T_\Sigma \Xi$;
  moreover, all variables $\mathtt{y}_j$ appear in $\mathtt{t}$;
- $D_i \subseteq A$;
- $w_{a,i} \in W$;
- $b_1, b_2, \ldots, b_k, c \in A$;
- $u_1, \ldots, u_k$ are pairwise distinct *weight variables* taken from a fixed countably infinite set Υ;
- $\beta : W^k \to W$ is a multiadditive function on $\mathfrak{W}$.

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

We now provide some terminology, notation and intuitions to aid the understanding of $\mathfrak{W}$-GSOS rules. The expression under the horizontal line in a rule is called the *conclusion*. The left side of the conclusion is called the *source* of a rule, and the right side is the *target*. Expressions above the horizontal line are *premises*. Each rule has premises of two kinds: *total weight* premises, depicted with $\longrightarrow\!\!\!\!\blacktriangleleft$ arrows, and *transition* premises, where $\longrightarrow\!\!\!\!\blacktriangleright$ arrows are used.

Total weight premises form a set, i.e., their order in a rule is irrelevant. The set of total weight premises in a rule defines a partial function from $\{1,\ldots,n\} \times A$ to W, and the sets D_i describe its domain of definition for each $i = 1..n$. Intuitively, a total weight premise $\mathtt{x} \xrightarrow{a}\!\!\!\!\blacktriangleleft\, w$ is satisfied for a process x in a $\mathfrak{W}$-LTS, if the sum of weights of all a-labelled transitions from x equals w.

Transition premises in a rule form a sequence, i.e., their order is relevant. Note that the u_j in transition premises are not fixed weights (elements of W), but variables. The meaning of a premise $\mathtt{x} \xrightarrow{b,u}\!\!\!\!\blacktriangleright\, \mathtt{y}$ applied to a source process x and a target process y in a $\mathfrak{W}$-LTS is to assign the transition weight $\rho(x \xrightarrow{b} y)$ to the variable u, used then as an argument in the function β mentioned in the rule conclusion. This process is formally described in Definition 6 below.

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

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

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

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

Definition 5 ($\mathfrak{W}$-GSOS specification). Given a signature Σ and a set A of labels, a $\mathfrak{W}$-*GSOS specification* Λ is a set of $\mathfrak{W}$-GSOS rules such that only finitely many rules share the same operator in the source ($\mathtt{f}$), the same label in the conclusion (c), and the same partial function from $\{1,\ldots,n\} \times A$ to W arising from their sets of total weight premises.

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

Definition 6 (induced $\mathfrak{W}$-LTS). The $\mathfrak{W}$-LTS induced by a $\mathfrak{W}$-GSOS specification Λ over a signature Σ and a set of labels A, has the set $T_\Sigma\emptyset$ of closed Σ-terms as states and A as the set of labels. The weight function $\rho : T_\Sigma\emptyset \times A \times T_\Sigma\emptyset \to W$ is defined by structural induction on the first argument. To this end, consider a process $s = \mathtt{f}(s_1,\ldots,s_n) \in T_\Sigma\emptyset$ and assume that all $\rho(s_i \xrightarrow{a} t)$ have been determined for all $a \in A$ and $t \in T_\Sigma\emptyset$. For a fixed label $c \in C$ and a process $t \in T_\Sigma\emptyset$, define $\rho(s \xrightarrow{c} t)$ as follows.

We shall say that a rule R as in (4) *fits* $s \xrightarrow{c} t$ if all of the following hold:

(i) the operator in the source of R is $\mathtt{f}$,
(ii) the label in the conclusion of R is c,
(iii) for each total weight premise $\mathtt{x}_i \xrightarrow{a} \lhd w$ in R, there is $\rho(s_i, a, T_\Sigma\emptyset) = w$ (i.e. total weight premises are satisfied),
(iv) there exists a substitution $\sigma : \Xi_R \to T_\Sigma\emptyset$ such that:
- $\sigma\mathtt{x}_i = s_i$ for $i = 1,\ldots,n$, and
- $\sigma[\![\mathtt{t}]\!] = t$.

