Acta Informatica manuscript No.
(will be inserted by the editor)

Distributive Laws for Monotone Specifications

Jurriaan Rot

Received: date / Accepted: date

Abstract 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 [30], 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,

The research leading to these results has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement nr. 320571.

Radboud University, Toernooiveld 212, Nijmegen, The Netherlands
E-mail: jrot@cs.ru.nl

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 [16,17,4].

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:

$$\frac{x \xrightarrow{a} x' \qquad y \xrightarrow{a} y'}{x \otimes y \xrightarrow{a} x' \otimes y'} \tag{1}$$

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 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 $\Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^*$, where Σ^* is the *free monad* for Σ (assuming it exists). If B is a functor that models (image-finite) transition systems, and Σ is a functor representing a signature, then such specifications correspond to classical GSOS specifications [30,4]. As opposed to the basic format, GSOS rules allow complex terms in conclusions, as in the following rule specifying a constant c:

$$\frac{}{c \xrightarrow{a} \sigma(c)} \tag{2}$$

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 $\Sigma B^\infty \Rightarrow B(\Sigma + \mathsf{Id})$, 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* [30]. A typical coGSOS rule is the following:

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

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 [30]. 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 [19].

The use of negative premises in the above example (and in [19]) is crucial. In the present paper, we introduce the notion of *monotonicity* of biGSOS specifications, generalising monotone abstract GSOS [9]. 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 [12] 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 current paper is an extended version of a conference paper presented at EXPRESS/SOS 2017 [24]. The main new material is proofs of all results, most of which were missing in the conference version; more backgroud material, in particular on biGSOS rule formats; a result showing that the constructed distributive law extends a given biGSOS specification in the sense of [18].

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 [17], 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 [18,19]. 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 [29]. 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 [22], but it does not include characterisations in terms of simpler natural transformations.

Structure of the paper Section 2 contains preliminaries on (co)algebras, and Section 3 recalls (abstract) rule formats based on various kinds of distributive laws. In Section 4 the notion of similarity on coalgebras is recalled, which is then used in Section 5 to define monotone specifications and prove the existence of least supported models. Section 6 contains our main result: canonical distributive laws for monotone biGSOS specifications. In Section 7, this result is extended to countably accessible functors.

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

2 Preliminaries

We recall the necessary definitions on order theory, algebras, coalgebras, and distributive laws of monads over comonads. For an introduction to coalgebra see [27, 14]. All of the definitions and results below and most of the examples can be found in [17], which provides an overview of bialgebraic semantics. Unless mentioned otherwise, all functors considered are endofunctors on the category Set of sets and functions.

Notation. By $\mathcal{P}$ we denote the (covariant) 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 $\pi_1 \colon R \to X$ and $\pi_2 \colon R \to Y$ 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 $R^{\mathsf{op}} = \{(y,x) \mid (x,y) \in R\}$. For a set X, we let $\Delta_X = \{(x,x) \mid x \in X\}$. The graph of a function $f \colon X \to Y$ is $\mathsf{Graph}(f) = \{(x, f(x)) \mid x \in X\}$. 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 $f, g \colon X \to Y$ we have $f \mathrel{R} g$ iff $f(x) \mathrel{R} g(x)$ for all $x \in X$, or, equivalently, $(f \times g)(\Delta_X) \subseteq R$. Composition of functors F, G (of the appropriate type) is denoted by $F \circ G$ or simply by FG, and composition of natural transformations α, β by $\alpha \circ \beta$, given by $(\alpha \circ \beta)_X = \alpha_X \circ \beta_X$.

DCPOs. We recall the basic order-theoretic structures that we will use in fixed point constructions; see, e.g., [7] for details. Let $(P, \leq)$ be a poset. Given a subset $S \subseteq P$ we denote the least upper bound of S by $\bigvee S$, if it exists. A subset $S \subseteq P$ is called *directed* if S is non-empty, and every finite subset of S has an upper bound in S. A *directed complete partial order (DCPO)* is a poset in which every directed subset has a least upper bound (these are sometimes referred to as CPOs, but we prefer DCPO as it emphasises the directedness aspect). Further, a DCPO is called *pointed* if it has a least element $\perp$. In this paper, every DCPO is assumed to be pointed, and thus will sometimes refer to a pointed DCPO simply as a DCPO. A map $f \colon P \to Q$ between pointed DCPOs is called *strict* if $f(\perp) = \perp$, and *continuous* if, for every directed set $S \subseteq P$, $f(S)$ is again directed, and $f(\bigvee S) = \bigvee f(S)$. We denote by $\mathsf{DCPO}_\perp$ the category of pointed DCPOs and strict continuous maps.

Every monotone map $f \colon P \to P$ on a pointed DCPO has a least fixed point μf (see, e.g., [7, Theorem 8.22]). Moreover, this can be computed (e.g., Exercise 8.19 in *loc. cit.*) via the ordinal-indexed chain defined for all α, λ with λ a limit ordinal by:

$$f^0(\perp) = \perp\,, \qquad f^{\alpha+1}(\perp) = f(f^\alpha(\perp))\,, \qquad f^\lambda(\perp) = \bigvee_{\beta<\lambda} f^\beta(\perp)\,.$$

This gives rise to a proof technique: to show that a predicate Φ holds for all $x \in \mu f$, it suffices to show that, if $x \in \Phi$ then $f(x) \in \Phi$, and if $S \subseteq \Phi$ is a directed set then $\bigvee S \in \Phi$.

2.1 Algebras and monads

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

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

We assume that a *free monad* (Σ^*, η, μ) for Σ exists; in the sequel we often refer to this monad by the underlying functor Σ^*. This means that there is a natural transformation $\iota \colon \Sigma\Sigma^* \Rightarrow \Sigma^*$ such that for any set X, the copairing

$$[\iota_X, \eta_X] \colon \Sigma\Sigma^* X + X \to \Sigma^* X$$

is an initial algebra for the functor $\Sigma + X$. By Lambek's lemma, $[\iota_X, \eta_X]$ is an isomorphism. Any algebra $f \colon \Sigma X \to X$ induces a $\Sigma + X$-algebra $[f, \mathsf{id}]$, and therefore by initiality a Σ^*-algebra $f^* \colon \Sigma^* X \to X$, which we call the *inductive extension* of f. In particular, the inductive extension of ι_X is μ_X. This construction preserves algebra homomorphisms: if h is a homomorphism from a Σ-algebra f to a Σ-algebra g, then it is also a homomorphism from f^* to g^*. It establishes (one side of) an isomorphism of categories $\mathsf{alg}(\Sigma) \cong \mathsf{Alg}(\Sigma^*)$, where the other direction (from right to left) is given by precomposing with the natural transformation κ in:

$$\kappa := \Big(\Sigma \xRightarrow{\Sigma\eta} \Sigma\Sigma^* \xRightarrow{\iota} \Sigma^* \Big) .$$

A *signature* is a functor $\Sigma \colon \mathbb{N} \to \mathsf{Set}$, where $\mathbb{N}$ is the discrete category of natural numbers. This models a countable collection of operator names with finite arity: for each n, $\Sigma(n)$ is the set of operator names of arity n. Any signature Σ gives rise to a polynomial Set endofunctor which, abusing notation, we also denote by Σ:

$$\Sigma X = \coprod_{n \in \mathbb{N}} \Sigma(n) \times X^n .$$

The free monad Σ^* constructs terms, that is, $\Sigma^* X$ is given by the grammar

$$t ::= \sigma(t_1, \ldots, t_n) \mid x$$

where x ranges over X and σ ranges over the operator names (and n is the arity of σ). 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 $f \colon X \to BX$. A *(coalgebra) homomorphism* from $f \colon X \to BX$ to $g \colon Y \to BY$ is a function $h \colon X \to Y$ such that $Bh \circ f = g \circ h$. The category of B-coalgebras and their homomorphisms is denoted by $\mathsf{coalg}(B)$. The associated forgetful functor is denoted by $U \colon \mathsf{coalg}(B) \to \mathsf{Set}$. An F-coalgebra (Z, z) is called *final* if it is a final object in $\mathsf{coalg}(B)$, i.e., if there exists a unique homomorphism from every F-coalgebra into (Z, z).

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

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

$$\langle \theta_X, \epsilon_X \rangle \colon B^\infty X \to (BB^\infty X) \times X$$

is a final coalgebra for the functor $B \times X$. Any coalgebra $f \colon X \to BX$ induces a $B \times X$-coalgebra $\langle f, \mathsf{id} \rangle$, and therefore by finality a B^∞-coalgebra $f^\infty \colon X \to B^\infty X$, which we call the *coinductive extension* of f. In particular, the coinductive extension of θ_X is δ_X. Similar to the case of algebras, this construction is functorial, and establishes an isomorphism $\mathsf{coalg}(B) \cong \mathsf{CoAlg}(B^\infty)$. From right to left, this is given by composing with the natural transformation ν in:

$$\nu := \Big(B^\infty \xRightarrow{\theta} BB^\infty \xRightarrow{B\epsilon} B \Big) .$$

Example 2.1 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., $A^\omega = \{\sigma \mid \sigma \colon \omega \to A\}$ where ω is the set of natural numbers. For a set X, $B^\infty X = (A \times X)^\omega$. Given $f \colon X \to A \times X$, its coinductive extension $f^\infty \colon X \to B^\infty X$ 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 2.2 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. For an LTS $f \colon X \to (\mathcal{P}X)^A$, we sometimes write $x \xrightarrow{a} x'$ if $x' \in f(x)(a)$. A coalgebra homomorphism from an LTS X to an LTS Y is a map $h \colon X \to Y$ such that, for all $x \in X$ and $a \in A$:

- if $x \xrightarrow{a} x'$ then $h(x) \xrightarrow{a} h(x')$, and
- if $h(x) \xrightarrow{a} y'$ then there is x' such that $x \xrightarrow{a} x'$ and $h(x') = y'$.

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 (equivalence classes 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. The counit ϵ_X maps such a tree to its root.

Example 2.3 A *complete monoid* is a (necessarily commutative) monoid M together with an infinitary sum operation consistent with the finite sum [8]. Define the functor $\mathcal{M} \colon \mathsf{Set} \to \mathsf{Set}$ by $\mathcal{M}(X) = \{\varphi \mid \varphi \colon X \to M\}$ and, for $f \colon X \to Y$, $\mathcal{M}(h)(\varphi) = \lambda y. \sum_{x \in f^{-1}(y)} \varphi(x)$. A *weighted transition system* over a set of labels A is a coalgebra $f \colon X \to (\mathcal{M}X)^A$. 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 $\mathcal{M}_c(X) = \{\varphi \colon X \to M \mid \varphi(x) \neq 0 \text{ for countably many } x \in X\}$. 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., [16]). 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 Distributive laws

In Section 3, we will use distributive laws as a common generalisation of several abstract rule formats. For now, we only recall a few basic definitions and properties.

