

Distributive Laws for Monotone Specifications*

Jurriaan Rot

Radboud University, Nijmegen

jrot@cs.ru.nl

Turi and Plotkin introduced an elegant approach to structural operational semantics based on universal coalgebra, parametric in the type of syntax and the type of behaviour. Their framework includes abstract GSOS, a categorical generalisation of the classical GSOS rule format, as well as its categorical dual, coGSOS. Both formats are well behaved, in the sense that each specification has a unique model on which behavioural equivalence is a congruence. Unfortunately, the combination of the two formats does not feature these desirable properties. We show that *monotone* specifications—that disallow negative premises—do induce a canonical distributive law of a monad over a comonad, and therefore a unique, compositional interpretation.

1 Introduction

Structural operational semantics (SOS) is an expressive and popular framework for defining the operational semantics of programming languages and calculi. There is a wide variety of specification formats that syntactically restrict the full power of SOS, but guarantee certain desirable properties to hold [1]. A famous example is the so-called GSOS format [5]. Any GSOS specification induces a unique interpretation which is compositional with respect to (strong) bisimilarity.

In their seminal paper [22], Turi and Plotkin introduced an elegant mathematical approach to structural operational semantics, where the type of syntax is modeled by an endofunctor Σ and the type of behaviour is modeled by an endofunctor B . Operational semantics is then given by a *distributive law* of Σ over B . In this context, models are *bialgebras*, which consist of a Σ -algebra and a B -coalgebra over a common carrier. One major advantage of this framework over traditional approaches is that it is parametric in the type of behaviour. Indeed, by instantiating the theory to a particular functor B , one can obtain well behaved specification formats for probabilistic and stochastic systems, weighted transition systems, streams, and many more [14, 15, 4].

Turi and Plotkin introduced several kinds of natural transformations involving Σ and B , the most basic one being of the form $\Sigma B \Rightarrow B\Sigma$. If B is a functor representing labelled transition systems, then a typical rule that can be represented in this format is the following:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{a} y'}{x \otimes y \xrightarrow{a} x' \otimes y'} \quad (1)$$

This rule should be read as follows: if x can make an a -transition to x' , and y an a -transition to y' , then $x \otimes y$ can make an a -transition to $x' \otimes y'$. Any specification of the above kind induces a unique *supported*

*The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement nr. 320571, and the Netherlands Organisation for Scientific Research (NWO), CoRE project, dossier number: 612.063.920. Part of this research was carried out during a visit of the author to the University of Warsaw, supported by the Warsaw Center of Mathematics and Computer Science (WCMCS).

model, which is a B -coalgebra over the initial algebra of Σ . If Σ represents a signature and B represents labelled transition systems, then this model is a transition system of which the state space is the set of closed terms in the signature, and, informally, a term makes a transition to another term if and only if there is a rule in the specification justifying this transition.

A more interesting kind is an *abstract GSOS specification*, which is a natural transformation of the form $\Sigma(B \times \text{Id}) \Rightarrow B\Sigma^*$, where Σ^* is the *free monad* for Σ (assuming it exists). If B is the functor that models (image-finite) transition systems, and Σ is a functor representing a signature, then such specifications correspond to classical GSOS specifications [22, 4]. As opposed to the basic format, GSOS rules allow complex terms in conclusions, as in the following rule specifying a constant c :

$$\overline{c \xrightarrow{a} \sigma(c)} \quad (2)$$

where σ is some other operator in the signature (represented by Σ), which can itself be defined by some GSOS rules. The term $\sigma(c)$ is constructed from a constant and a unary operator from the signature, as opposed to the conclusion $x' \otimes y'$ of the rule in (1), which consists of a single operator and variables. Indeed, the free monad Σ^* occurring in an abstract GSOS specification is precisely what allows a complex term such as $\sigma(c)$ in the conclusion.

Dually, one can consider *coGSOS specifications*, which are of the form $\Sigma B^\infty \Rightarrow B(\Sigma + \text{Id})$, where B^∞ is the cofree comonad for B (assuming it exists). In the case of image-finite labelled transition systems, this format corresponds to the *safe ntree format* [22]. A typical coGSOS rule is the following:

$$\frac{x \xrightarrow{a} x' \quad x' \not\xrightarrow{a}}{\sigma(x) \xrightarrow{a} x'} \quad (3)$$

This rule uses two steps of lookahead in the premise; this is supported by the cofree comonad B^∞ in the natural transformation. The symbol $x' \not\xrightarrow{a}$ represents a *negative premise*, which is satisfied whenever x' does not make an a -transition.

Both GSOS and coGSOS specifications induce *distributive laws*, and as a consequence they induce unique supported models on which behavioural equivalence is a congruence. The two formats are incomparable in terms of expressive power: GSOS specifications allow rules that involve complex terms in the conclusion, whereas coGSOS allows arbitrary lookahead in the arguments. It is straightforward to *combine* GSOS and coGSOS as a natural transformation of the form $\Sigma B^\infty \Rightarrow B\Sigma^*$, called a *biGSOS specification*, generalising both formats. However, such specifications are, in some sense, too expressive: they do not induce unique supported models, as already observed in [22]. For example, the rules (2) and (3) above (which are GSOS and coGSOS respectively) can be combined into a single biGSOS specification. Suppose this combined specification has a model. By the axiom for c , there is a transition $c \xrightarrow{a} \sigma(c)$ in this model. However, is there a transition $\sigma(c) \xrightarrow{a} \sigma(c)$? If there is not, then by the rule for σ , there is; but if there is such a transition, then it is not derivable, so it is not in the model! Thus, a supported model does not exist. In fact, it was recently shown that, for biGSOS, it is undecidable whether a (unique) supported model exists [17].

The use of negative premises in the above example (and in [17]) is crucial. In the present paper, we introduce the notion of *monotonicity* of biGSOS specifications, generalising monotone abstract GSOS [8]. In the case that B is a functor representing labelled transition systems, this corresponds to the absence of negative premises, but the format does allow lookahead in premises as well as complex terms in conclusions. Monotonicity requires an *order* on the functor B —technically, our definition of monotonicity is based on the *similarity* order [10] induced on the final coalgebra.

We show that if there is a pointed DCPO structure on the functor B , then any monotone biGSOS specification yields a *least* model as its operational interpretation. Indeed, monotone specifications do not necessarily have a unique model, but it is the least model which makes sense operationally, since this corresponds to the natural notion that every transition has a *finite* proof. Our main result is that if the functor B has a DCPO structure, then every monotone specification yields a canonical *distributive law* of the free monad for Σ over the cofree comonad for B . Its unique model coincides with the least supported model of the specification. As a consequence, behavioural equivalence on this model is a congruence.

However, the conditions of these results are a bit too restrictive: they rule out labelled transition systems, the main example. The problem is that the functors typically used to model transition systems either fail to have a cofree comonad (the powerset functor) or to have a DCPO structure (the finite or countable powerset functor). In the final section, we mitigate this problem using the theory of (countably) presentable categories and accessible functors. This allows us to relax the requirement of DCPO structure only to countable sets, given that the functor B is countably accessible (this is weaker than being finitary, a standard condition in the theory of coalgebras) and the syntax consists only of countably many operations each with finite arity. In particular, this applies to labelled transition systems (with countable branching) and certain kinds of weighted transition systems.

Related work The idea of studying distributive laws of monads over comonads that are not induced by GSOS or coGSOS specifications has been around for some time (e.g., [4]), but, according to a recent overview paper [15], general bialgebraic formats (other than GSOS or coGSOS) which induce such distributive laws have not been proposed so far. In fact, it is shown by Klin and Nachyła that the general problem of extending biGSOS specifications to distributive laws is undecidable [16, 17]. The current paper shows that one does obtain distributive laws from biGSOS specifications when monotonicity is assumed (negative premises are disallowed). A fundamentally different approach to positive formats with lookahead, not based on the framework of bialgebraic semantics but on labelled transition systems modeled very generally in a topos, was introduced in [21]. It is deeply rooted in labelled transition systems, and hence seems incomparable to our approach based on generic coalgebras for ordered functors. An abstract study of distributive laws of monads over comonads and possible morphisms between them is in [18], but it does not include characterisations in terms of simpler natural transformations.