It is important to note that if R fits $s \xrightarrow{c} t$, then the fitting substitution σ is unique. Indeed, the action of σ on the $\mathtt{x}_i$ is explicitly defined by $\sigma\mathtt{x}_i = s_i$, and the action on the $\mathtt{y}_j$ is determined by the condition $\sigma[\![\mathtt{t}]\!] = t$. This is easily proved by structural induction on $\mathtt{t}$, using the assumption that all variables $\mathtt{y}_j$ are present in $\mathtt{t}$.

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

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

We then define $\rho(s \xrightarrow{c} t)$ as the sum, taken in $\mathfrak{W}$, of contributions of all rules in Λ that fit $s \xrightarrow{c} t$. The sum exists thanks to the finiteness condition in Definition 5.

Theorem 2. Every $\mathfrak{W}$-GSOS specification Λ gives rise to a natural transformation λ as in (3). Moreover, the coalgebra induced from λ according to (2), coincides with the $\mathfrak{W}$-LTS induced from Λ according to Definition 6.

Proof. See Appendix C.

Corollary 1. For any $\mathfrak{W}$-GSOS specification Λ, $\mathfrak{W}$-bisimilarity is a congruence on the $\mathfrak{W}$-LTS induced by Λ.

Proof. Use Theorems 1 and 2 with Propositions 2 and 3.

7 Examples

7.1 GSOS as 2-GSOS

To relate $\mathfrak{W}$-GSOS to a more familiar format, we shall now see what $\mathfrak{W}$-GSOS specifications look like for $\mathfrak{W} = 2$ (see Example 1).

First, there are only two kinds of total weight premises to consider: ones of the form $\mathtt{x} \xrightarrow{a}\!\!\!\!\triangleleft\, \mathtt{tt}$ that require some a-transitions from a process corresponding to $\mathtt{x}$ to exist, and ones of the form $\mathtt{x} \xrightarrow{a}\!\!\!\!\triangleleft\, \mathtt{ff}$, that forbid such transitions. One can rewrite the former as $\mathtt{x} \xrightarrow{a}\!\!\!\!\triangleright$, and the latter as $\mathtt{x} \not\xrightarrow{a}\!\!\!\!\triangleright$.

Next, it is easy to see that for any $k \in \mathbb{N}$, there are only two multiadditive functions $\beta : \{\mathtt{tt}, \mathtt{ff}\}^k \to \{\mathtt{tt}, \mathtt{ff}\}$ on the monoid 2. Indeed, by multiadditivity axioms, β is fully determined by its value on the all-$\mathtt{tt}$ vector. Assigning $\beta(\mathtt{tt}, \ldots, \mathtt{tt}) = \mathtt{tt}$ or $\beta(\mathtt{tt}, \ldots, \mathtt{tt}) = \mathtt{ff}$ one obtains respectively the k-ary conjunction or the constantly $\mathtt{ff}$ function as β. It is easy to check that both are multiadditive. However, a rule with the constantly $\mathtt{ff}$ function as β cannot make a nonzero contribution to any transition in the induced 2-LTS, therefore any 2-GSOS specification does not change its meaning if all such rules are removed from it. One can therefore safely restrict attention to rules with logical conjunction as β; this means that β can be left implicit in the description of each rule. Moreover, since conjunction is commutative, the order of transition premises in 2-GSOS rules is irrelevant.

The above observations let one write 2-GSOS rules in the form:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a}\!\!\!\!\triangleright\right\}_{a\in E_i, 1\le i\le n} \quad \left\{\mathtt{x}_i \not\xrightarrow{a}\!\!\!\!\triangleright\right\}_{a\in B_i, 1\le i\le n} \quad \left\{\mathtt{x}_i \xrightarrow{b_{ij}}\!\!\!\!\triangleright\, \mathtt{y}_{ij}\right\}_{1\le i\le n, 1\le j\le k_i}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \xrightarrow{c}\!\!\!\!\triangleright\, \mathtt{t}} \tag{5}$$

where

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

The induction process described in Definition 6 specializes to the following procedure. For a process $s = \mathtt{f}(s_1, \ldots, s_n) \in T_\Sigma$, assume that all outgoing transitions from the s_i have been determined. For a fixed label $c \in C$ and a process $t \in T_\Sigma \emptyset$, determine whether the transition $s \xrightarrow{c} t$ is present, as follows.