A *distributive law* of a monad $\mathcal{T} = (T, \eta, \mu)$ over a comonad $\mathcal{D} = (D, \epsilon, \delta)$ is a natural transformation $\lambda\colon TD \Rightarrow DT$ making the following diagrams commute.

$$\begin{tikzcd}
TTD \arrow[r, Rightarrow, "T\lambda"] \arrow[d, Rightarrow, "\mu D"'] & TDT \arrow[r, Rightarrow, "\lambda T"] & DTT \arrow[d, Rightarrow, "D\mu"] \\
TD \arrow[rr, Rightarrow, "\lambda"] \arrow[d, Rightarrow, "T\delta"'] & & DT \arrow[d, Rightarrow, "\delta T"] \\
TDD \arrow[r, Rightarrow, "\lambda D"] & DTD \arrow[r, Rightarrow, "D\lambda"] & DDT
\end{tikzcd}
\qquad
\begin{tikzcd}
D \arrow[d, Rightarrow, "\eta D"'] \arrow[dr, Rightarrow, "D\eta"] & \\
TD \arrow[r, Rightarrow, "\lambda"] \arrow[d, Rightarrow, "T\epsilon"'] & DT \arrow[dl, Rightarrow, "\epsilon T"] \\
T &
\end{tikzcd}$$

The axioms ensure compatibility with the monad and comonad structure. For more details of their use in SOS, see [17].

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

$$\begin{tikzcd}
\mathsf{CoAlg}(\mathcal{D}) \arrow[r, "\overline{T}"] \arrow[d, "U"'] & \mathsf{CoAlg}(\mathcal{D}) \arrow[d, "U"] \\
\mathsf{Set} \arrow[r, "T"] & \mathsf{Set}
\end{tikzcd}$$

where the vertical arrows are the forgetful functor. Further, a monad $(\overline{T}, \overline{\eta}, \overline{\mu})$ on $\mathsf{CoAlg}(\mathcal{D})$ 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 $\mathsf{coalg}(B)$ is defined similarly. Distributive laws of $\mathcal{T}$ over $\mathcal{D}$ are in one-to-one correspondence with liftings of (T, η, μ) to $\mathsf{CoAlg}(\mathcal{D})$ [15,30]. If $\mathcal{D}$ is the cofree comonad for B, then $\mathsf{CoAlg}(\mathcal{D}) \cong \mathsf{coalg}(B)$, hence a further equivalent condition is that $\mathcal{T}$ lifts to $\mathsf{coalg}(B)$.

Given a distributive law $\lambda\colon TD \Rightarrow DT$, a λ-*bialgebra* is a triple (X, a, f) where X is a set, a is an EM-algebra for $\mathcal{T}$ and f is an EM-coalgebra for $\mathcal{D}$, such that the following diagram commutes:

$$\begin{tikzcd}
TX \arrow[r, "a"] \arrow[d, "Tf"] & X \arrow[r, "f"] & DX \\
TDX \arrow[rr, "\lambda_X"] & & DTX \arrow[u, "Da"']
\end{tikzcd}$$

Any distributive law λ induces, by initiality, a unique coalgebra $h\colon T\emptyset \to DT\emptyset$ such that $(T\emptyset, \mu_\emptyset, h)$ is λ-bialgebra, i.e., making the diagram on the left below commute:

$$\begin{tikzcd}
TT\emptyset \arrow[r, "\mu_\emptyset"] \arrow[d, "Th"] & T\emptyset \arrow[r, "h"] & DT\emptyset \\
TDT\emptyset \arrow[rr, "\lambda_{T\emptyset}"] & & DTT\emptyset \arrow[u, "D\mu_\emptyset"']
\end{tikzcd}
\qquad
\begin{tikzcd}
TD1 \arrow[r, "k"] \arrow[d, "T\delta_1"] & D1 \arrow[r, "\delta_1"] & DD1 \\
TDD1 \arrow[rr, "\lambda_{D1}"] & & DTD1 \arrow[u, "Dk"']
\end{tikzcd}$$

Similarly, there is a unique algebra structure $k\colon TD1 \to D1$ (where 1 is a singleton, so that $D1$ is the carrier of a final coalgebra) making the diagram on the right commute.

The map h is a coalgebra for the comonad D. If D is the cofree comonad B^∞ of a functor B, then h corresponds to a B-coalgebra; we refer to the latter as the *operational model* of λ.

The coalgebra $h\colon T\emptyset \to DT\emptyset$ plays a special role. As we will see in the next section, in the context of certain rule formats on, e.g., labelled transition systems, it is the labelled transition system over closed terms induced by a given specification. In this context, the semantics arises by finality, as the unique coalgebra morphism $s\colon T\emptyset \to D1$ in the diagram on the left:

$$\begin{tikzcd}
T\emptyset \arrow[r, dashed, "s"] \arrow[d, "h"] & D1 \arrow[d, "\delta_1"] \\
DT\emptyset \arrow[r, dashed, "Ds"] & DD1
\end{tikzcd}
\qquad
\begin{tikzcd}
TT\emptyset \arrow[r, "Ts"] \arrow[d, "\mu_\emptyset"] & TD1 \arrow[d, "Tk"] \\
T\emptyset \arrow[r, "s"] & D1
\end{tikzcd}$$

Dually, the algebra k is the interpretation of the operations defined in the specification on the final coalgebra. It is easy to show that the same map s (defined as the unique map to a final coalgebra) is also an algebra morphism from the initial algebra (as on the right above). This means that behavioural equivalence on h, i.e., identification of elements by the 'semantics map' s, is a congruence (since s is an algebra morphism). For a more extensive explanation, see [17,30].

3 Abstract rule formats: GSOS, coGSOS and biGSOS

In this section we recall the abstract rule formats originally introduced in [30], and their combination, called biGSOS in [18].

3.1 GSOS

Probably the most important instance of these formats is *abstract GSOS*, which generalises the classical GSOS format for labelled transition systems [5]. An abstract GSOS specification [30] is a natural transformation of the form

$$\rho\colon \Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^* \, .$$

This notion gives rise to concrete rule formats for different types of coalgebras, by varying the functor B. We recall this for streams and labelled transition systems.

Example 3.1 Given a signature Σ (Section 2.1), a *stream GSOS* rule [17,11] for an operator σ of arity n is of the form

$$\frac{x_1 \xrightarrow{a_1} x_1' \qquad \dots \qquad x_n \xrightarrow{a_n} x_n'}{\sigma(x_1, \dots x_n) \xrightarrow{b} t}$$

where $x_1, \dots, x_n, x_1', \dots, x_1'$ are pairwise distinct variables; t is a term over these variables and the signature Σ; and $a_1, \dots, a_n, b \in \mathbb{R}$.

A stream GSOS specification is a collection of stream GSOS rules over a common signature such that, for every operator σ (of arity n) and every sequence $(a_1, \dots, a_n) \in \mathbb{R}^n$, there is exactly one rule with premises $x_1 \xrightarrow{a_1} x_1'$, $\dots$, $x_n \xrightarrow{a_n} x_n'$.

Let $BX = \mathbb{R} \times X$, and let Σ be a signature. Every stream GSOS specification gives rise to an abstract GSOS specification $\Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^*$ involving these functors, and conversely, every abstract GSOS specification arises in this way [17]. Rather than recalling the exact construction, we consider an example of basic operations on streams: a binary sum $\oplus$, (shuffle) product $\otimes$ and a constant $\overline{r}$ for each $r \in \mathbb{R}$. The following stream GSOS rules define their semantics (corresponding to the usual behavioural differential equations [26]):

$$\frac{x \xrightarrow{r} x' \qquad y \xrightarrow{s} y'}{x \oplus y \xrightarrow{r+s} x' \oplus y'} \qquad \frac{x \xrightarrow{r} x' \qquad y \xrightarrow{s} y'}{x \otimes y \xrightarrow{r \times s} (x' \otimes y) \oplus (x \otimes y')} \qquad \frac{}{\overline{r} \xrightarrow{r} \overline{0}}$$

To model this as an abstract GSOS specification, we take $\Sigma X = (X \times X) + (X \times X) + \mathbb{R}$, and denote the left coproduct injection of a pair (x, y) by $x \oplus y$, the middle by $x \otimes y$ and the right (for an element $r \in \mathbb{R}$) by $\overline{r}$. Then the natural transformation $\rho\colon \Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^*$ (with $BX = \mathbb{R} \times X$) corresponding to the above specification is given on a component X by

$$\begin{aligned} \rho_X((x, r, x') \oplus (y, s, y')) &= (r + s, (x' \oplus y')) \\ \rho_X((x, r, x') \otimes (y, s, y')) &= (r \times s, (x' \otimes y) \oplus (x \otimes y')) \\ \rho_X(\overline{r}) &= (r, \overline{0}) \end{aligned}$$

Example 3.2 Given a signature Σ , a *GSOS* rule [5] for an operator σ of arity n is of the form

$$\frac{\{x_{i_j} \xrightarrow{a_j} y_j\}_{j=1..m} \qquad \{x_{i_k} \overset{b_k}{\not\to}\}_{k=1..l}}{\sigma(x_1, \ldots, x_n) \xrightarrow{c} t} \tag{4}$$

where m and l are the number of positive and negative premises respectively; $a_1, \ldots, a_m, b_1, \ldots, b_l, c \in A$ are labels; $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ are pairwise distinct variables, and t is a term over these variables and the signature Σ.

A GSOS *specification* is a collection of rules satisfying an image-finiteness condition. As first observed in [30], and proved in detail in [4], specifications in the GSOS format are generalised by abstract GSOS specifications, where Σ models the signature and $BX = (\mathcal{P}_f X)^A$.