Structure of the paper Section 2 contains the necessary preliminaries on bialgebras and distributive laws. In Section 3 we recall the notion of similarity on coalgebras, which we use in Section 4 to define monotone specifications and prove the existence of least supported models. Section 5 contains our main result: canonical distributive laws for monotone biGSOS specifications. In Section 6, this is extended to countably accessible functors.

Notation We use the categories \mathbf{Set} of sets and functions, \mathbf{PreOrd} of preorders and monotone functions, and \mathbf{DCPO}_\perp of pointed DCPOs and continuous maps. By \mathcal{P} we denote the (contravariant) power set functor; \mathcal{P}_c is the countable power set functor and \mathcal{P}_f the finite power set functor. Given a relation $R \subseteq X \times Y$, we write $\pi_1: R \rightarrow X$ and $\pi_2: R \rightarrow Y$ for its left and right projection, respectively. Given another relation $S \subseteq Y \times Z$ we denote the composition of R and S by $R \circ S$. We let $R^{\text{op}} = \{(y, x) \mid (x, y) \in R\}$. For a set X , we let $\Delta_X = \{(x, x) \mid x \in X\}$. The graph of a function $f: X \rightarrow Y$ is $\text{Graph}(f) = \{(x, f(x)) \mid x \in X\}$. The image of a set $S \subseteq X$ under f is denoted simply by $f(S) = \{f(x) \mid x \in S\}$, and the inverse image of $V \subseteq Y$ by $f^{-1}(V) = \{x \mid f(x) \in V\}$. The pairing of two functions f, g with a common domain is denoted by $\langle f, g \rangle$ and the copairing (for functions f, g with a common codomain) by $[f, g]$. The set of functions

from X to Y is denoted by Y^X . Any relation $R \subseteq Y \times Y$ can be lifted pointwise to a relation on Y^X ; in the sequel we will simply denote such a pointwise extension by the relation itself, i.e., for functions $f, g: X \rightarrow Y$ we have $f R g$ iff $f(x) R g(x)$ for all $x \in X$, or, equivalently, $(f \times g)(\Delta_X) \subseteq R$.

Acknowledgements The author is grateful to Henning Basold, Marcello Bonsangue, Bartek Klin and Beata Naczyła for inspiring discussions and suggestions.

2 (Co)algebras, (co)monads and distributive laws

We recall the necessary definitions on algebras, coalgebras, and distributive laws of monads over comonads. For an introduction to coalgebra see [20, 12]. All of the definitions and results below and most of the examples can be found in [15], which provides an overview of bialgebraic semantics. Unless mentioned otherwise, all functors considered are endofunctors on Set.

2.1 Algebras and monads

An *algebra* for a functor $\Sigma: \text{Set} \rightarrow \text{Set}$ consists of a set X and a function $f: \Sigma X \rightarrow X$. An (*algebra*) *homomorphism* from $f: \Sigma X \rightarrow X$ to $g: \Sigma Y \rightarrow Y$ is a function $h: X \rightarrow Y$ such that $h \circ f = g \circ \Sigma h$. The category of algebras and their homomorphisms is denoted by $\text{alg}(\Sigma)$.

A *monad* is a triple $\mathcal{T} = (T, \eta, \mu)$ where $T: \text{Set} \rightarrow \text{Set}$ is a functor and $\eta: \text{Id} \Rightarrow T$ and $\mu: TT \Rightarrow T$ are natural transformations such that $\mu \circ T\eta = \text{id} = \mu \circ \eta T$ and $\mu \circ \mu T = \mu \circ T\mu$. An (*Eilenberg-Moore, or EM*)-*algebra* for \mathcal{T} is a T -algebra $f: TX \rightarrow X$ such that $f \circ \eta_X = \text{id}$ and $f \circ \mu_X = f \circ Tf$. We denote the category of EM-algebras by $\text{Alg}(\mathcal{T})$.

We assume that a *free monad* (Σ^*, η, μ) for Σ exists. This means that there is a natural transformation $\iota: \Sigma\Sigma^* \Rightarrow \Sigma^*$ such that ι_X is a *free algebra* on the set X of generators, that is, the copairing of

$$\Sigma\Sigma^*X \xrightarrow{\iota_X} \Sigma^*X \xleftarrow{\eta_X} X$$

is an initial algebra for $\Sigma + X$. By Lambek's lemma, $[\iota_X, \eta_X]$ is an isomorphism. Any algebra $f: \Sigma X \rightarrow X$ induces a $\Sigma + X$ -algebra $[f, \text{id}]$, and therefore by initiality a Σ^* -algebra $f^*: \Sigma^*X \rightarrow X$, which we call the *inductive extension* of f . In particular, the inductive extension of ι_X is μ_X . This construction preserves homomorphisms: if h is a homomorphism from f to g , then it is also a homomorphism from f^* to g^* .

Example 1. An algebraic signature (a countable collection of operator names with finite arities) induces a polynomial functor Σ , meaning here a countable coproduct of finite products. The free monad Σ^* constructs terms, that is, Σ^*X is given by the grammar $t ::= \sigma(t_1, \dots, t_n) \mid x$ where x ranges over X and σ ranges over the operator names (and n is the arity of σ), so in particular $\Sigma^*\emptyset$ is the set of closed terms over Σ .

2.2 Coalgebras and comonads

A *coalgebra* for the functor B consists of a set X and a function $f: X \rightarrow BX$. A (*coalgebra*) *homomorphism* from $f: X \rightarrow BX$ to $g: Y \rightarrow BY$ is a function $h: X \rightarrow Y$ such that $Bh \circ f = g \circ h$. The category of B -coalgebras and their homomorphisms is denoted by $\text{coalg}(B)$.

A *comonad* is a triple $\mathcal{D} = (D, \varepsilon, \delta)$ consisting of a functor $D: \text{Set} \rightarrow \text{Set}$ and natural transformations $\varepsilon: D \Rightarrow \text{Id}$ and $\delta: D \Rightarrow DD$ satisfying axioms dual to the monad axioms. The category of Eilenberg-Moore coalgebras for \mathcal{D} , defined dually to EM-algebras, is denoted by $\text{CoAlg}(\mathcal{D})$.

We assume that a *cofree comonad* $(B^\infty, \delta, \varepsilon)$ for B exists. This means that there is a natural transformation $\theta: B^\infty \Rightarrow BB^\infty$ such that θ_X is a *cofree coalgebra* on the set X , that is, the pairing of

$$BB^\infty X \xleftarrow{\theta_X} B^\infty X \xrightarrow{\varepsilon_X} X$$

is a final coalgebra for $B \times X$. Any coalgebra $f: X \rightarrow BX$ induces a $B \times X$ -coalgebra $\langle f, \text{id} \rangle$, and therefore by finality a B^∞ -coalgebra $f^\infty: X \rightarrow B^\infty X$, which we call the *coinductive extension* of f . In particular, the coinductive extension of θ_X is δ_X . This construction preserves homomorphisms: if h is a homomorphism from f to g , then it is also a homomorphism from f^∞ to g^∞ .

Example 2. Consider the Set functor $BX = A \times X$ for a fixed set A . Coalgebras for B are called *stream systems*. There exists a final B -coalgebra, whose carrier can be presented as the set A^ω of all streams over A , i.e., $A^\omega = \{\sigma \mid \sigma: \omega \rightarrow A\}$ where ω is the set of natural numbers. For a set X , $B^\infty X = (A \times X)^\omega$. Given $f: X \rightarrow A \times X$, its coinductive extension $f^\infty: X \rightarrow B^\infty X$ maps a state $x \in X$ to its infinite unfolding. The final coalgebra of $GX = A \times X + 1$ consists of finite and infinite streams over A , that is, elements of $A^* \cup A^\omega$. For a set X , $G^\infty X = (A \times X)^\omega \cup (A \times X)^* \times X$.