A rule R as in (5) *fits* $s \xrightarrow{c} t$ if all of the following hold:

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

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

It is clear that both the format (5) and the associated induction procedure are almost exactly those for the well-known GSOS format [3, 1]; the only difference in rule presentation is than in GSOS, premises $\mathtt{x} \xrightarrow{a}\triangleright$ are equipped with dummy target variables, and consequently the condition that all target variables of transition premises are present in $\mathtt{t}$, is dropped. It is not difficult to see that this makes no semantic difference in this case.

7.2 SGSOS rules as $\mathbb{R}_0^+$-GSOS rules

It is even easier to see that SGSOS specifications, defined in [15] as a way to specify Markovian transition systems, conform to the $\mathbb{R}_0^+$-GSOS format, for $\mathbb{R}_0^+$ the monoid of nonnegative real numbers under addition (see Example 2). An SGSOS rule is an expression of the form:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a@w_{ai}}\triangleright\right\}_{a\in D_i, 1\le i\le n} \quad \left\{\mathtt{x}_{i_j} \xrightarrow{b_j}\triangleright\, \mathtt{y}_j\right\}_{1\le j\le k}}{\mathtt{f}(\mathtt{x}_1, \ldots, \mathtt{x}_n) \xrightarrow{c,w}\triangleright\, \mathtt{t}}, \tag{6}$$

subject to conditions similar to that of $\mathfrak{M}$-GSOS (4); the only differences are the following:

(i) total weight premises are denoted $\mathtt{x} \xrightarrow{a@w}\triangleright$ in SGSOS rather than $\mathtt{x} \xrightarrow{a}\triangleleft\, w$ in $\mathbb{R}_0^+$-GSOS,
(ii) transition premises in SGSOS form a set rather than a sequence, i.e., their order is disregarded,
(iii) in the SGSOS rule conclusion, a non-zero weight $w \in \mathbb{R}^+$ is used rather than a multiadditive function β,
(iv) in SGSOS it is required that $b_{i_j} \in D_{i_j}$ for each $j = 1, \ldots, k$; in other words, each transition premise has a corresponding total weight premise, and moreover the corresponding total weight w_{b_j,i_j} is required to be non-zero.

The induction procedure of a stochastic transition system from an SGSOS specification is similar to that given in Definition 6. The notion of a fitting rule is exactly the same, but the contribution of a rule is defined a bit differently:

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

Any SGSOS rule, written down as in (6), can be encoded as a $\mathbb{R}_0^+$-GSOS rule:

$$\frac{\left\{\mathtt{x}_i \xrightarrow{a} \!\!\blacktriangleleft\, w_{a,i}\right\}_{a\in D_i, 1\le i\le n} \quad \left\langle \mathtt{x}_{i_j} \xrightarrow{b_j} \!\!\blacktriangleright\, \mathtt{y}_j \right\rangle_{1\le j\le k}}{\mathtt{f}(\mathtt{x}_1,\ldots,\mathtt{x}_n) \xrightarrow{c,\beta} \!\!\blacktriangleright\, \mathtt{t}} \tag{7}$$

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

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

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

7.3 $\mathbb{R}^{+\infty}$-GSOS

The appearance of $\mathfrak{W}$-GSOS rules does not depend on the monoid structure of $\mathfrak{W}$ apart from the choice of functions β, so $\mathbb{R}^{+\infty}$-GSOS and $\mathbb{R}_0^+$-GSOS specifications look almost the same. Note that for the monoid $\mathbb{R}^{+\infty}$ (see Example 3), where the minimum operation is taken as addition, a function β is multiadditive if and only if it is monotonic and preserves ∞, i.e. $\beta(w_1,\ldots,w_n) = \infty$ whenever some $w_i = \infty$. As a result, $\mathbb{R}^{+\infty}$-GSOS rules look as in (4), with the requirement that all $w_{a,i} \in \mathbb{R}^{+\infty}$ and β is a monotonic, ∞-preserving function.