A *model* of an abstract GSOS specification $\rho\colon \Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^*$ is a triple (X, a, f) where X is a set, $a\colon \Sigma X \to X$ an algebra and $f\colon X \to BX$ a coalgebra, such that the diagram on the left below commutes:

$$\begin{tikzcd} \Sigma X \arrow[r, "a"] \arrow[d, "\Sigma\langle f, \mathsf{id}\rangle"] & X \arrow[r, "f"] & BX \\ \Sigma(B \times \mathsf{Id})X \arrow[rr, "\lambda_X"] & & B\Sigma^* X \arrow[u, "Ba^*"'] \end{tikzcd} \qquad \begin{tikzcd} \Sigma\Sigma^*\emptyset \arrow[r, "\iota_\emptyset"] \arrow[d, "\Sigma\langle f, \mathsf{id}\rangle"] & \Sigma^*\emptyset \arrow[r, "f"] & B\Sigma^*\emptyset \\ \Sigma(B \times \mathsf{Id})\Sigma^*\emptyset \arrow[rr, "\lambda_{\Sigma^*\emptyset}"] & & B\Sigma^*\Sigma^*\emptyset \arrow[u, "B\mu_\emptyset"'] \end{tikzcd}$$

Of particular interest are models on the initial algebra, i.e., where the algebra part is given by $\iota_\emptyset\colon \Sigma\Sigma^*\emptyset \to \Sigma^*\emptyset$. It turns out that, in this case, there is a *unique* coalgebra structure f turning $(\Sigma^*\emptyset, \iota_\emptyset, f)$ into a model, as depicted on the right above. We call this the *supported model* of the abstract GSOS specification ρ, since it generalises the usual notion of supported model for labelled transition systems, informally stating that a transition $\sigma(t_1, \dots, t_n) \xrightarrow{a} t'$ is in the model iff it is provable by the rules.

The uniqueness of the supported model follows from the theory of distributive laws, using that every abstract GSOS specification ρ induces a distributive law $\lambda \colon \Sigma^* B^\infty \Rightarrow B^\infty \Sigma^*$ of the free monad over the cofree comonad [20,17]. It also follows from this correspondence that behavioural equivalence on the supported model of ρ is a congruence.

3.2 coGSOS

The 'dual' of GSOS is coGSOS, introduced in [30]. It is used a lot less than GSOS, since it appears most examples (such as the standard operators of the stream calculus [26]) fit in the GSOS format and not in the coGSOS format. A *coGSOS specification* is a natural transformation of the form

$$\rho \colon \Sigma B^\infty \Rightarrow B(\Sigma + \mathsf{Id})\,.$$

In case of labelled transition systems, these natural transformations can be obtained from specifications in the *safe ntree* format, which we will see as a special case of a more general situation below (Example 3.6).

Example 3.3 In case of streams (i.e., $BX = \mathbb{R} \times X$), coGSOS specifications can be presented concretely as a rule format which allows lookahead in premises, but where the term t, the right-hand side of the conclusion, can only be a single operation, or a variable. Concretely, a *stream coGSOS* rule [17] for an operator σ of arity n is of the form

$$\frac{\{x_1^i \xrightarrow{a_1^i} x_1^{i+1}\}_{i \in \mathbb{N}_{>0}} \qquad \dots \qquad \{x_n^i \xrightarrow{a_n^i} x_n^{i+1}\}_{i \in \mathbb{N}_{>0}}}{\sigma(x_1, \dots x_n) \xrightarrow{b} t}$$

where $\mathbb{N}_{>0}$ is the set of positive natural numbers, $x_1^1, \dots, x_n^1, x_1^2, \dots, x_1^2, \dots$ are pairwise distinct variables; t is either a single operator over these variables or a single variable drawn from this set; and $b, a_1^1, \dots, a_n^1, a_1^2, \dots, a_n^2, \dots \in \mathbb{R}$.

Note that coGSOS specifications allow infinitely many premises; in particular, one can inspect the entire argument stream in the premise. This is operationally rather questionable, but for practicular purposes, one may give rules which only mention a finite number of premises for each argument. For instance, consider the following rule:

$$\frac{x \xrightarrow{a} x' \qquad x' \xrightarrow{a'} x''}{\mathsf{even}(x) \xrightarrow{a} \mathsf{even}(x'')}$$

It is straightforward to extend this to a collection of rules in the coGSOS format, by extending the premises to infinite sequences in all possible ways; see [17] for details.

The notion of (supported) model of a coGSOS specification is defined similarly (dually) to that of a GSOS specification; we will see a more definition that generalises both in the context of the biGSOS format, introduced next. Just like the GSOS case, coGSOS specifications have unique supported models, on which behavioural equivalence is a congruence [17].

3.3 biGSOS

The abstract GSOS and coGSOS formats have a straightforward common generalisation, called *biGSOS* in [18]. This notion, as well as the associated notion of model which we define later, subsumes both abstract GSOS and coGSOS.

Definition 3.4 A *biGSOS specification* is a natural transformation of the form

$$\rho\colon \Sigma B^{\infty} \Rightarrow B\Sigma^{*} .$$

Example 3.5 For streams (i.e., $BX = \mathbb{R} \times X$), a concrete presentation of the biGSOS format is very similar to the (coGSOS) format given in Example 3.3; however, the term t on the right-hand side of the conclusion of a rule is now allowed to be an arbitrary term over the variables introduced in the premises, rather than just a single operator or variable.

Of course, all GSOS and coGSOS rules form examples of biGSOS rules. As pointed out in [4], a nice example that fits in the biGSOS format but not in one of the other formats, is the Δ_c operator from [28], called Laplace-Carson transform (on the left below):

$$\frac{x \xrightarrow{r} x' \quad x' \xrightarrow{r'} x''}{\Delta_c(x) \xrightarrow{r} \Delta_c(x' \oplus (X \otimes x''))} \qquad \frac{}{X \xrightarrow{0} \overline{1}}$$

It makes use of the shuffle product, sum (Example 3.1) and the operator X (defined above, using the constant $\overline{1}$), so together with those operations it forms a biGSOS specification.

Example 3.6 If $BX = (\mathcal{P}_f X)^A$, then one can obtain biGSOS specifications from concrete rules in the *ntree* format. An *ntree* rule (as taken from [17]) for an operator σ with arity n is of the form

$$\frac{\{z_i \xrightarrow{a_i} y_i\}_{i\in I} \qquad \{w_j \overset{b_j}{\not\rightarrow}\}_{j\in J}}{\sigma(x_1, \ldots, x_n) \xrightarrow{c} t} \tag{5}$$

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 a term over the variables. An ntree rule is called *safe* if t is either a single variable or a term built of a single operator from the signature and the variables.

As stated in [30], every ntree specification (a collection of ntree rules with a certain image-finiteness condition) induces a biGSOS specification where Σ models the signature and $BX = (\mathcal{P}_f X)^A$. Safe ntree rules induce coGSOS specifications. The construction of a biGSOS specification ρ from an ntree specification is sketched in [30]. Here, we fill in some of the details. Note that it is not known whether there is a converse, i.e., whether every biGSOS specification is derived from an ntree specification [17].

First, note that by the conditions imposed on ntree rules (5), the premises give rise to collection of n disjoint trees with roots $x_1, \ldots, x_n$ respectively, edge-labelled in A and node-labelled in variables, with a unique label for each node. Let V be the set of variables that occur in the nodes of these trees. Given $b \in B^{\infty}X$ and a map $h\colon V \to B^{\infty}X$, we write $x_i \models_h b_i$ if

1. $h(x_i) = b_i$, and
2. if $x \xrightarrow{a} x'$ (in the rooted tree induced by the premises) then $h(x) \xrightarrow{a} h(x')$, and
3. if $x \stackrel{a}{\not\to}$ (an explicit negative premise in the rule) then $h(x) \stackrel{a}{\not\to}$ (no a-transition from $h(x)$).

Now, $\rho\colon \Sigma B^\infty \Rightarrow (\mathcal{P}_f \Sigma^* X)^A$ (for $BX = (\mathcal{P}_f X)^A$) is defined as follows: $u \in \rho_X(\sigma(b_1, \ldots, b_n))(a)$ iff $u = \Sigma^*(\epsilon_X \circ h)(t)$ for some $t \in \Sigma^* X$ and h as above such that there is a rule of the form (5) with for each i, $x_i \models_h b_i$. We prove naturality in Appendix A.

A triple (X, a, f) consisting of a set X, an algebra $a\colon \Sigma X \to X$ and a coalgebra $f\colon X \to BX$ (i.e., a bialgebra) is called a *ρ-model* if the diagram on the left below commutes.

$$\begin{tikzcd}
\Sigma X \arrow[r, "a"] \arrow[d, "\Sigma f^\infty"] & X \arrow[r, "f"] & BX & & \Sigma\Sigma^*\emptyset \arrow[r, "\iota_\emptyset"] \arrow[d, "\Sigma f^\infty"] & \Sigma^*\emptyset \arrow[r, "f"] & B\Sigma^*\emptyset \\
\Sigma B^\infty X \arrow[rr, "\lambda_X"] & & B\Sigma^* X \arrow[u, "Ba^*"] & & \Sigma B^\infty \Sigma^*\emptyset \arrow[rr, "\lambda_{\Sigma^*\emptyset}"] & & B\Sigma^*\Sigma^*\emptyset \arrow[u, "B\mu_\emptyset"]
\end{tikzcd} \tag{6}$$

In particular, we define a *supported* model as a ρ-model on the initial algebra, i.e., a coalgebra $f\colon \Sigma^*\emptyset \to B\Sigma^*\emptyset$ such that the diagram on the right above commutes.

Contrary to the case of GSOS and coGSOS, in general there is no (unique) supported model. This is witnessed, for instance, by the example specification in the introduction. In Example 3.7, we shall see that biGSOS specifications may also have multiple supported models, and that behavioural equivalence is not even a congruence, in general.

Example 3.7 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:

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

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, on the set of closed terms. We are interested in the *least* model, since that 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, as we will see in the next sections.

Every GSOS and coGSOS specification induces a distributive law. Using this, one can prove that such specifications have unique supported models on which behavioural equivalence is a congruence. The above examples of biGSOS specifications, where some do not have a model and some have multiple models, suggest that the correspondence with distributive laws does not go through for biGSOS specifications. To make this slightly more precise, we say a distributive law $\lambda\colon \Sigma^* B^\infty \Rightarrow B^\infty \Sigma^*$ of monad over comonad *extends* a biGSOS specification $\rho\colon \Sigma B^\infty \Rightarrow B\Sigma^*$ if the following diagram commutes [18]:

$$\begin{tikzcd}
\Sigma B^\infty \arrow[r, Rightarrow, "\rho"] \arrow[d, Rightarrow, "\kappa B^\infty"'] & B\Sigma^* \\
\Sigma^* B^\infty \arrow[r, Rightarrow, "\lambda"'] & B^\infty \Sigma^* \arrow[u, Rightarrow, "\nu \Sigma^*"']
\end{tikzcd}$$

Klin and Nachyła [18] show that, in case of stream systems or labelled transition systems, it is undecidable whether a given biGSOS specification ρ extends (uniquely) to a distributive law λ. In fact, not even the full power of biGSOS is used there: every individual rule is required to be either in the GSOS or in the coGSOS format.