Example 3. Labelled transition systems are coalgebras for the functor $(\mathcal{P}-)^A$, where A is a fixed set of labels. Image-finite transition systems are coalgebras for the functor $(\mathcal{P}_f-)^A$, and coalgebras for $(\mathcal{P}_c-)^A$ are transition systems which have, for every action $a \in A$ and every state x , a countable set of outgoing a -transitions from x . A final coalgebra for $(\mathcal{P}-)^A$ does not exist (so there is no cofree comonad for it). However, both $(\mathcal{P}_f-)^A$ and $(\mathcal{P}_c-)^A$ have a final coalgebra, consisting of possibly infinite rooted trees, edge-labelled in A , modulo strong bisimilarity, where for each label, the set of children is finite respectively countable. The cofree comonad of $(\mathcal{P}_f-)^A$ respectively $(\mathcal{P}_c-)^A$, applied to a set X , consist of all trees as above, node-labelled in X .

Example 4. A *complete monoid* is a (necessarily commutative) monoid M together with an infinitary sum operation consistent with the finite sum [7]. Define the functor $\mathcal{M}: \text{Set} \rightarrow \text{Set}$ by $\mathcal{M}(X) = \{\varphi \mid \varphi: X \rightarrow M\}$ and, for $f: X \rightarrow Y$, $\mathcal{M}(h)(\varphi) = \lambda y. \sum_{x \in f^{-1}(y)} \varphi(x)$. A *weighted transition system* over a set of labels A is a coalgebra $f: X \rightarrow (\mathcal{M}X)^A$. Similar to the case of labelled transition systems, we obtain weighted transition systems whose branching is countable for each label as coalgebras for the functor $(\mathcal{M}_c-)^A$, where \mathcal{M}_c is defined by $\mathcal{M}_c(X) = \{\varphi: X \rightarrow M \mid \varphi(x) \neq 0 \text{ for countably many } x \in X\}$. We note that this only requires a countable sum on M to be well-defined and, by further restricting to finite support, weighted transition systems are defined for any commutative monoid (see, e.g., [14]). Labelled transition systems are retrieved by taking the monoid with two elements and logical disjunction as sum. Another example arises by taking the monoid $M = \mathbb{R}^+ \cup \{\infty\}$ of non-negative reals extended with a top element ∞ , with the supremum operation.

2.3 GSOS, coGSOS and distributive laws

Given a signature, a *GSOS rule* [5] σ of arity n is of the form

$$\frac{\{x_{i_j} \xrightarrow{a_j} y_j\}_{j=1..m} \quad \{x_{i_k} \xrightarrow{b_k} \cdot\}_{k=1..l}}{\sigma(x_1, \dots, x_n) \xrightarrow{c} t} \quad (4)$$

where m and l are the number of positive and negative premises respectively; $a_1, \dots, a_m, b_1, \dots, b_l, c \in A$ are labels; $x_1, \dots, x_n, y_1, \dots, y_m$ are pairwise distinct variables, and t is a term over these variables. An *abstract GSOS specification* is a natural transformation of the form

$$\Sigma(B \times \text{Id}) \Rightarrow B\Sigma^*.$$

As first observed in [22], specifications in the GSOS format are generalised by abstract GSOS specifications, where Σ models the signature and $BX = (\mathcal{P}_f X)^A$.

A *safe ntree* rule (as taken from [15]) for σ is of the form $\frac{\{z_i \xrightarrow{a_i} y_i\}_{i \in I} \quad \{w_j \xrightarrow{b_j} \lambda\}_{j \in J}}{\sigma(x_1, \dots, x_n) \xrightarrow{c} t}$ where I and J are countable possibly infinite sets, the z_i, y_i, w_j, x_k are variables, and $b_j, c, a_i \in A$; the x_k and y_i are all distinct and they are the only variables that occur in the rule; the dependency graph of premise variables (where positive premises are seen as directed edges) is well-founded, and t is either a variable or a term built of a single operator from the signature and the variables. A *coGSOS specification* is a natural transformation of the form

$$\Sigma B^\infty \Rightarrow B(\Sigma + \text{Id}).$$

As stated in [22], every safe ntree specification induces a coGSOS specification where Σ models the signature and $BX = (\mathcal{P}_f X)^A$.

A *distributive law* of a monad $\mathcal{T} = (T, \eta, \mu)$ over a comonad $\mathcal{D} = (D, \varepsilon, \delta)$ is a natural transformation $\lambda: TD \Rightarrow DT$ so that $\lambda \circ D\eta = \eta D$, $\varepsilon T \circ \lambda = T\varepsilon$, $\lambda \circ \mu T = D\mu \circ \lambda T \circ T\lambda$ and $D\lambda \circ \lambda D \circ T\delta = \delta T \circ \lambda$. A λ -*bialgebra* is a triple (X, f, g) where X is a set, f is an EM-algebra for \mathcal{T} and g is an EM-coalgebra for \mathcal{D} , such that $g \circ f = Df \circ \lambda_X \circ Tg$.

Every distributive law λ induces, by initiality, a unique coalgebra $h: T\emptyset \rightarrow DT\emptyset$ such that $(T\emptyset, \mu_\emptyset, h)$ is λ -bialgebra. If \mathcal{D} is the cofree comonad for B , then h is the coinductive extension of a B -coalgebra $m: T\emptyset \rightarrow BT\emptyset$, which we call the *operational model* of λ . Behavioural equivalence on the operational model is a congruence. This result applies in particular to abstract GSOS and coGSOS specifications, which both extend to distributive laws of monad over comonad.

A *lifting* of a functor $T: \text{Set} \rightarrow \text{Set}$ to $\text{CoAlg}(\mathcal{D})$ is a functor \overline{T} making the following commute:

$$\begin{array}{ccc} \text{CoAlg}(\mathcal{D}) & \xrightarrow{\overline{T}} & \text{CoAlg}(\mathcal{D}) \\ \downarrow & & \downarrow \\ \text{Set} & \xrightarrow{T} & \text{Set} \end{array}$$

where the vertical arrows represent the forgetful functor, sending a coalgebra to its carrier. Further, a monad $(\overline{T}, \overline{\eta}, \overline{\mu})$ on $\text{CoAlg}(\mathcal{D})$ is a lifting of a monad $\mathcal{T} = (T, \eta, \mu)$ on Set if \overline{T} is a lifting of T , $U\overline{\eta} = \eta U$ and $U\overline{\mu} = \mu U$. A lifting of \mathcal{T} to $\text{coalg}(B)$ is defined similarly.

Distributive laws of \mathcal{T} over \mathcal{D} are in one-to-one correspondence with liftings of (T, η, μ) to $\text{CoAlg}(\mathcal{D})$ (see [13, 22]). If \mathcal{D} is the cofree comonad for B , then $\text{CoAlg}(\mathcal{D}) \cong \text{coalg}(B)$, hence a further equivalent condition is that \mathcal{T} lifts to $\text{coalg}(B)$. In that case, the operational model of a distributive law can be retrieved by applying the corresponding lifting to the unique coalgebra $!: \emptyset \rightarrow B\emptyset$.

3 Similarity

In this section, we recall the notion of *simulations* of coalgebras from [10], and prove a few basic results concerning the similarity preorder on final coalgebras.

An *ordered functor* is a pair (B, \sqsubseteq) of functors $B: \text{Set} \rightarrow \text{Set}$ and $\sqsubseteq: \text{Set} \rightarrow \text{PreOrd}$ such that

$$\begin{array}{ccc} & & \text{PreOrd} \\ & \nearrow \sqsubseteq & \downarrow \\ \text{Set} & \xrightarrow{B} & \text{Set} \end{array}$$

commutes, where the arrow from PreOrd to Set is the forgetful functor. Thus, given an ordered functor, there is a preorder $\sqsubseteq_{BX} \subseteq BX \times BX$ for any set X , and for any map $f: X \rightarrow Y$, Bf is monotone.