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

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

can be defined by rules:

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

where in the first rule w represents the function constant at w, and in the other two rules the identity function is taken for β. By Definition 6 applied to $\mathfrak{W} = \mathbb{R}^{+\infty}$, contributions of different rules to single transitions are combined

using the minimum operation. As a result, for example, the process $(a.2).\mathtt{nil} + (a,3).\mathtt{nil}$ is $\mathbb{R}^{+\infty}$-bisimilar to $(a.2).\mathtt{nil}$, which corresponds to the intuition that nondeterministic processes always choose the lowest possible cost of transition.

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

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

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

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

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

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

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

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

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

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

for each $w \leq v \in \mathbb{R}^{+\infty}$. The resulting set of rules is uncountable, but it satisfies the finiteness condition of Definition 5. The operator defined by these rules preserves all a-labeled transitions if the minimal weight of an a-labeled outgoing transition is not bigger than the minimal weight of a b-labeled one, and vice versa.

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

8 Conclusions and future work

Several research directions are left open here. First of all, some interesting kinds of transition systems do not exactly fit in our framework, although they seem

quite close to it. For example, reactive probabilistic transition systems [7, 22] are almost like $\mathbb{R}_0^+$-LTSs of Example 2, except for the requirement that for each process P and label a, weights of all a-labelled transitions from P add up to 1.

This motivates the study of *constrained* weighted transition systems, i.e., coalgebras for functors $\mathcal{F}_{\mathfrak{W}}^V$ (where $V \subseteq W$ is the *constraint*), defined on sets by:

$$\mathcal{F}_{\mathfrak{W}}^V X = \Big\{ \phi \in \mathcal{F}_{\mathfrak{W}} X \mid \sum_{x \in X} \phi(x) \in V \Big\}$$

and as $\mathcal{F}_{\mathfrak{W}}$ on functions. For example, the probability distribution functor used in coalgebraic modeling of probabilistic systems is naturally isomorphic to $\mathcal{F}_{\mathbb{R}_0^+}^{\{0,1\}}$, and the subprobability distribution functor to $\mathcal{F}_{\mathbb{R}_0^+}^{[0,1]}$. Also various versions of deterministic systems can be modeled as coalgebras for suitable constrained weighted functors. However, a characterization of abstract GSOS natural transformations in terms of rule formats remains to be provided for bounded functors. We conjecture that the formats can be obtained by subjecting their unconstrained counterparts to additional constraint conditions on (collections of) multiadditive functions β used in rule conclusions.

Furthermore, contrary to previous developments on GSOS [13] and SGSOS [15] formats, we do not claim that $\mathfrak{W}$-GSOS fully characterizes abstract GSOS for $\mathfrak{W}$-LTSs, i.e. that every natural transformation as in (3) arises from a $\mathfrak{W}$-GSOS specification. We conjecture that this is not the case in general, and a characterization of those monoids $\mathfrak{W}$ for which the full characterization does hold is currently missing.

Another promising topic of future work is the incorporation of commutative monoid morphisms as a means of inducing morphisms between weighted GSOS specifications. The general goal here is a modular framework for the specification of weighted systems, inspired by MSOS [9, 10].

Last but not least, modal logics for reasoning about weighted systems should be developed along the lines of coalgebraic modal logic.