The current paper takes a different approach, by disallowing negative premises—which are crucial in the proof of Klin and Nachyła. Further, we will construct a canonical distributive law that extends a given biGSOS specification (satisfying a monotonicity condition generalising absence of negative premises), but we make no claims about uniqueness.

4 Similarity on cofree coalgebras

In this section, we recall the notion of *simulation* of coalgebras from [12], and prove a few basic results concerning the similarity preorder on final coalgebras. This similarity preorder will be used in Section 5 to formulate what it means for a biGSOS specification to be monotone.

Throughout this section, let $B\colon \mathsf{Set} \to \mathsf{Set}$ be a functor for which a cofree coalgebra B^∞ exists. The (canonical) *relation lifting* of B is defined on a relation $R \subseteq X \times Y$ by

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

It thus extends a relation $R \subseteq X \times Y$ to a relation $\mathsf{Rel}(B)(R) \subseteq BX \times BY$, by taking the image of BR along the pairing $\langle B\pi_1, B\pi_2\rangle\colon BR \to BX \times BY$. For instance, for the functor $B(X) = A \times X$, we have

$$\mathsf{Rel}(B)(R) = \{((a, x), (a, y)) \mid a \in A \text{ and } (x, y) \in R\}\,.$$

Relation lifting can be used to define a notion of bisimulation, which we recall, as the notion of simulation that we use later is a variation on it. Let (X, f) and (Y, g) be B-coalgebras. A relation $R \subseteq X \times X$ is a *bisimulation* if $R \subseteq (f \times g)^{-1}(\mathsf{Rel}(B)(R))$. For a detailed account of bisimulations and relation lifting, see, e.g., [13]. We recall the following standard properties, which will be useful later.

Lemma 4.1 *For any functor $F\colon \mathsf{Set} \to \mathsf{Set}$ and any $R \subseteq X \times X'$, $S \subseteq Y \times Y'$:*

1. $\mathsf{Rel}(F)(\Delta_X) = \Delta_{FX}$.
2. *For any $R' \subseteq X \times X'$: if $R \subseteq R'$ then $\mathsf{Rel}(F)(R) \subseteq \mathsf{Rel}(F)(R')$.*
3. *For any $f\colon X \to Y$, $g\colon X' \to Y'$: $(Ff \times Fg)(\mathsf{Rel}(F)(R)) \subseteq \mathsf{Rel}(F)((f \times g)(R))$.*
4. $\mathsf{Rel}(F)(R)^{\mathsf{op}} = \mathsf{Rel}(F)(R^{\mathsf{op}})$.
5. *If F preserves weak pullbacks, then for any $f\colon X \to Y$: $\mathsf{Rel}(F)(\mathsf{Graph}(f)) = \mathsf{Graph}(Ff)$.*
6. *For any $f\colon X \to Y$, $g\colon X' \to Y'$: if $f\ S\ g$ then $F(f)\ \mathsf{Rel}(F)(S)\ F(g)$,*

Proof The first five properties are standard (see, e.g., [13]). For (6), by assumption we have $(f \times g)(\Delta_X) \subseteq S$. By (2) this implies $\mathsf{Rel}(F)((f \times g)(\Delta_X)) \subseteq \mathsf{Rel}(F)(S)$. Further, by (1) and (3), we have $(Ff \times Fg)(\Delta_{FX}) = (Ff \times Fg)(\mathsf{Rel}(F)(\Delta_X)) \subseteq \mathsf{Rel}(F)((f \times g)(\Delta_X))$. Thus $(Ff \times Fg)(\Delta_{FX}) \subseteq \mathsf{Rel}(F)(S)$ as desired. □

An *ordered functor* is a pair $(B, \sqsubseteq)$ of functors $B\colon \mathsf{Set} \to \mathsf{Set}$ and $\sqsubseteq\colon \mathsf{Set} \to \mathsf{PreOrd}$ such that

$$\begin{tikzcd}
 & \mathsf{PreOrd} \arrow[d] \\
\mathsf{Set} \arrow[ur, "\sqsubseteq"] \arrow[r, "B"'] & \mathsf{Set}
\end{tikzcd}$$

commutes, where the arrow from PreOrd to Set is the forgetful functor mapping a preorder to its carrier. Concretely, this means:

- For any set X, there is a preorder $\sqsubseteq_{BX} \subseteq BX \times BX$;
- For any map $f\colon X \to Y$, Bf is monotone, i.e., $b \sqsubseteq_{BX} c$ implies $Bf(b) \sqsubseteq_{BY} Bf(c)$.

Let $(B, \sqsubseteq)$ be an ordered functor. The *lax relation lifting* $\mathsf{Rel}_{\sqsubseteq}$ is defined as follows:

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

Let (X, f) and (Y, g) be B-coalgebras. A relation $R \subseteq X \times Y$ is a *simulation* (between f and g) if $R \subseteq (f \times g)^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)(R))$. 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 $(B \times X, \widetilde{\sqsubseteq})$ by

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

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

Lemma 4.2 *Let $\lesssim$ be the similarity relation between coalgebras $\langle f, f'\rangle\colon X \to BX \times Z$ and $\langle g, g'\rangle\colon X \to BX \times Z$. Then for any relation $R \subseteq X \times X$:*

$$R \subseteq (\langle f, f'\rangle \times \langle g, g'\rangle)^{-1}(\mathsf{Rel}_{\widetilde{\sqsubseteq}}(B \times Z)(R))$$

iff $R \subseteq (f \times g)^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)(R))$ and for all $(x, y) \in R$: $f'(x) = g'(x)$.

Proof We compute a more concrete presentation of $\mathsf{Rel}_{\widetilde{\sqsubseteq}}(B \times Z)(R)$:

$$\begin{aligned}\mathsf{Rel}_{\widetilde{\sqsubseteq}}(B \times Z)(R) &= \widetilde{\sqsubseteq} \circ \mathsf{Rel}(B \times Z)(R) \circ \widetilde{\sqsubseteq} \\ &= \widetilde{\sqsubseteq} \circ \{((B \times Z)(\pi_1)(z), (B \times Z)(\pi_2)(z)) \mid z \in (B \times Z)(R)\} \circ \widetilde{\sqsubseteq} \\ &= \widetilde{\sqsubseteq} \circ \{((B\pi_1(u), v), (B\pi_2(u), v)) \mid u \in BR,\ v \in Z\} \circ \widetilde{\sqsubseteq} \\ &= \widetilde{\sqsubseteq} \circ \{((b, v), (c, v)) \mid (b, c) \in \mathsf{Rel}(B)(R),\ v \in Z\} \circ \widetilde{\sqsubseteq} \\ &= \{((b, v), (c, v)) \mid (b, c) \in \mathsf{Rel}_{\sqsubseteq}(B)(R),\ v \in Z\}\,.\end{aligned}$$

The desired equivalence follows easily from this presentation. □

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

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

for the similarity order induced by the ordered functor $(B \times X, \widetilde{\sqsubseteq})$ on the cofree coalgebra $(B^\infty X, \langle \theta_X, \epsilon_X \rangle)$. We discuss a few examples of ordered functors and similarity; see [12] for many more.

Example 4.3 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 R between the underlying trees of p and q, so that pairs in R agree on labels in the set X. Concretely, for all $p, q \in R$:

- $\epsilon_X(p) = \epsilon_X(q)$, and
- if $p \xrightarrow{a} p'$ then there is q' such that $q \xrightarrow{a} q'$ and $(p', q') \in R$.

Example 4.4 For any $G\colon \mathsf{Set} \to \mathsf{Set}$, the functor $B = G + 1$, where $1 = \{\bot\}$, can be ordered as follows: $x \leq y$ iff $x = \bot$ or $x = y$, for all $x, y \in BX$. If $G = A \times \mathsf{Id}$ then $B^\infty X$ consists of finite and infinite sequences of the form $x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} x_2 \xrightarrow{a_2} \ldots$ with $x_i \in X$ and $a_i \in A$ for each i (cf. Example 2.1). For $\sigma, \tau \in B^\infty X$ we have $\sigma \lesssim_{B^\infty X} \tau$ if τ does not terminate before σ does, and σ and τ agree on labels in X and A on each position where σ is defined.

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.

Lemma 4.5 *Coalgebra homomorphisms h, k preserve similarity: if $x \lesssim y$ then $h(x) \lesssim k(y)$.*

Proof Let $h\colon X \to X'$ be a coalgebra homomorphism from (X, f) to (X', f'), and $k\colon Y \to Y'$ a coalgebra homomorphism from (Y, g) to (Y', g'). In [12], it is shown that $x \lesssim y$ iff $h(x) \lesssim k(y)$ under the assumption of a stable order (meaning that $\mathsf{Rel}_{\sqsubseteq}(B)$ is a fibred functor); but this is not necessary for the implication from left to right. In fact, the desired implication follows from the abstract theory of bifibrations, see, e.g., [23, Proposition 3.3.7].

For a more concrete proof, we show that the direct image of a simulation under $(h \times k)$ is again a simulation. To this end, let $R \subseteq (f \times g)^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)(R))$. Then

$$(h \times k)(R) \subseteq (h \times k)((f \times g)^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)(R)))$$

and the result follows (i.e., $(h \times k)(R)$ is a simulation) once we show that

$$(h \times k)((f \times g)^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)(R))) \subseteq (f' \times g')^{-1}(\mathsf{Rel}_{\sqsubseteq}(B)((h \times k)(R)))\,.$$

The latter inclusion in turn follows from the following two facts:

- $(h \times k)(f \times g)^{-1} \subseteq (f' \times g')^{-1}(Bh \times Bk)$ (pointwise inclusion). This is a straightforward computation: let $S \subseteq BX \times BY$, and suppose $(f(x), g(y)) \in S$; we need to prove that $(h(x), h(y)) \in (f' \times g')^{-1}(Bh \times Bk)(S)$. Indeed, we have $((Bh)(f(x)), (Bk)(g(y))) \in (Bh \times Bk)(S)$, so we have $(f'(h(x)), k'(g(y))) \in (Bh \times Bk)(S)$ (since h and k are homomorphisms), as desired.
- $(Bh \times Bk)(\mathsf{Rel}_{\sqsubseteq}(B)(R)) \subseteq \mathsf{Rel}_{\sqsubseteq}(B)((h \times k)(R))$ is shown concretely in [12, Lemma 4.2].