The (canonical) *relation lifting* of B is defined on a relation $R \subseteq X \times Y$ by

$$\text{Rel}(B)(R) = \{(b, c) \in BX \times BY \mid \exists d \in BR. B\pi_1(d) = b \text{ and } B\pi_2(d) = c\}.$$

For a detailed account of relation lifting, see, e.g., [11]. Let (B, \sqsubseteq) be an ordered functor. The *lax relation lifting* Rel_{\sqsubseteq} is defined as follows:

$$\text{Rel}_{\sqsubseteq}(B)(R \subseteq X \times Y) = \sqsubseteq_{BX} \circ \text{Rel}(B)(R) \circ \sqsubseteq_{BY}.$$

Let (X, f) and (Y, g) be B -coalgebras. A relation $R \subseteq X \times Y$ is a *simulation* (between f and g) if $R \subseteq (f \times g)^{-1}(\text{Rel}_{\sqsubseteq}(B)(R))$. The greatest simulation between coalgebras f and g is called *similarity*, denoted by \lesssim_f^g , or \lesssim_f if $f = g$, or simply \lesssim if f and g are clear from the context.

Given a set X and an ordered functor (B, \sqsubseteq) , we define the ordered functor $(B \times X, \tilde{\sqsubseteq})$ by

$$(b, x) \tilde{\sqsubseteq}_{BX} (c, y) \quad \text{iff} \quad b \sqsubseteq_{BX} c \text{ and } x = y.$$

The induced notion of simulation can naturally be expressed in terms of the original one:

Lemma 1. *Let \lesssim be the similarity relation between coalgebras $\langle f, f' \rangle: X \rightarrow BX \times Z$ and $\langle g, g' \rangle: X \rightarrow BX \times Z$. Then for any relation $R \subseteq X \times X$, we have $R \subseteq (\langle f, f' \rangle \times \langle g, g' \rangle)^{-1}(\text{Rel}_{\tilde{\sqsubseteq}}(B \times Z)(R))$ iff $R \subseteq (f \times g)^{-1}(\text{Rel}_{\sqsubseteq}(B)(R))$ and for all $(x, y) \in R$: $f'(x) = g'(x)$.*

Given an ordered functor (B, \sqsubseteq) we write

$$\lesssim_{B^\infty X}$$

for the similarity order induced by $(B \times X, \tilde{\sqsubseteq})$ on the cofree coalgebra $(B^\infty X, \langle \theta_X, \varepsilon_X \rangle)$. We discuss a few examples of ordered functors and similarity—see [10] for many more.

Example 5. For the functor $L_f X = (\mathcal{P}_f X)^A$ ordered by (pointwise) subset inclusion, a simulation as defined above is a (strong) simulation in the standard sense. For elements $p, q \in L_f^\infty X$, we have $p \lesssim_{L_f^\infty X} q$ iff there exists a (strong) simulation between the underlying trees of p and q , so that related pairs agree on labels in X .

Example 6. For any $G: \text{Set} \rightarrow \text{Set}$, the functor $B = G + 1$, where $1 = \{\perp\}$, can be ordered as follows: $x \leq y$ iff $x = \perp$ or $x = y$, for all $x, y \in BX$. If $G = A \times \text{Id}$ then $B^\infty X$ consists of finite and infinite sequences of the form $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} \dots$ with $x_i \in X$ and $a_i \in A$ for each i (cf. Example 2). For $\sigma, \tau \in B^\infty X$ we have $\sigma \lesssim_{B^\infty X} \tau$ if σ does not terminate before τ does, and σ and τ agree on labels in X and A on each position where σ is defined.

Lemma 2. *Coalgebra homomorphisms h, k preserve similarity: if $x \lesssim y$ then $h(x) \lesssim k(y)$.*

In the remainder of this section we state a few technical properties concerning similarity on cofree comonads, which will be necessary in the following sections. The proofs use Lemma 2 and a few basic, standard properties of relation lifting.

Pointwise inequality of coalgebras implies pointwise similarity of coinductive extensions:

Lemma 3. *Let (B, \sqsubseteq) be an ordered functor, and let f and g be B -coalgebras on a common carrier X . If $(f \times g)(\Delta_X) \subseteq \sqsubseteq_{BX}$ then $(f^\infty \times g^\infty)(\Delta_X) \subseteq \lesssim_{B^\infty X}$.*

Recall from Section 2 that any B -homomorphism yields a B^∞ -homomorphism between coinductive extensions. A similar fact holds for inequalities.

Lemma 4. *Let (B, \sqsubseteq) be an ordered functor where B preserves weak pullbacks, and let $f: X \rightarrow BX$, $g: Y \rightarrow BY$ and $h: X \rightarrow Y$.*

- If $Bh \circ f \sqsubseteq_{BY} g \circ h$ then $B^\infty h \circ f^\infty \lesssim_{B^\infty Y} g^\infty \circ h$, and conversely,
- if $Bh \circ f \sqsupseteq_{BY} g \circ h$ then $B^\infty h \circ f^\infty \gtrsim_{B^\infty Y} g^\infty \circ h$.

4 Monotone biGSOS specifications

As discussed in the introduction, GSOS and coGSOS have a straightforward common generalisation, called *biGSOS* specifications. Throughout this section we assume (B, \sqsubseteq) is an ordered functor, B has a cofree comonad and Σ has a free monad.

Definition 1. A *biGSOS specification* is a natural transformation of the form $\rho : \Sigma B^\infty \Rightarrow B\Sigma^*$. A triple (X, a, f) consisting of a set X , an algebra $a : \Sigma X \rightarrow X$ and a coalgebra $f : X \rightarrow BX$ (i.e., a bialgebra) is called a ρ -*model* if the following diagram commutes:

$$\begin{array}{ccc} \Sigma X & \xrightarrow{a} & X \\ \Sigma f^\infty \downarrow & & \downarrow f \\ \Sigma B^\infty X & \xrightarrow{\rho_X} B\Sigma^* X \xrightarrow{Ba^*} & BX \end{array}$$

If $BX = (\mathcal{P}_f X)^A$, then one can obtain biGSOS specifications from concrete rules in the *ntree* format, which combines GSOS and safe ntree, allowing lookahead in premises, negative premises and complex terms in conclusions.

Of particular interest are ρ -models on the initial algebra $\iota_\emptyset : \Sigma\Sigma^*\emptyset \rightarrow \Sigma^*\emptyset$:

$$\begin{array}{ccc} \Sigma\Sigma^*\emptyset & \xrightarrow{\iota_\emptyset} & \Sigma^*\emptyset \\ \Sigma f^\infty \downarrow & & \downarrow f \\ \Sigma B^\infty \Sigma^*\emptyset & \xrightarrow{\rho_{\Sigma^*\emptyset}} B\Sigma^* \Sigma^*\emptyset \xrightarrow{B\mu_\emptyset} & B\Sigma^*\emptyset \end{array} \quad (5)$$

(Notice that $\iota_\emptyset^* = \mu_\emptyset$.) We call these *supported models*. Indeed, for labelled transition systems, this notion coincides with the standard notion of the supported model of an SOS specification (e.g., [1]).

In the introduction, we have seen that biGSOS specifications do not necessarily induce a supported model. But even if they do, such a model is not necessarily unique, and behavioural equivalence is not even a congruence, in general, as shown by the following example.

Example 7. In this example we consider a signature with constants c and d , and unary operators σ and τ . Consider the specification (represented by concrete rules) on labelled transition systems where c and d are not assigned any behaviour, and σ and τ are given by the following rules:

$$\frac{x \xrightarrow{a} x' \quad x' \xrightarrow{a} x''}{\sigma(x) \xrightarrow{a} x''} \quad \frac{}{\tau(x) \xrightarrow{a} \sigma(\tau(x))}$$

The behaviour of $\tau(x)$ is independent of its argument x . Which transitions can occur in a supported model? First, for any t there is a transition $\tau(t) \xrightarrow{a} \sigma(\tau(t))$. Moreover, a transition $\sigma(\tau(t)) \xrightarrow{a} t''$ can be in the model, although it does not need to be. But if it is there, it is supported by an *infinite* proof.