References

1. Aceto, L., Fokkink, W.J., Verhoef, C.: Structural operational semantics. In Bergstra, J.A., Ponse, A., Smolka, S., eds.: Handbook of Process Algebra. Elsevier (2002) 197–292
2. Milner, R.: Communication and Concurrency. Prentice Hall (1988)
3. Bloom, B., Istrail, S., Meyer, A.: Bisimulation can't be traced. Journal of the ACM **42** (1995) 232–268
4. Plotkin, G.D.: A structural approach to operational semantics. DAIMI Report FN-19, Computer Science Department, Aarhus University (1981)
5. Plotkin, G.D.: A structural approach to operational semantics. Journal of Logic and Algebraic Programming **60-61** (2004) 17–139
6. Sangiorgi, D., Walker, D.: The π-Calculus: a Theory of Mobile Processes. Cambridge University Press (2003)

7. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Information and Computation **94** (1991) 1–28
8. Moller, F., Tofts, C.: A temporal calculus of communicating systems. In: Proc. CONCUR'90. Volume 458 of LNCS. (1990) 401–415
9. Mosses, P.D.: Foundations of Modular SOS. In: Proc. MFCS'99. Volume 1672 of LNCS. (1999) 70–80
10. Mosses, P.D.: Modular structural operational semantics. Journal of Logic and Algebraic Programming **60-61** (2004) 195–228
11. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. Theoretical Computer Science **249** (2000) 3–80
12. Turi, D., Plotkin, G.D.: Towards a mathematical operational semantics. In: Proc. LICS'97. IEEE Computer Society Press (1997) 280–291
13. Bartels, F.: On Generalised Coinduction and Probabilistic Specification Formats. PhD dissertation, CWI, Amsterdam (2004)
14. Kick, M.: Rule formats for timed processes. In: Proc. CMCIM'02. Volume 68 of ENTCS., Elsevier (2002) 12–31
15. Klin, B., Sassone, V.: Structural operational semantic for stochastic systems. In: Proc. FOSSACS'08. Volume 4962 of LNCS. (2008) 428–442
16. Kick, M., Power, J., Simpson, A.: Coalgebraic semantics for timed processes. Information and Computation **204** (2006) 588–609
17. Lenisa, M., Power, J., Watanabe, H.: Category theory for operational semantics. Theoretical Computer Science **327**(1-2) (2004) 135–154
18. Mac Lane, S.: Categories for the Working Mathematician. Second edn. Springer (1998)
19. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press (1996)
20. Klin, B.: Bialgebraic methods and modal logic in structural operational semantics. Information and Computation **207** (2009) 237–257
21. Barr, M.: Terminal coalgebras in well-founded set theory. Theoretical Computer Science **114** (1993) 299–315
22. de Vink, E.P., Rutten, J.J.M.M.: Bisimulation for probabilistic transition systems: A coalgebraic approach. Theoretical Computer Science **221**(1-2) (1999) 271–293
23. Moss, L.: Coalgebraic logic. Annals of Pure and Applied Logic **96** (1999) 177–317

A Proof of Proposition 2.

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

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

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

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

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

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

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

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

this is well-defined since, by Definition 3, $\rho(x, a, C') = \rho(y, a, C')$ for any $x, y \in C$. Furthermore, the quotient map $[-]_R : X \to X/R$ is a coalgebra morphism, since $\overleftarrow{[-]_R}(C) = C$. □

B $\mathcal{F}_{\mathfrak{W}}$ and weak pullback preservation

When coalgebras for a functor B are considered, it is often assumed that B preserves weak pullbacks (see [11] for more details). Several useful results follow from this assumption. In particular, the notion of observational equivalence adopted in this paper coincides with another canonical notion, that of *coalgebraic bisimulation*, defined abstractly by means of spans of coalgebra morphisms. Also, the abstract GSOS machinery works for coalgebraic bisimulation only if the behaviour functor involved preserves weak pullbacks.

It turns out that our behaviour functors do not preserve weak pullbacks in general, therefore we choose to adopt the notion of observational equivalence, encouraged by Proposition 2.

We shall now see a characterization of those functors $\mathcal{F}_{\mathfrak{W}}$ that do preserve weak pullbacks, in terms of the underlying monoids.