□

Pointwise inequality of coalgebras implies pointwise similarity of coinductive extensions:

Lemma 4.6 *Let $(B, \sqsubseteq)$ be an ordered functor, and let f and g be B-coalgebras on a common carrier X. If $f \sqsubseteq_{BX} g$ then $f^{\infty} \lesssim_{B^{\infty}X} g^{\infty}$.*

Proof Suppose $f \sqsubseteq_{BX} g$. This is equivalent to $(f \times g)(\Delta_X) \subseteq \sqsubseteq_{BX}$. Since $\sqsubseteq_{BX}$ is reflexive, we have $\sqsubseteq_{BX} \subseteq (\sqsubseteq_{BX} \circ \Delta_{BX} \circ \sqsubseteq_{BX})$. Further $\Delta_{BX} = \mathsf{Rel}(B)(\Delta_X)$ by Lemma 4.1 (1), so that $\sqsubseteq_{BX} \subseteq (\sqsubseteq_{BX} \circ \mathsf{Rel}(B)(\Delta_X) \circ \sqsubseteq_{BX}) = \mathsf{Rel}_{\sqsubseteq}(B)(\Delta_X)$. Combined with the assumption, this yields $(f \times g)(\Delta_X) \subseteq \mathsf{Rel}_{\sqsubseteq}(B)(\Delta_X)$, so Δ_X is a simulation between f and g. By Lemma 4.2 this implies that $x \lesssim x$ for any $x \in X$, where $\lesssim$ is the similarity order between $(X, \langle f, \mathsf{id}\rangle)$ and $(X, \langle g, \mathsf{id}\rangle)$ induced by $(B \times X, \widetilde{\sqsubseteq})$. Notice that f^{∞} and g^{∞} are coalgebra homomorphisms from $\langle f, \mathsf{id}\rangle$ and $\langle g, \mathsf{id}\rangle$ respectively into the cofree coalgebra $(B^{\infty}X, \langle \theta, \epsilon\rangle)$, so by Lemma 4.5 we have $(f^{\infty} \times g^{\infty})(\Delta_X) \subseteq \lesssim_{B^{\infty}X}$. □

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

Lemma 4.7 *Let $(B, \sqsubseteq)$ be an ordered functor where B preserves weak pullbacks, and let $f \colon X \to BX$, $g \colon Y \to BY$ and $h \colon X \to Y$.*

1. *If $Bh \circ f \sqsubseteq_{BY} g \circ h$ then $B^{\infty}h \circ f^{\infty} \lesssim_{B^{\infty}Y} g^{\infty} \circ h$.*
2. *If $Bh \circ f \sqsupseteq_{BY} g \circ h$ then $B^{\infty}h \circ f^{\infty} \gtrsim_{B^{\infty}Y} g^{\infty} \circ h$.*

Proof The assumption in item 1. is equivalent to: $(f(x), g(h(x)) \in (\mathsf{Graph}(Bh) \circ \sqsubseteq_{BY})$ for all $x \in X$. Thus $(f(x), g(h(x)) \in (\sqsubseteq_{BX} \circ \mathsf{Graph}(Bh) \circ \sqsubseteq_{BY})$ since $\sqsubseteq_{BX}$ is reflexive. We have $\mathsf{Graph}(Bh) = \mathsf{Rel}(B)(\mathsf{Graph}(h))$ by Lemma 4.1 (5) and the assumption that B preserves weak pullbacks, so by the above, $\mathsf{Graph}(h)$ is a simulation between f and g. By Lemma 4.2 it follows that $x \lesssim h(x)$, where $\lesssim$ is the similarity order between $\langle f, h\rangle \colon X \to BX \times Y$ and $\langle g, \mathsf{id}\rangle \colon Y \to BY \times Y$ induced by $(B \times Y, \widetilde{\sqsubseteq})$.

Next, observe that from the following commutative diagram it follows that $B^\infty h \circ f^\infty$ is a $B \times Y$-coalgebra homomorphism:

$$\begin{tikzcd}
X \arrow[r, "f^\infty"] \arrow[d, "\langle f, \mathsf{id}\rangle"'] & B^\infty X \arrow[r, "B^\infty h"] \arrow[d, "\langle \theta_X, \epsilon_X\rangle"] & B^\infty Y \arrow[d, "\langle \theta_Y, \epsilon_Y\rangle"] \\
BX \times X \arrow[r, "Bf^\infty \times \mathsf{id}"] \arrow[d, "B\mathsf{id}\times h"'] & BB^\infty X \times X \arrow[r, "BB^\infty h \times h"] \arrow[d, "\mathsf{id}\times h"'] & BB^\infty Y \times Y \\
BX \times Y \arrow[r, "Bf^\infty \times \mathsf{id}"'] & BB^\infty \times Y \arrow[ur, "BB^\infty h \times \mathsf{id}"'] &
\end{tikzcd}$$

The upper left square commutes by definition of f^∞, the upper right by naturality of θ and ϵ, and the rest trivially. Since g^∞ is a coalgebra homomorphism as well (from $\langle g, \mathsf{id}\rangle$ to $(B^\infty Y, \langle \theta_Y, \epsilon_Y\rangle)$), by the fact that coalgebra homomorphisms preserve similarity (Lemma 4.5) we obtain $B^\infty h \circ f^\infty \lesssim_{B^\infty Y} g^\infty \circ h$ as desired.

For item 2. by assumption we have $g(h(x)) \sqsubseteq_{BY} Bh(f(x))$ for all $x \in X$, so $(g(h(x)), f(x)) \in (\sqsubseteq_{BY} \circ \mathsf{Graph}(Bh)^{\mathsf{op}})$, which implies $(g(h(x)), f(x)) \in (\sqsubseteq_{BY} \circ \mathsf{Graph}(Bh)^{\mathsf{op}} \circ \sqsubseteq_{BX})$. But we have

$$\mathsf{Graph}(Bh)^{\mathsf{op}} = (\mathsf{Rel}(B)(\mathsf{Graph}(h)))^{\mathsf{op}} = \mathsf{Rel}(B)(\mathsf{Graph}(h)^{\mathsf{op}})$$

by Lemma 4.1 (item 4 and 5 respectively), so by the above, $\mathsf{Graph}(h)^{\mathsf{op}}$ is a simulation between g and f. Similar to the above case, applying homomorphisms g^∞ and $B^\infty h \circ f^\infty$ yields the desired result. □

5 Monotone biGSOS specifications

Definition 5.1 A biGSOS specification $\rho \colon \Sigma B^\infty \Rightarrow B\Sigma^*$ is *monotone* if the restriction of $\rho_X \times \rho_X$ to $\mathsf{Rel}(\Sigma)(\lesssim_{B^\infty X})$ corestricts to $\sqsubseteq_{B\Sigma^* X}$, for any set X.

If Σ represents a 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 $\sigma(c_1, \dots, c_n)$.

Example 5.2 In the case of an ntree specification for labelled transition systems, if there are no negative premises then the induced biGSOS specification (Example 3.6) is monotone.

To see this, suppose that for all i, we have $x_i \models_h b_i$ (notation of Example 3.6), and $b_i \lesssim_{B^\infty X} c_i$. By definition of $\lesssim_{B^\infty X}$ (cf. Example 4.3) and $\models_h$, and the absence of negative premises, it follows that there is a map $k \colon V \to B^\infty X$ from the set of variables occuring in the premises of the rules (notation of Example 3.6), such that $\epsilon_X \circ k = \epsilon_X \circ h$ (i.e., they agree on node labels), and for all i: $x_i \models_k c_i$. It follows that $\rho_X(\sigma(b_1, \dots, b_n)) \sqsubseteq_{B\Sigma^* X} \rho_X(\sigma(c_1, \dots, c_n))$, as needed.

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.

5.1 Models of monotone specifications

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

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

$$\varphi(f) = B\mu_\emptyset \circ \rho_{\Sigma^*\emptyset} \circ \Sigma f^\infty \circ \iota_\emptyset^{-1} \tag{7}$$

where $\iota_\emptyset \colon \Sigma\Sigma^*\emptyset \to \Sigma^*\emptyset$ is the initial Σ-algebra. Since $\iota_\emptyset$ is an isomorphism (Section 2.1), a function f is a fixed point of φ if and only if it is a supported model of ρ (Equation (6)). We are interested in the *least* supported model. To show that it exists, since $\mathsf{coalg}(B)_{\Sigma^*\emptyset}$ is a pointed DCPO, it suffices to show that φ is monotone.

Lemma 5.3 *The function φ is monotone.*

Proof Suppose $f, g\colon \Sigma^*\emptyset \to B\Sigma^*\emptyset$ and $f \sqsubseteq_{B\Sigma^*\emptyset} g$. By Lemma 4.6, we have $f^\infty \lesssim_{B^\infty\Sigma^*\emptyset} g^\infty$. By Lemma 4.1 (6) we derive

$$\Sigma f^\infty \ \mathsf{Rel}(\Sigma)(\lesssim_{B^\infty\Sigma^*\emptyset})\ \Sigma g^\infty$$

and now the result follows by monotonicity of ρ (assumption) and monotonicity of $B\mu_\emptyset$ (B is ordered). □

Corollary 5.4 *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 4.4), 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 5.5 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 $\mathit{ones}, \mathit{pos}$, c:

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

$$\frac{}{\mathit{ones} \xrightarrow{1} \mathit{ones}} \qquad \frac{}{\mathit{pos} \xrightarrow{1} \mathit{ones} \oplus \mathit{pos}} \qquad \frac{}{c \xrightarrow{1} \sigma(c)}$$

where $+$ and $\times$ denote addition and multiplication of natural numbers, respectively. The rules for $\oplus$ and $\otimes$ define standard pointwise addition and multiplication by a constant, respectively. the operator σ is slightly more curious; of interest here is that it requires lookahead, and is therefore not a GSOS rule. In fact, 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 $\sigma(pos)$ is the stream $(1, 6, 120, \ldots) = (1!, 3!, 5!, \ldots)$. 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 5.4 applies. The specification in Example 3.7 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 $t \in \Sigma^*\emptyset$.