In fact, one can easily construct a model in which the behaviour of $\sigma(\tau(c))$ is different from that of $\sigma(\tau(d))$ —for example, a model where $\sigma(\tau(c))$ does not make any transitions, whereas $\sigma(\tau(d)) \xrightarrow{a} t$ for some t . Then behavioural equivalence is not a congruence; c is bisimilar to d , but $\sigma(\tau(c))$ is not bisimilar to $\sigma(\tau(d))$.

The above example features a specification that has many different interpretations as a supported model. However, there is only one which makes sense: the *least* model, which only features *finite* proofs. It is sensible to speak about the least model of this specification, since it does not contain any negative premises. More generally, absence of negative premises can be defined based on an ordered functor and the induced similarity order.

Definition 2. A biGSOS specification $\rho: \Sigma B^\infty \Rightarrow B\Sigma^*$ is *monotone* if the restriction of $\rho_X \times \rho_X$ to $\text{Rel}(\Sigma)(\lesssim_{B^\infty X})$ corestricts to $\sqsubseteq_{B\Sigma^* X}$, for any set X .

If Σ represents an algebraic signature, then monotonicity can be conveniently restated as follows (c.f. [6], where monotone GSOS is characterised in a similar way). For every operator σ :

$$\frac{b_1 \lesssim_{B^\infty X} c_1 \quad \dots \quad b_n \lesssim_{B^\infty X} c_n}{\rho_X(\sigma(b_1, \dots, b_n)) \sqsubseteq_{B\Sigma^* X} \rho_X(\sigma(c_1, \dots, c_n))}$$

for every set X and every $b_1, \dots, b_n, c_1, \dots, c_n \in B^\infty X$. Thus, in a monotone specification, if c_i simulates b_i for each i , then the behaviour of $\sigma(b_1, \dots, b_n)$ is “less than” the behaviour of $\sigma(c_1, \dots, c_n)$.

In the case of labelled transition systems, it is straightforward that monotonicity rules out (non-trivial use of) negative premises. Notice that the example specification in the introduction consisting of rules (2) and (3), which does not have a model, is not monotone. This is no coincidence: every monotone biGSOS specification has a model, if $B\Sigma^*\emptyset$ is a pointed DCPO, as we will see next. In fact, the proper canonical choice is the *least* model, corresponding to behaviour obtained in finitely many proof steps.

4.1 Models of monotone specifications

Let ρ be a monotone biGSOS specification. Suppose $B\Sigma^*\emptyset$ is a pointed DCPO. Then the set of coalgebras $\text{coalg}(B)_{\Sigma^*\emptyset} = \{f \mid f: \Sigma^*\emptyset \rightarrow B\Sigma^*\emptyset\}$, ordered pointwise, is a pointed DCPO as well.

Consider the function $\varphi: \text{coalg}(B)_{\Sigma^*\emptyset} \rightarrow \text{coalg}(B)_{\Sigma^*\emptyset}$, defined as follows:

$$\varphi(f) = B\mu_\emptyset \circ \rho_{\Sigma^*\emptyset} \circ \Sigma f^\infty \circ \iota_\emptyset^{-1} \quad (6)$$

Since ι_\emptyset is an isomorphism, a function f is a fixed point of φ if and only if it is a supported model of ρ (Equation (5)). We are interested in the *least* supported model. To show that it exists, since $\text{coalg}(B)_{\Sigma^*\emptyset}$ is a pointed DCPO, it suffices to show that φ is monotone.

Lemma 5. *The function φ is monotone.*

Proof. Suppose $f, g: \Sigma^*\emptyset \rightarrow B\Sigma^*\emptyset$ and $f \sqsubseteq_{B\Sigma^*\emptyset} g$. By Lemma 3, we have $f^\infty \lesssim_{B^\infty \Sigma^*\emptyset} g^\infty$. From standard properties of relation lifting we derive $\Sigma f^\infty \text{Rel}(\Sigma)(\lesssim_{B^\infty \Sigma^*\emptyset}) \Sigma g^\infty$ and now the result follows by monotonicity of ρ (assumption) and monotonicity of $B\mu_\emptyset$ (B is ordered). \square

Corollary 1. *If $B\Sigma^*\emptyset$ is a pointed DCPO and ρ is a monotone biGSOS specification, then ρ has a least supported model.*

The condition of the Corollary is satisfied if B is of the form $B = G + 1$ (c.f. Example 6), that is, $B = G + 1$ for some functor G (where the element in the singleton 1 is interpreted as the least element of the pointed DCPO). Consider, as an example, the functor $BX = A \times X + 1$ of finite and infinite streams over A . Any specification that does not mention termination (i.e., a specification for the functor $GX = A \times X$) yields a monotone specification for B .

Example 8. Consider the following specification (in terms of rules) for the functor $BX = \mathbb{N} \times X + 1$ of (possibly terminating) stream systems over the natural numbers. It specifies a unary operator σ , a binary operator \oplus , infinitely many unary operators $m \otimes -$ (one for each $m \in \mathbb{N}$), and constants *ones*, *pos*, *c*:

$$\frac{x \xrightarrow{n} x' \quad x' \xrightarrow{m} x''}{\sigma(x) \xrightarrow{n} n \otimes (m \otimes \sigma(x''))} \quad \frac{x \xrightarrow{n} x' \quad y \xrightarrow{m} y'}{x \oplus y \xrightarrow{n+m} x' \oplus y'} \quad \frac{x \xrightarrow{n} x'}{m \otimes x \xrightarrow{m \times n} m \otimes x'}$$

$$\overline{\text{ones} \xrightarrow{1} \text{ones}} \quad \overline{\text{pos} \xrightarrow{1} \text{ones} \oplus \text{pos}} \quad \overline{c \xrightarrow{1} \sigma(c)}$$

where $+$ and \times denote addition and multiplication of natural numbers, respectively. This induces a monotone biGSOS specification; the rule for σ is GSOS nor coGSOS, since it uses both lookahead and a complex conclusion. By the above Corollary, it has a model. The coinductive extension maps pos to the increasing stream of positive integers, and $\sigma(\text{pos})$ is the stream $(1, 6, 120, \dots) = (1!, 3!, 5!, \dots)$. But c does not represent an infinite stream, since $\sigma(c)$ is undefined.

The case of *labelled transition systems* is a bit more subtle. The problem is that $(\mathcal{P}_f \Sigma^* \emptyset)^A$ and $(\mathcal{P}_c \Sigma^* \emptyset)^A$ are not DCPOs, in general, whereas the functor $(\mathcal{P} -)^A$ does not have a cofree comonad. However, if the set of closed terms $\Sigma^* \emptyset$ is countable, then $(\mathcal{P}_c \Sigma^* \emptyset)^A$ is a pointed DCPO, and thus Corollary 1 applies. The specification in Example 7 can be viewed as a specification for the functor $(\mathcal{P}_c -)^A$, and it has a countable set of terms. Therefore it has, by the Corollary, a least supported model. In this model, the behaviour of $\sigma(t)$ is empty, for any $t \in \Sigma^* \emptyset$.

5 Distributive laws for biGSOS specifications

In the previous section we have seen how to construct a least supported model of a monotone biGSOS specification, as the least fixed point of a monotone function. In the present section we show that, given a monotone biGSOS specification, the construction of a least model generalizes to a *lifting* of the free monad Σ^* to the category of B -coalgebras. It then immediately follows that there exists a canonical distributive law of the monad Σ^* over the comonad B^∞ , and that the (unique) operational model of this distributive law corresponds to the least supported model as constructed above.

In order to proceed we define a DCPO $_{\perp}$ -ordered functor as an ordered functor (Section 3) where PreOrd is replaced by DCPO $_{\perp}$. Below we assume that (B, \sqsubseteq) is DCPO $_{\perp}$ -ordered, and Σ and B are as before (having a free monad and cofree comonad respectively).