Inspired by Theorem 3.6 in [23], we say that a commutative monoid $\mathfrak{W}$ has the *row-column property* if for every two vectors $(w_i)_{i=1..n}$, $(v_j)_{j=1..m}$ of elements of W such that $\sum_{i=1}^{n} w_i = \sum_{j=1}^{m} v_j$, there exists a rectangular matrix $(u_{ij})_{i=1..n, j=1..m}$ of elements of W such that $\sum_{j=1}^{m} u_{ij} = w_i$ for each $i = 1..n$ and $\sum_{i=1}^{n} u_{ij} = v_j$ for each $j = 1..m$, as illustrated below:

$$\begin{array}{cccc|c} u_{11} & u_{12} & \cdots & u_{1m} & w_1 \\ u_{21} & u_{22} & \cdots & u_{2m} & w_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nm} & w_n \\ \hline v_1 & v_2 & \cdots & v_m & \sum . \end{array}$$

Proposition 4. $\mathcal{F}_{\mathfrak{W}}$ preserves weak pullbacks if and only if $\mathfrak{W}$ has the row-column property.

Proof. The reasoning in Example 3.5 in [23] works here without any essential change. □

All monoids used in examples throughout this paper have the row-column property. For a simple example of one that does not have it, consider the four-element monoid $\{0, a, b, 1\}$, with 0 as the zero element and with addition defined

by:

$+$	0	a	b	1
0	0	a	b	1
a	a	1	1	1
b	b	1	1	1
1	1	1	1	1

and check that the property fails for $n = m = 2$ and $w_1 = w_2 = a$, $v_1 = v_2 = b$.

C Proof of Theorem 2

First we shall see how a single $\mathfrak{W}$-GSOS rule defines a natural transformation λ as in (3). For a $\mathfrak{W}$-GSOS rule R, and for any set X, consider an arbitrary $s = \mathtt{f}(x_1, \delta_1, \ldots, x_n, \delta_n) \in \Sigma(X \times (\mathcal{F}_{\mathfrak{W}} X)^A)$. To define $\lambda_X(s) \in (\mathcal{F}_{\mathfrak{W}} T_\Sigma X)^A$, pick an arbitrary $c \in A$ and $t \in T_\Sigma X$ and define $\lambda_X(s)(c)(t) \in W$ as follows.

Say that R fits s, c, t if:

(i) the operator in the source of R is $\mathtt{f}$,
(ii) the label in the conclusion of R is c,
(iii) for each total weight premise $\mathtt{x}_i \xrightarrow{a}\!\!\!\!\!\triangleleft\, w$ in R, there is $\sum_{y \in X} \delta_i(a)(y) = w$,
(iv) there exists a substitution $\sigma : \Xi_R \to X$ such that:
- $\sigma \mathtt{x}_i = x_i$ for $i = 1, \ldots, n$, and
- $\sigma[\mathtt{t}] = t$.

Then define:

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

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

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

The left side of this equation is:

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

and the right side:

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

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

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

the second equality makes use of the multiadditivity of β.

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

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

Recall that θ above, if it exists, is unique. Now if θ exists, then tuples $y_1, \ldots, y_k \in X$ such that $gy_j = \theta\mathtt{y}_j$ are in bijective correspondence with substitutions $\sigma : \Xi_R \to X$ such that $\sigma\mathtt{x}_i = x_i$ and $g[\sigma[\mathtt{t}]] = t$. Moreover, the existence of such σ implies that an appropriate θ exists (take $\theta = g\circ\sigma$). As a result, we can rewrite:

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

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

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

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

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

where λ^R arises from every single rule R as described above. Thanks to the finiteness condition in Definition 5, for each s and c the sum contains only finitely many non-zero summands, therefore the sum is well-defined and $\lambda_X(s)(c)$ is finitely supported. Naturality of λ follows easily.

It remains to be seen that the $\mathfrak{W}$-LTS induced from a $\mathfrak{W}$-GSOS specification coincides with the $(\mathcal{F}_{\mathfrak{W}}-)^A$-coalgebra arising from the corresponding λ according to (2). To this end, it is enough to show that the induced LTS, seen as a coalgebra, makes (2) commute. A straightforward way to prove this is to notice that the induction step in Definition 6 corresponds exactly to the definition of λ, along the correspondence between $\mathfrak{W}$-LTSs and $(\mathcal{F}_{\mathfrak{W}})^A$-coalgebras. □