6 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 certain monotone function. In the present section we show that, given a monotone biGSOS specification, the construction of a least model generalises 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 $\mathsf{DCPO}_\bot$*-ordered functor* as an ordered functor (Section 4) where PreOrd is replaced by $\mathsf{DCPO}_\bot$. Below we assume that $(B, \sqsubseteq)$ is $\mathsf{DCPO}_\bot$-ordered, and Σ and B are as before (having a free monad and cofree comonad respectively). Throughout this section we assume a monotone biGSOS specification $\rho\colon \Sigma B^\infty \Rightarrow B\Sigma^*$.

Example 6.1 A general class of functors that are $\mathsf{DCPO}_\bot$-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 4.4). 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 7.

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

$$\begin{tikzcd}
\Sigma B^\infty \Sigma^* X \arrow[r, "\rho_{\Sigma^* X}"] & B\Sigma^*\Sigma^* X \arrow[r, "B\mu_X"] & B\Sigma^* X & BX \arrow[l, "B\eta_X"'] \\
\Sigma\Sigma^* X \arrow[u, "\Sigma(\overline{c})^\infty"] \arrow[rr, "\iota_X"'] & & \Sigma^* X \arrow[u, "\overline{c}"] & X \arrow[l, "\eta_X"] \arrow[u, "c"']
\end{tikzcd} \tag{8}$$

Equivalently, $\overline{c}$ is the least fixed point of the operator

$$\begin{aligned} \varphi_c \ :\ & \mathsf{coalg}(B)_{\Sigma^* X} \to \mathsf{coalg}(B)_{\Sigma^* X} \\ & f \mapsto [B\mu_X \circ \rho_{\Sigma^*\emptyset} \circ \Sigma f^\infty, B\eta_X \circ c] \circ [\iota_X, \eta_X]^{-1} . \end{aligned}$$

Following the proof of Lemma 5.3 it is easy to verify:

Lemma 6.2 *For any $c \colon X \to BX$, the function φ_c is monotone.*

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

Theorem 6.3 *The functor $\overline{\Sigma^*} \colon \mathsf{coalg}(B) \to \mathsf{coalg}(B)$ defined by*

$$\overline{\Sigma^*}(X, c) = (\Sigma^* X, \overline{c}) \qquad \textit{and} \qquad \overline{\Sigma^*}(h) = \Sigma^* h$$

is a lifting of the functor Σ^*.

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

The proof is by transfinite induction on the iterative construction of $\overline{c}$ and $\overline{d}$ 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 $f_i \colon \Sigma^* X \to B\Sigma^* X$ and another directed family $g_i \colon \Sigma^* Y \to B\Sigma^* Y$, 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 $f \colon \Sigma^* X \to B\Sigma^* X$ and $g \colon \Sigma^* Y \to B\Sigma^* Y$ 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:

$$\begin{tikzcd}
\Sigma^* X \arrow[r, "{[\iota_X, \eta_X]^{-1}}"] \arrow[d, "\Sigma^* h"'] & \Sigma\Sigma^* X + X \arrow[r, "\Sigma f^\infty + c"] \arrow[d, "\Sigma\Sigma^* h + h"] & \Sigma B^\infty \Sigma^* X + BX \arrow[r, "\rho_{\Sigma^* X} + \mathsf{id}"] \arrow[d, "\Sigma B^\infty \Sigma^* h + Bh"] & B\Sigma^*\Sigma^* X + BX \arrow[r, "{[B\mu_X, B\eta_X]}"] \arrow[d, "B\Sigma^*\Sigma^* h + Bh"] & B\Sigma^* X \arrow[d, "B\Sigma^* h"] \\
\Sigma^* Y \arrow[r, "{[\iota_Y, \eta_Y]^{-1}}"'] & \Sigma\Sigma^* Y + Y \arrow[r, "\Sigma g^\infty + d"'] & \Sigma B^\infty \Sigma^* Y + BY \arrow[r, "\rho_{\Sigma^* Y} + \mathsf{id}"'] & B\Sigma^*\Sigma^* Y + BY \arrow[r, "{[B\mu_Y, B\eta_Y]}"'] & B\Sigma^* Y
\end{tikzcd}$$

From left to right, the first square commutes by naturality of $[\iota, \eta]$ (and the fact that it is an isomorphism), the second by assumption that $\Sigma^* h$ 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. □

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

Theorem 6.4 *If B preserves weak pullbacks, then the functor $\overline{\Sigma^*}$ extends to a monad $(\overline{\Sigma^*}, \eta, \mu)$ on $\mathsf{coalg}(B)$, which is a lifting of the free monad (Σ^*, η, μ).*

Proof Let (X, c) be a $B\Sigma^*$-coalgebra. We need to show that $\eta_X\colon X \to \Sigma^* X$ is a coalgebra homomorphism from (X, c) to $\overline{\Sigma^*}(X, c)$, and $\mu_X\colon \Sigma^*\Sigma^* X \to \Sigma^* X$ is a coalgebra homomorphism from $\overline{\Sigma}(\overline{\Sigma}(X, c))$ to $\overline{\Sigma}(X, c)$:

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "c"'] & \Sigma^* X \arrow[d, "\overline{c}"] \\
BX \arrow[r, "B\eta_X"] & B\Sigma^* X
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma^*\Sigma^* X \arrow[r, "\mu_X"] \arrow[d, "\overline{\overline{c}}"'] & \Sigma^* X \arrow[d, "\overline{c}"] \\
B\Sigma^*\Sigma^* X \arrow[r, "B\mu_X"] & B\Sigma^* X
\end{tikzcd}$$

For η_X, this is immediate from the definition of $\overline{c}$. For μ_X, we split the proof in two inequalities, both established by transfinite induction. First we prove $B\mu_X \circ \overline{\overline{c}} \sqsubseteq_{B\Sigma^* X} \overline{c} \circ \mu_X$. Suppose $B\mu_X \circ f_i \sqsubseteq_{B\Sigma^* X} \overline{c} \circ \mu_X$ for a (possibly empty) directed family of coalgebras $f_i\colon \Sigma^*\Sigma^* X \to B\Sigma^*\Sigma^* X$. Then

$$B\mu_X \circ \bigvee_i f_i = \bigvee_i (B\mu_X \circ f_i) \sqsubseteq_{B\Sigma^* X} \overline{c} \circ \mu_X$$

by assumption and since $B\mu_X$ is continuous (B is $\mathsf{DCPO}_\perp$-ordered).

Now suppose $B\mu_X \circ f \sqsubseteq_{B\Sigma^* X} \overline{c} \circ \mu_X$ for some $f\colon \Sigma^*\Sigma^* X \to B\Sigma^*\Sigma^* X$. We must show that $B\mu_X \circ \varphi_{\overline{c}}(f) \sqsubseteq_{B\Sigma^* X} \overline{c} \circ \mu_X$. Applying Lemma 4.7 to the assumption yields $B^\infty\mu_X \circ f^\infty \lesssim_{B^\infty\Sigma^* X} (\overline{c})^\infty \circ \mu_X$. Applying Lemma 4.1 (6) yields

$$\Sigma(B^\infty\mu_X \circ f^\infty) \ \mathsf{Rel}(\Sigma)(\lesssim_{B^\infty\Sigma^* X}) \ \Sigma((\overline{c})^\infty \circ \mu_X)\,.$$

Since ρ is monotone, and $B\mu_X$ as well, we obtain

$$B\mu_X \circ \rho_{\Sigma^* X} \circ \Sigma(B^\infty\mu_X \circ f^\infty) \sqsubseteq_{B\Sigma^* X} B\mu_X \circ \rho_{\Sigma^* X} \circ \Sigma((\overline{c})^\infty \circ \mu_X)\,. \tag{9}$$

The following commutes by naturality of ρ and a monad axiom:

$$\begin{tikzcd}
\Sigma B^\infty \Sigma^*\Sigma^* X \arrow[r, "\Sigma B^\infty \mu_X"] \arrow[d, "\rho_{\Sigma^*\Sigma^* X}"] & \Sigma B^\infty \Sigma^* X \arrow[d, "\rho_{\Sigma^* X}"] \\
B\Sigma^*\Sigma^*\Sigma^* X \arrow[r, "B\Sigma^*\mu_X"] \arrow[d, "B\mu_{\Sigma^* X}"] & B\Sigma^*\Sigma^* X \arrow[d, "B\mu_X"] \\
B\Sigma^*\Sigma^* X \arrow[r, "B\mu_X"] & B\Sigma^* X
\end{tikzcd} \tag{10}$$

Combined with (9), this yields

$$B\mu_X \circ B\mu_{\Sigma^* X} \circ \rho_{\Sigma^*\Sigma^* X} \circ \Sigma f^\infty \sqsubseteq_{B\Sigma^* X} B\mu_X \circ \rho_{\Sigma^* X} \circ \Sigma(\overline{c})^\infty \circ \Sigma\mu_X\,. \tag{11}$$

Now consider the following diagram:

$$\begin{tikzcd}
\Sigma^*\Sigma^*X \arrow[r, "\mu_X"] \arrow[d, "{[\iota_{\Sigma^*X},\eta_{\Sigma^*X}]^{-1}}"] \arrow[dddd, bend right=60, "\varphi_{\overline{c}}(f)"'] & \Sigma^*X \arrow[d, "{[\iota_X,\eta_X]^{-1}}"'] \arrow[dddd, bend left=60, "\overline{c}"] \\
\Sigma\Sigma^*\Sigma^*X + \Sigma^*X \arrow[r, "{[\mathsf{in}_l\circ\Sigma\mu_X,[\iota_X,\eta_X]^{-1}]}"] \arrow[d, "\Sigma f^\infty+\overline{c}"] & \Sigma\Sigma^*X + X \arrow[d, "\Sigma(\overline{c})^\infty+c"'] \\
\Sigma B^\infty\Sigma^*\Sigma^*X + B\Sigma^*X \arrow[d, "\rho_{\Sigma^*\Sigma^*X}+\mathsf{id}"] & \Sigma B^\infty\Sigma^*X + BX \arrow[d, "\rho_{\Sigma^*X}+\mathsf{id}"'] \\
B\Sigma^*\Sigma^*\Sigma^*X + B\Sigma^*X \arrow[d, "{[B\mu_{\Sigma^*X},B\eta_{\Sigma^*X}]}"] & B\Sigma^*\Sigma^*X + BX \arrow[d, "{[B\mu_X,B\eta_X]}"'] \\
B\Sigma^*\Sigma^*X \arrow[r, "B\mu_X"] & B\Sigma^*X
\end{tikzcd}$$

The left-most and right-most parts commute; by definition of φ and since $\varphi_c(\overline{c}) = \overline{c}$. For the upper square, one uses that $[\iota_X, \eta_X]$ and $[\iota_{\Sigma^*X}, \eta_{\Sigma^*X}]$ are isomorphisms, and proves that for both components of $\Sigma\Sigma^*\Sigma^*X + \Sigma^*X$, the two paths to Σ^* (obtained by inverting the iso) commute; for the left component this is because μ_X is an algebra homomorphism from ι_{Σ^*X} to ι_X, for the right component since $\mu_X \circ \eta_{\Sigma^*X} = \mathsf{id} = [\iota_X, \eta_X] \circ [\iota_X, \eta_X]^{-1}$.