Example 9. A general class of functors that are DCPO $_{\perp}$ -ordered are those of the form $B + 1$, where the singleton 1 is interpreted as the least element and all other distinct elements are incomparable (see Example 6). Another example is the functor $(\mathcal{P} -)^A$ of labelled transition systems with arbitrary branching, but this example can not be treated here because there exists no cofree comonad for it. The case of labelled transition systems is treated in Section 6.

Let $\text{coalg}(B)_{\Sigma^* X}$ be the set of B -coalgebras with carrier $\Sigma^* X$, pointwise ordered as a DCPO by the order on B . The lifting of Σ^* to $\text{coalg}(B)$ that we are about to define maps a coalgebra $c: X \rightarrow BX$ to the least coalgebra $\bar{c}: \Sigma^* X \rightarrow B\Sigma^* X$, w.r.t. the above order on $\text{coalg}(B)_{\Sigma^* X}$, making the following diagram commute.

$$\begin{array}{ccccc} \Sigma B^\infty \Sigma^* X & \xrightarrow{\rho_{\Sigma^* X}} & B\Sigma^* \Sigma^* X & \xrightarrow{B\mu_X} & B\Sigma^* X & \xleftarrow{B\eta_X} & BX \\ \Sigma(\bar{c})^\infty \uparrow & & & & \bar{c} \uparrow & & \uparrow c \\ \Sigma \Sigma^* X & \xrightarrow{\quad \quad \quad} & \Sigma^* X & \xleftarrow{\eta_X} & X & & \end{array}$$

Equivalently, \bar{c} is the least fixed point of the operator $\varphi_c: \text{coalg}(B)_{\Sigma^* X} \rightarrow \text{coalg}(B)_{\Sigma^* X}$ defined by

$$\varphi_c(f) = [B\mu_X \circ \rho_{\Sigma^* \emptyset} \circ \Sigma f^\infty, B\eta_X \circ c] \circ [t_X, \eta_X]^{-1}.$$

Following the proof of Lemma 5 it is easy to verify:

Lemma 6. *For any $c: X \rightarrow BX$, the function φ_c is monotone.*

For the lifting of Σ^* , we need to show that the above construction preserves coalgebra morphisms.

Theorem 1. *The functor $\overline{\Sigma}^* : \text{coalg}(B) \rightarrow \text{coalg}(B)$ defined by*

$$\overline{\Sigma}^*(X, c) = (\Sigma^*X, \bar{c}) \quad \text{and} \quad \overline{\Sigma}^*(h) = \Sigma^*h$$

is a lifting of the functor Σ^ .*

Proof. Let (X, c) and (Y, d) be $B\Sigma^*$ -coalgebras. We need to prove that, if $h: X \rightarrow Y$ is a coalgebra homomorphism from c to d , then Σ^*h is a homomorphism from \bar{c} to \bar{d} .

The proof is by transfinite induction on the iterative construction of \bar{c} and \bar{d} as limits of the ordinal-indexed initial chains of φ_c and φ_d respectively. For the limit (and base) case, given a (possibly empty) directed family of coalgebras $f_i: \Sigma^*X \rightarrow B\Sigma^*X$ and another directed family $g_i: \Sigma^*Y \rightarrow B\Sigma^*Y$, such that $B\Sigma^*h \circ f_i = g_i \circ \Sigma^*h$ for all i , we have $B\Sigma^*h \circ \bigvee_i f_i = \bigvee_i (B\Sigma^*h \circ f_i) = \bigvee_i (g_i \circ \Sigma^*h) = (\bigvee_i g_i) \circ \Sigma^*h$ by continuity of $B\Sigma^*h$ and assumption.

Let $f: \Sigma^*X \rightarrow B\Sigma^*X$ and $g: \Sigma^*Y \rightarrow B\Sigma^*Y$ be such that $B\Sigma^*h \circ f = g \circ \Sigma^*h$. To prove: $B\Sigma^*h \circ \varphi_c(f) = \varphi_d(g) \circ \Sigma^*h$, i.e., commutativity of the outside of:

$$\begin{array}{ccccccc} \Sigma^*X & \xrightarrow{[l_X, \eta_X]^{-1}} & \Sigma\Sigma^*X + X & \xrightarrow{\Sigma f^\infty + c} & \Sigma B^\infty \Sigma^*X + BX & \xrightarrow{\rho_{\Sigma^*X} + \text{id}} & B\Sigma^*\Sigma^*X + BX & \xrightarrow{[B\mu_X, B\eta_X]} & B\Sigma^*X \\ \Sigma^*h \downarrow & & \downarrow \Sigma\Sigma^*h + h & & \downarrow \Sigma B^\infty \Sigma^*h + Bh & & \downarrow B\Sigma^*\Sigma^*h + Bh & & \downarrow B\Sigma^*h \\ \Sigma^*Y & \xrightarrow{[l_Y, \eta_Y]^{-1}} & \Sigma\Sigma^*Y + Y & \xrightarrow{\Sigma g^\infty + d} & \Sigma B^\infty \Sigma^*Y + BY & \xrightarrow{\rho_{\Sigma^*Y} + \text{id}} & B\Sigma^*\Sigma^*Y + BY & \xrightarrow{[B\mu_Y, B\eta_Y]} & B\Sigma^*Y \end{array}$$

From left to right, the first square commutes by naturality of $[l, \eta]$ (and the fact that it is an isomorphism), the second by assumption that Σ^*h is a B -coalgebra homomorphism from f to g (and therefore a B^∞ -coalgebra homomorphism) and the assumption that h is a coalgebra homomorphism from c to d , the third by naturality of ρ , and the fourth by naturality of μ and η . \square

We show that the (free) monad (Σ^*, η, μ) lifts to $\text{coalg}(B)$. This is the heart of the matter. The main proof obligation is to show that μ_X is a coalgebra homomorphism from $\overline{\Sigma}^*(\Sigma^*(X, c))$ to $\overline{\Sigma}^*(X, c)$, for any B -coalgebra (X, c) .

Theorem 2. *The monad (Σ^*, η, μ) on Set lifts to the monad $(\overline{\Sigma}^*, \eta, \mu)$ on $\text{coalg}(B)$, if B preserves weak pullbacks.*

The lifting gives rise to a distributive law of monad over comonad.

Theorem 3. *Let $\rho: \Sigma B^\infty \Rightarrow B\Sigma^*$ be a monotone biGSOS specification, where B is DCPO_\perp -ordered and preserves weak pullbacks. There exists a distributive law $\lambda: \Sigma^*B^\infty \Rightarrow B^\infty\Sigma^*$ of the free monad Σ^* over the cofree comonad B^∞ such that the operational model of λ is the least supported model of ρ .*

Proof. By Theorem 2, we obtain a lifting of (Σ^*, η, μ) to $\text{coalg}(B)$. As explained in the preliminaries, such a lifting corresponds uniquely to a distributive law of the desired type. The operational model of λ is obtained by applying the lifting to the unique coalgebra $! : \emptyset \rightarrow B\emptyset$. But that coincides, by definition of the lifting, with the least supported model as defined in Section 4. \square

It follows from the general theory of bialgebras that the unique coalgebra morphism from the least supported model to the final coalgebra is an algebra homomorphism, i.e., behavioural equivalence on the least supported model of a monotone biGSOS specification is a congruence.

Labelled transition systems The results above do not apply to labelled transition systems. The problem is that the cofree comonad for the functor $(\mathcal{P}-)^A$ does not exist. A first attempt would be to restrict to the finitely branching transition systems, i.e., coalgebras for the functor $(\mathcal{P}_f-)^A$. But this functor is not DCPO_\perp -ordered, and indeed, contrary to the case of GSOS and coGSOS, even with a finite biGSOS specification one can easily generate a least model with infinite branching, so that a lifting as in the previous section can not exist.

Example 10. Consider the following specification on (finitely branching) labelled transition systems, involving a unary operator σ and a constant c :

$$\frac{}{c \xrightarrow{a} \sigma(c)} \quad \frac{}{\sigma(x) \xrightarrow{a} \sigma(\sigma(x))} \quad \frac{x \xrightarrow{a} x' \xrightarrow{a} x'' \xrightarrow{a} x'''}{\sigma(x) \xrightarrow{a} x'''}$$

The left rule for σ constructs an infinite chain of transitions from $\sigma(x)$ for any x , so in particular for $\sigma(c)$. The right rule takes the transitive closure of transitions from $\sigma(c)$, so in the least model there are infinitely many transitions from $\sigma(c)$.

The model in the above example has countable branching. One might ask whether it can be adapted to generate uncountable branching, i.e., that we can construct a biGSOS specification for the functor $(\mathcal{P}_c-)^A$, such that the model of this specification would feature uncountable branching. However, as it turns out, this is not the case, at least if we assume Σ to be a polynomial functor (a countable coproduct of finite products, modelling a signature with countably many operations each of finite arity), and the set of labels A to be countable. This is shown more generally in the next section.

6 Liftings for countably accessible functors

In the previous section, we have seen that one of the most important instances of the framework—the case of labelled transition systems—does not work, because of size issues: the functors in question either do not have a cofree comonad, or are not DCPO -ordered. In the current section, we solve this problem by showing that, if both functors B, Σ are reasonably well-behaved, then it suffices to have a DCPO -ordering of B only on *countable* sets.

More precisely, let cSet be the full subcategory of countable sets, with inclusion $I: \text{cSet} \rightarrow \text{Set}$. We assume that (B, \sqsubseteq) is an ordered functor on Set , and that its restriction to countable sets is DCPO_\perp -ordered:

$$\begin{array}{ccccc} & & \text{DCPO}_\perp & \longrightarrow & \text{PreOrd} \\ & \nearrow \sqsubseteq & & & \downarrow \\ \text{cSet} & \xrightarrow{I} & \text{Set} & \xrightarrow{B} & \text{Set} \\ & & \nearrow \sqsubseteq & & \end{array}$$

This is a weaker assumption than in Section 5: before, every set BX was assumed to be a pointed DCPO , whereas here, they only need to be pointed DCPO s when X is countable (and just a preorder otherwise).

Example 11. The functor $(\mathcal{P}_c-)^A$ coincides with the DCPO_\perp -ordered functor $(\mathcal{P}-)^A$ when restricted to countable sets, hence it satisfies the above assumption. Notice that $(\mathcal{P}_c-)^A$ is not DCPO_\perp -ordered. The functor $(\mathcal{P}_f-)^A$ does not satisfy the above assumption.

The functor $(\mathcal{M}-)^A$, for the complete monoid $\mathbb{R}^+ \cup \{\infty\}$ (Example 4), is ordered as a complete lattice [19], so also DCPO_\perp -ordered. Similar to the above, the functor $(\mathcal{M}_c-)^A$ is DCPO_\perp -ordered when restricted to countable sets, i.e., satisfies the above assumption.

We define $\text{coalg}_c(B)$ to be the full subcategory of B -coalgebras whose carrier is a countable set, with inclusion $\bar{I}: \text{coalg}_c(B) \rightarrow \text{coalg}(B)$. The associated forgetful functor is denoted by $U: \text{coalg}_c(B) \rightarrow \text{cSet}$.

The pointed DCPO structure on each BX , for X countable, suffices to carry out the fixed point constructions from the previous sections for coalgebras over countable sets, if we assume that Σ^* preserves countable sets. Notice, moreover, that the (partial) order on the functor B is still necessary to define the simulation order on $B^\infty X$, and hence speak about monotonicity of biGSOS specifications. The proof of the following theorem is essentially the same as in the previous section.

Theorem 4. *Suppose Σ^* preserves countable sets, and B is an ordered functor which preserves weak pull-backs and whose restriction to cSet is DCPO $_{\perp}$ -ordered. Let $(\Sigma_c^*, \eta^c, \mu^c)$ be the restriction of (Σ^*, η, μ) to cSet . Any monotone biGSOS specification $\rho: \Sigma B^\infty \Rightarrow B\Sigma^*$ gives rise to a lifting $(\bar{\Sigma}_c^*, \bar{\eta}^c, \bar{\mu}^c)$ of the monad $(\Sigma_c^*, \eta^c, \mu^c)$ to $\text{coalg}_c(B)$.*

In the remainder of this section, we will show that, under certain assumptions on B and Σ^* , the above lifting extends to a lifting of the monad Σ^* from Set to $\text{coalg}(B)$, and hence a distributive law of the monad Σ^* over the cofree comonad B^∞ . It relies on the fact that, under certain conditions, we can present every coalgebra as a (filtered) colimit of coalgebras over countable sets.

We use the theory of locally (countably, i.e., ω_1 -) presentable categories and (countably) accessible categories. Because of space limits we can not properly recall that theory in detail here (see [3]); we only recall a concrete characterisation of when a functor on Set is countably accessible, since that will be assumed both for B and Σ^* later on. On Set , a functor $B: \text{Set} \rightarrow \text{Set}$ is *countably accessible* if for every set X and element $x \in BX$, there is an injective function $i: Y \rightarrow X$ from a finite set Y and an element $y \in BY$ such that $Bi(y) = x$. Intuitively, such functors are determined by how they operate on countable sets.

Example 12. Any finitary functor is countably accessible. Further, the functors $(\mathcal{P}_c -)^A$ and $(\mathcal{M}_c -)^A$ (c.f. Example 11) are countably accessible if A is countable.

A functor is called *strongly countably accessible* if it is countably accessible and additionally preserves countable sets, i.e., it restricts to a functor $\text{cSet} \rightarrow \text{cSet}$. We will assume this for our “syntax” functor Σ^* . If Σ corresponds to a signature with countably many operations each of finite arity (so is a countable coproduct of finite products) then Σ^* is strongly countably accessible.

The central idea of obtaining a lifting to $\text{coalg}(B)$ from a lifting to $\text{coalg}_c(B)$ is to *extend* the monad on $\text{coalg}_c(B)$ along the inclusion $\bar{I}: \text{coalg}_c(B) \rightarrow \text{coalg}(B)$. Concretely, a functor $T: \text{Set} \rightarrow \text{Set}$ extends $T_c: \text{cSet} \rightarrow \text{cSet}$ if there is a natural isomorphism $\alpha: IT_c \Rightarrow TI$. A monad (T, η, μ) on Set extends a monad (T_c, η_c, μ_c) on cSet if T_c extends T with some isomorphism α such that $\alpha \circ I\eta_c = \eta I$ and $\alpha \circ I\mu_c = \mu I \circ T\alpha \circ \alpha T_c$. This notion of extension is generalised naturally to arbitrary locally countably presentable categories. Monads on the category of countably presentable objects can always be extended.

Lemma 7. *Let \mathcal{C} be a locally countably presentable category, with $I: \mathcal{C}_c \rightarrow \mathcal{C}$ the subcategory of countably presentable objects. Any monad (T_c, η^c, μ^c) on \mathcal{C}_c extends uniquely to a monad (T, η, μ) on \mathcal{C} , along $I: \mathcal{C}_c \rightarrow \mathcal{C}$.*

Since B is countably accessible, $\text{coalg}(B)$ is locally countably presentable and $\text{coalg}_c(B)$ is the associated category of countably presentable objects [2]. This means every B -coalgebra can be presented as a filtered colimit of B -coalgebras with countable carriers. The above lemma applies, so we can extend the monad on $\text{coalg}_c(B)$ of Theorem 4 to a monad on $\text{coalg}(B)$, resulting in Theorem 6 below. The latter relies on Theorem 5, which ensures that, doing so, we will get a lifting of the monad on Set that we started with.

In the remainder of this section, we will consider a slightly relaxed version of functor liftings, up to isomorphism, similar to extensions defined before. This is harmless—those still correspond to distributive laws—but since the monad on $\text{coalg}(B)$ is constructed only up to isomorphism, it is more natural to work with in this setting. We say $(\bar{T}, \bar{\eta}, \bar{\mu})$ lifts (T, η, μ) (up to isomorphism) if there is a natural isomorphism $\alpha: U\bar{T} \Rightarrow TU$ such that $\alpha \circ U\bar{\eta} = \eta U$ and $\alpha \circ U\bar{\mu} = \mu U \circ T\alpha \circ \alpha\bar{T}$.