Now, proving that $B\mu_X \circ \varphi_{\overline{c}}(f) \sqsubseteq_{B\Sigma^*X} \overline{c} \circ \mu_X$, by the above diagram reduces to Equation (11)—which we already have—and

$$B\mu_X \circ B\eta_{\Sigma^*X} \circ \overline{c} \sqsubseteq_{B\Sigma^*X} [B\mu_X \circ \rho_{\Sigma^*\emptyset} \circ \Sigma(\overline{c})^\infty, B\eta_X \circ c] \circ [\iota_X, \eta_X]^{-1} (= \varphi_c(\overline{c})) \quad (12)$$

which holds since $B\mu_X \circ B\eta_{\Sigma^*X} = \mathsf{id}$ (a monad axiom), and $\overline{c} = \varphi_c(\overline{c})$. This concludes the proof for the successor step.

For the other inequality, i.e., $B\mu_X \circ \overline{\overline{c}} \sqsupseteq_{B\Sigma^*X} \overline{c} \circ \mu_X$, one proves again by transfinite induction that $B\mu_X \circ \overline{\overline{c}} \sqsupseteq_{B\Sigma^*X} f \circ \mu_X$ for any $f\colon \Sigma^*X \to B\Sigma^*X$ in the chain approximating $\overline{c}$. The limit case is straightforward. For the successor step, assume $B\mu_X \circ \overline{\overline{c}} \sqsupseteq_{B\Sigma^*X} f \circ \mu_X$; we must prove that $B\mu_X \circ \overline{\overline{c}} \sqsupseteq_{B\Sigma^*X} \varphi_c(f) \circ \mu_X$.

We apply Lemma 4.7 to the assumption, to obtain $B^\infty\mu_X \circ (\overline{\overline{c}})^\infty \gtrsim_{B^\infty\Sigma^*X} f^\infty \circ \mu_X$. Similar to the previous case, we derive

$$B\mu_X \circ B\mu_{\Sigma^*X} \circ \rho_{\Sigma^*\Sigma^*X} \circ \Sigma(\overline{\overline{c}})^\infty \sqsupseteq_{B\Sigma^*X} B\mu_X \circ \rho_{\Sigma^*X} \circ \Sigma(f^\infty \circ \mu_X)\,. \quad (13)$$

And we consider the following diagram:

$$\begin{tikzcd}
\Sigma^*\Sigma^*X \arrow[r, "\mu_X"] \arrow[d, "{[\iota_{\Sigma^*X},\eta_{\Sigma^*X}]^{-1}}"] \arrow[dddd, bend right=60, swap, "\overline{\overline{c}}"] & \Sigma^*X \arrow[d, swap, "{[\iota_X,\eta_X]^{-1}}"] \arrow[dddd, bend left=60, "\varphi_c(f)"] \\
\Sigma\Sigma^*\Sigma^*X+\Sigma^*X \arrow[r, "{[\mathsf{in}_l\circ\Sigma\mu_X,[\iota_X,\eta_X]^{-1}]}"] \arrow[d, "\Sigma(\overline{\overline{c}})^\infty+\overline{c}"] & \Sigma\Sigma^*X+X \arrow[d, swap, "\Sigma f^\infty+c"] \\
\Sigma B^\infty\Sigma^*\Sigma^*X+B\Sigma^*X \arrow[d, "\rho_{\Sigma^*\Sigma^*X}+\mathsf{id}"] & \Sigma B^\infty\Sigma^*X+BX \arrow[d, swap, "\rho_{\Sigma^*X}+\mathsf{id}"] \\
B\Sigma^*\Sigma^*\Sigma^*X+B\Sigma^*X \arrow[d, "{[B\mu_{\Sigma^*X},B\eta_{\Sigma^*X}]}"] & B\Sigma^*\Sigma^*X+BX \arrow[d, swap, "{[B\mu_X,B\eta_X]}"] \\
B\Sigma^*\Sigma^*X \arrow[r, "B\mu_X"] & B\Sigma^*X
\end{tikzcd}$$

The left-most and right-most parts commute; by definition of φ and since $\varphi_{\overline{c}}(\overline{\overline{c}}) = \overline{\overline{c}}$. Similar to the previous case, the upper square commutes, and because of (13) the only remaining proof obligation is:

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

which holds: we have $\overline{c} \sqsupseteq_{B\Sigma^*X} f$ by assumption, hence $\varphi_c(\overline{c}) \sqsupseteq_{B\Sigma^*X} \varphi_c(f)$ since φ_c is monotone, and combined with $\varphi_c(\overline{c}) = \overline{c}$ this gives the desired result. □

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

Theorem 6.5 *Let $\rho\colon \Sigma B^\infty \Rightarrow B\Sigma^*$ be a monotone biGSOS specification, where B is $\mathsf{DCPO}_\perp$-ordered and preserves weak pullbacks. There exists a distributive law $\lambda\colon \Sigma^* B^\infty \Rightarrow B^\infty \Sigma^*$ 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 6.4, we obtain a lifting of (Σ^*, η, μ) to $\mathsf{coalg}(B)$. As explained in Section 2.3, such a lifting corresponds uniquely to a distributive law of the desired type. The operational model of λ can be obtained by applying the lifting to the unique coalgebra $!\colon \emptyset \to B\emptyset$. But that coincides, by definition of the lifting, with the least supported model as defined in Section 5. □

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.

At the end of Section 3, we recalled that in general it is undecidable whether a given biGSOS specification extends uniquely to a distributive law. Above, we have shown how to construct a distributive law for a *monotone* biGSOS specification, in terms of a lifting. We next show that the constructed distributive law indeed extends the original biGSOS specification.

Theorem 6.6 *The distributive law λ constructed in Theorem 6.5 extends the original monotone biGSOS specification ρ.*

Proof We first recall how the distributive law λ is obtained from the monad lifting $\overline{\Sigma^*}$. To this end, let $\overline{\theta_X} \colon \Sigma^* X \to B\Sigma^* X$ be the coalgebra obtained by applying the lifting $\overline{\Sigma^*}$ to the coalgebra $\theta_X \colon B^\infty \to BB^\infty$. The distributive law λ is characterised, on a component X, by finality, as the unique map making the following diagram commute:

$$\begin{tikzcd}
 & \Sigma^* X \\
\Sigma^* B^\infty X \arrow[ur, "\Sigma^* \epsilon_X"] \arrow[r, "\lambda_X"] \arrow[d, "\overline{\theta_X}"'] & B^\infty \Sigma^* X \arrow[u, "\epsilon_{\Sigma^* X}"'] \arrow[d, "\theta_{\Sigma^* X}"] \\
B\Sigma^* B^\infty X \arrow[r, "B\lambda_X"] & BB^\infty \Sigma^* X
\end{tikzcd} \tag{14}$$

Now, consider the following diagram. We will prove that the entire diagram commutes, which implies that λ is an extension of ρ.

$$\begin{tikzcd}
\Sigma B^\infty X \arrow[r, "\Sigma\delta_X"] \arrow[rr, equal, bend left] \arrow[d, "\Sigma\eta_{B^\infty X}"] \arrow[ddd, bend right=60, "\kappa_{B^\infty X}"'] & \Sigma B^\infty B^\infty X \arrow[r, "\Sigma B^\infty \epsilon_X"] \arrow[d, "\Sigma B^\infty \eta_{B^\infty X}"] & \Sigma B^\infty X \arrow[d, "\Sigma B^\infty \eta_X"] \arrow[rdd, bend left, "\rho_X"] & \\
\Sigma\Sigma^* B^\infty X \arrow[r, "\Sigma(\overline{\theta_X})^\infty"] \arrow[dd, "\iota_{B^\infty X}"] & \Sigma B^\infty \Sigma^* B^\infty X \arrow[r, "\Sigma B^\infty \Sigma^* \epsilon_X"] \arrow[d, "\rho_{\Sigma^* B^\infty X}"] & \Sigma B^\infty \Sigma^* X \arrow[d, "\rho_{\Sigma^* X}"] & \\
 & B\Sigma^* \Sigma^* B^\infty X \arrow[r, "B\Sigma^* \Sigma^* \epsilon_X"] \arrow[d, "B\mu_{B^\infty X}"] & B\Sigma^* \Sigma^* X \arrow[d, "B\mu_X"] & B\Sigma^* X \arrow[l, "B\Sigma^* \eta_X"'] \arrow[dl, equal] \\
\Sigma^* B^\infty X \arrow[r, "\overline{\theta_X}"] \arrow[d, "\lambda_X"] & B\Sigma^* B^\infty X \arrow[r, "B\Sigma^* \epsilon_X"] \arrow[d, "B\lambda_X"] & B\Sigma^* X \arrow[d, equal] & \\
B^\infty \Sigma^* X \arrow[r, "\theta_{\Sigma^* X}"] \arrow[rr, bend right, "\nu_{\Sigma^* X}"'] & BB^\infty \Sigma^* X \arrow[r, "B\epsilon_{\Sigma^* X}"] & B\Sigma^* X &
\end{tikzcd}$$

The upper crescent commutes by a comonad axiom, the left by definition of κ and the bottom by definition of ν. The rectangle on the left in the middle commutes by definition of $\overline{\theta_X}$, see (8). The squares on the bottom row commute by (14). Most of the remaining parts commute by naturality; except the upper square on the left. For that, observe first that $\eta_{B^\infty X}$ is a coalgebra morphism by Theorem 6.4, as on the left below:

$$\begin{tikzcd}
B^\infty X \arrow[r, "\eta_{B^\infty X}"] \arrow[d, "\theta_X"'] & \Sigma^* B^\infty X \arrow[d, "\overline{\theta_X}"] \\
BB^\infty X \arrow[r, "B\eta_{B^\infty X}"] & B\Sigma^* B^\infty X
\end{tikzcd}
\qquad
\begin{tikzcd}
B^\infty X \arrow[r, "\eta_{B^\infty X}"] \arrow[d, "\delta_X=\theta_X^\infty"'] & \Sigma^* B^\infty X \arrow[d, "(\overline{\theta_X})^\infty"] \\
B^\infty B^\infty X \arrow[r, "B\eta_{B^\infty X}"] & B^\infty \Sigma^* B^\infty X
\end{tikzcd}$$

Since coinductive extensions preserve homomorphisms, the diagram on the right commutes; finally, we use that $\theta_X^\infty = \delta_X$. □

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 $\mathsf{DCPO}_\perp$-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 6.7 Consider the following specification on (finitely branching) labelled transition systems, involving a unary operator σ and a constant c:

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

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.