Theorem 5. *Let $B: \text{Set} \rightarrow \text{Set}$ be countably accessible. Suppose (T_c, η^c, μ^c) is a monad on cSet , which lifts to a monad $(\bar{T}_c, \bar{\eta}^c, \bar{\mu}^c)$ on $\text{coalg}_c(B)$. Then*

1. (T_c, η^c, μ^c) extends to (T, η, μ) along $I: \text{Set}_c \rightarrow \text{Set}$,
2. $(\bar{T}_c, \bar{\eta}^c, \bar{\mu}^c)$ extends to $(\bar{T}, \bar{\eta}, \bar{\mu})$ along $\bar{I}: \text{coalg}_c(B) \rightarrow \text{coalg}(B)$,
3. $(\bar{T}, \bar{\eta}, \bar{\mu})$ is a lifting (up to isomorphism) of (T, η, μ) .

By instantiating the above theorem with the lifting of Theorem 4, the third point gives us the desired lifting to $\text{coalg}(B)$. In particular T_c is instantiated to the restriction Σ_c^* of Σ^* , which means that the extension in the first point is just Σ^* itself.

Theorem 6. *Let $\rho: \Sigma B^\infty \Rightarrow B\Sigma^*$ be a monotone biGSOS specification, where B is an ordered functor whose restriction to countable sets is DCPO_\perp -ordered, B is countably accessible, B preserves weak pullbacks, and Σ^* is strongly countably accessible. There exists a distributive law $\lambda: \Sigma^* B^\infty \Rightarrow B^\infty \Sigma^*$ of the free monad Σ^* over the cofree comonad B^∞ such that the operational model of λ is the least supported model of ρ .*

As explained in Example 12 and Example 11, if B is either $(\mathcal{P}_c -)^A$ or $(\mathcal{M}_c -)^A$ (weighted in the non-negative real numbers) with A countable, then it satisfies the above hypotheses (that \mathcal{M}_c preserves weak pullbacks follows essentially from [9]). So the above theorem applies to labelled transition systems and weighted transition systems (of the above type) over a countable set of labels, as long as the syntax is composed of countably many operations each with finite arity. Hence, behavioural equivalence on the operational model of any biGSOS specification for such systems is a congruence.

7 Future work

In this paper we provided a bialgebraic foundation of positive specification formats over ordered functors, involving rules that feature lookahead in the premises as well as complex terms in conclusions. From a practical point of view, it would be interesting to find more concrete rules formats corresponding to the abstract format of the present paper. In particular, concrete GSOS formats for weighted transition systems exist [14]; they could be a good starting point.

It is currently unclear to us whether the assumption of weak pullback preservation in the main results is necessary. This assumption is used in our proof of Lemma 4, which in turn is used in the proof that the free monad lifts to the category of coalgebras (Theorem 2). Finally, we would like to study *continuous* specifications, as opposed to specifications that are only monotone, as in the current paper. Continuous specifications should be better behaved than monotone ones. However, it is currently not yet clear how to characterize continuity of a specification both at the concrete, syntactic level.

References

- [1] L. Aceto, W. Fokkink & C. Verhoef (2001): *Structural Operational Semantics*. In: *Handbook of Process Algebra*, Elsevier Science, pp. 197–292, doi:10.1016/B978-044482830-9/50021-7.

- [2] J. Adámek & H-E. Porst (2004): *On tree coalgebras and coalgebra presentations*. *Theor. Comput. Sci.* 311(1-3), pp. 257–283, doi:10.1016/S0304-3975(03)00378-5.
- [3] J. Adámek & J. Rosický (1994): *Locally Presentable and Accessible Categories*. Cambridge Tracts in Mathematics, Cambridge University Press, doi:10.1017/CBO9780511600579.
- [4] F. Bartels (2004): *On generalised coinduction and probabilistic specification formats*. Ph.D. thesis, CWI, Amsterdam.
- [5] B. Bloom, S. Istrail & A. Meyer (1995): *Bisimulation Can't be Traced*. *J. ACM* 42(1), pp. 232–268, doi:10.1145/200836.200876.
- [6] F. Bonchi, D. Petrisan, D. Pous & J. Rot (2017): *A general account of coinduction up-to*. *Acta Inf.* 54(2), pp. 127–190, doi:10.1007/s00236-016-0271-4.
- [7] M. Droste & W. Kuich (2009): *Semirings and formal power series*. In: *Handbook of Weighted Automata*, Springer, pp. 3–28, doi:10/bj2xgm.
- [8] M. Fiore & S. Staton (2010): *Positive structural operational semantics and monotone distributive laws*. In: *CMCS Short Contributions*, p. 8.
- [9] H. P. Gumm & T. Schröder (2001): *Monoid-labeled transition systems*. *Electr. Notes Theor. Comput. Sci.* 44(1), pp. 185–204, doi:10.1016/S1571-0661(04)80908-3.
- [10] J. Hughes & B. Jacobs (2004): *Simulations in coalgebra*. *Theor. Comput. Sci.* 327(1-2), pp. 71–108, doi:10.1016/j.tcs.2004.07.022.
- [11] B. Jacobs (2016): *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science 59, Cambridge University Press, doi:10.1017/CBO9781316823187.
- [12] B. Jacobs & J. Rutten (2011): *An introduction to (co)algebras and (co)induction*. In: *Advanced Topics in Bisimulation and Coinduction*, Cambridge University Press, pp. 38–99, doi:10.1017/CBO9780511792588.003.
- [13] P. T. Johnstone (1975): *Adjoint lifting theorems for categories of algebras*. *Bulletin of the London Mathematical Society* 7(3), pp. 294–297, doi:10.1112/blms/7.3.294.
- [14] B. Klin (2009): *Structural Operational Semantics for Weighted Transition Systems*. In J. Palsberg, editor: *Semantics and Algebraic Specification*, LNCS 5700, Springer, pp. 121–139, doi:10/cxqzcf.
- [15] B. Klin (2011): *Bialgebras for structural operational semantics: An introduction*. *TCS* 412(38), pp. 5043–5069, doi:10.1016/j.tcs.2011.03.023.
- [16] B. Klin & B. Nachyła (2014): *Distributive Laws and Decidable Properties of SOS Specifications*. In Johannes Borgström & Silvia Crafa, editors: *Proc. EXPRESS/SOS 2014*, EPTCS 160, pp. 79–93, doi:10.4204/EPTCS.160.8.
- [17] B. Klin & B. Nachyła (2017): *Some undecidable properties of SOS specifications*. *J. Log. Algebr. Meth. Program.* 87, pp. 94–109, doi:10.1016/j.jlamp.2016.08.005.
- [18] J. Power & H. Watanabe (2002): *Combining a monad and a comonad*. *Theor. Comput. Sci.* 280(1-2), pp. 137–162, doi:10.1016/S0304-3975(01)00024-X.
- [19] J. Rot & M. M. Bonsangue (2016): *Structural congruence for bialgebraic semantics*. *J. Log. Algebr. Meth. Program.* 85(6), pp. 1268–1291, doi:10.1016/j.jlamp.2016.08.001.
- [20] J. J. M. Rutten (2000): *Universal coalgebra: a theory of systems*. *TCS* 249(1), pp. 3–80. Available at [http://dx.doi.org/10.1016/S0304-3975\(00\)00056-6](http://dx.doi.org/10.1016/S0304-3975(00)00056-6).
- [21] S. Staton (2008): *General Structural Operational Semantics through Categorical Logic*. In: *LICS*, IEEE Computer Society, pp. 166–177, doi:10.1109/LICS.2008.43.
- [22] D. Turi & G. Plotkin (1997): *Towards a Mathematical Operational Semantics*. In: *LICS*, IEEE Computer Society, pp. 280–291, doi:10.1109/LICS.1997.614955.