7 Liftings for countably accessible functors

In the previous section, we have seen that one of the most important instances—the case of labelled transition systems—does not meet the necessary assumptions for our construction of a distributive law, 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. The general approach is to first construct a lifting of the free monad to the category of coalgebras with a countable carrier, and subsequently extend it to a lifting to the category of all coalgebras.

To state the assumption more precisely, let cSet be the full subcategory of countable sets, with inclusion $I\colon \mathsf{cSet} \to \mathsf{Set}$. We assume that $(B, \sqsubseteq)$ is an ordered functor on Set, and that its restriction to countable sets is $\mathsf{DCPO}_\perp$-ordered:

$$\begin{tikzcd}
 & \mathsf{DCPO}_\perp \arrow[r] & \mathsf{PreOrd} \arrow[d] \\
\mathsf{cSet} \arrow[ur, "\sqsubseteq"] \arrow[r, "I"'] & \mathsf{Set} \arrow[ur, "\sqsubseteq"] \arrow[r, "B"'] & \mathsf{Set}
\end{tikzcd}$$

This is a weaker assumption than in Section 6: before, every set BX was assumed to be a pointed DCPO, whereas here, they only need to be pointed DCPOs when X is countable (and just a preorder otherwise).

Example 7.1 The functor $(\mathcal{P}_c-)^A$ coincides with the $\mathsf{DCPO}_\perp$-ordered functor $(\mathcal{P}-)^A$ when restricted to countable sets, hence it satisfies the above assumption. Notice that $(\mathcal{P}_c-)^A$ is not $\mathsf{DCPO}_\perp$-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 2.3), is ordered as a complete lattice [25], so also $\mathsf{DCPO}_\perp$-ordered. Similar to the above, the functor $(\mathcal{M}_c-)^A$ is $\mathsf{DCPO}_\perp$-ordered when restricted to countable sets, i.e., satisfies the above assumption.

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

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

The main result of this section is that, under certain additional conditions on B and Σ^*, the above lifting extends to a lifting of the monad Σ^* from Set to $\mathsf{coalg}(B)$, and hence a distributive law of the monad Σ^* over the cofree comonad B^∞ (Theorem 7.5). 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. For now, we only recall a concrete characterisation of when a functor on Set is countably accessible, since this will be assumed both for B and Σ^* later on. For details, see [3] and the appendix; the latter also contains the proofs of the results below.

On Set, a functor $B\colon \mathsf{Set} \to \mathsf{Set}$ is *countably accessible* if for every set X and element $x \in BX$, there is an injective function $i\colon Y \to X$ 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 7.3 Any finitary functor is countably accessible. Further, the functors $(\mathcal{P}_c-)^A$ and $(\mathcal{M}_c-)^A$ (c.f. Example 7.1) 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 $\mathsf{cSet} \to \mathsf{cSet}$. We will assume this for our "syntax" functor Σ^*. If Σ correponds 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 $\mathsf{coalg}(B)$ from a lifting to $\mathsf{coalg}_c(B)$ is to *extend* the monad on $\mathsf{coalg}_c(B)$ along the inclusion $\overline{I}\colon \mathsf{coalg}_c(B) \to \mathsf{coalg}(B)$. Concretely, a functor $T\colon \mathsf{Set} \to \mathsf{Set}$ extends $T_c\colon \mathsf{cSet} \to \mathsf{cSet}$ if there is a natural isomorphism $\alpha\colon IT_c \overset{\cong}{\Rightarrow} TI$. 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 subcategory of countably presentable objects[1] can always be extended (Lemma B.2).

Since B is countably accessible, $\mathsf{coalg}(B)$ is locally countably presentable and $\mathsf{coalg}_c(B)$ 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 $\mathsf{coalg}_c(B)$ of Theorem 7.2 to a monad on $\mathsf{coalg}(B)$, resulting in Theorem 7.5 below. The latter relies on Theorem 7.4, 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 $\mathsf{coalg}(B)$ is constructed only up to isomorphism, it is more natural to work with in this setting. We say $(\overline{T}, \overline{\eta}, \overline{\mu})$ lifts (T, η, μ) (up to isomorphism) if there is a natural isomorphism $\alpha\colon U\overline{T} \Rightarrow TU$ such that $\alpha \circ U\overline{\eta} = \eta U$ and $\alpha \circ U\overline{\mu} = \mu U \circ T\alpha \circ \alpha\overline{T}$.

Theorem 7.4 *Let $B\colon \mathsf{Set} \to \mathsf{Set}$ be countably accessible. Suppose (T_c, η^c, μ^c) is a monad on cSet, which lifts to a monad $(\overline{T}_c, \overline{\eta}^c, \overline{\mu}^c)$ on $\mathsf{coalg}_c(B)$. Then*

1. *(T_c, η^c, μ^c) extends to (T, η, μ) along $I\colon \mathsf{cSet} \to \mathsf{Set}$,*
2. *$(\overline{T}_c, \overline{\eta}^c, \overline{\mu}^c)$ extends to $(\overline{T}, \overline{\eta}, \overline{\mu})$ along $\overline{I}\colon \mathsf{coalg}_c(B) \to \mathsf{coalg}(B)$,*
3. *$(\overline{T}, \overline{\eta}, \overline{\mu})$ is a lifting (up to isomorphism) of (T, η, μ).*

The (functor) liftings are summarised in the following diagram. The back face represents the assumed lifting, the left and right faces commute trivially, the bottom and top faces depict the extensions of the first two items respectively, and the

[1] On Set, the countably presentable objects are precisely the countable sets.

front face the lifting in the third item.

$$\begin{tikzcd}
\mathsf{coalg}_c(B) \arrow[rr, "\overline{T}_c"] \arrow[dr, "\overline{I}"] \arrow[dd, "U"'] & & \mathsf{coalg}_c(B) \arrow[dr, "\overline{I}"] \arrow[dd, "U"' near end] & \\
& \mathsf{coalg}(B) \arrow[rr, crossing over, "\overline{T}" near end] & & \mathsf{coalg}(B) \arrow[dd, "U"] \\
\mathsf{cSet} \arrow[rr, "T_c" near end] \arrow[dr, "I"'] & & \mathsf{cSet} \arrow[dr, "I"] & \\
& \mathsf{Set} \arrow[rr, "T"] \arrow[from=uu, crossing over, "U" near end] & & \mathsf{Set}
\end{tikzcd}$$

By instantiating the above theorem with the lifting of Theorem 7.2, the third point gives us the desired lifting to $\mathsf{coalg}(B)$. 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 7.5 *Let $\rho\colon \Sigma B^\infty \Rightarrow B\Sigma^*$ be a monotone biGSOS specification, where B is an ordered functor whose restriction to countable sets is $\mathsf{DCPO}_\perp$-ordered, B is countably accessible, B preserves weak pullbacks, and Σ^* is strongly countably accessible. There exists a distributive law $\lambda\colon \Sigma^* B^\infty \Rightarrow B^\infty \Sigma^*$ 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 7.3 and Example 7.1, 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 [10]). 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.

8 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 [16]; 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 4.7, which in turn is used in the proof that the free monad lifts to the category of coalgebras (Theorem 6.4). Further, in the current paper we focused on *monotone* biGSOS specifications $\rho\colon \Sigma B^\infty \Rightarrow B\Sigma^*$. It could be interesting to study stronger order-theoretic conditions; for instance, *continuous* biGSOS specifications. This would require extending the DCPO-order on B to a DCPO-order on the cofree comonad B^∞, for which the techniques of [12] may be of use. The potential advantage of continuous specifications is that they could prohibit rules which use the infinite behaviour of premises, an operationally questionable feature of coGSOS and biGSOS. We leave an investigation of continuous specifications, both at the abstract level of biGSOS and more concrete syntactic formats, for future work.

References

1. Aceto, L., Fokkink, W., Verhoef, C.: Structural operational semantics. In: Handbook of Process Algebra, pp. 197–292. Elsevier Science (2001). DOI 10.1016/B978-044482830-9/50021-7
2. Adámek, J., Porst, H.E.: On tree coalgebras and coalgebra presentations. Theor. Comput. Sci. **311**(1-3), 257–283 (2004). DOI 10.1016/S0304-3975(03)00378-5
3. Adámek, J., Rosický, J.: Locally Presentable and Accessible Categories. Cambridge Tracts in Mathematics. Cambridge University Press (1994). DOI 10.1017/CBO9780511600579
4. Bartels, F.: On generalised coinduction and probabilistic specification formats. Ph.D. thesis, CWI, Amsterdam (2004)
5. Bloom, B., Istrail, S., Meyer, A.: Bisimulation can't be traced. J. ACM **42**(1), 232–268 (1995). DOI 10.1145/200836.200876
6. Bonchi, F., Petrisan, D., Pous, D., Rot, J.: A general account of coinduction up-to. Acta Inf. **54**(2), 127–190 (2017). DOI 10.1007/s00236-016-0271-4
7. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order (2. ed.). Cambridge University Press (2002)
8. Droste, M., Kuich, W.: Semirings and formal power series. In: Handbook of Weighted Automata, pp. 3–28. Springer (2009). DOI 10/bj2xgm
9. Fiore, M., Staton, S.: Positive structural operational semantics and monotone distributive laws. In: CMCS Short Contributions, p. 8 (2010)
10. Gumm, H.P., Schröder, T.: Monoid-labeled transition systems. Electr. Notes Theor. Comput. Sci. **44**(1), 185–204 (2001). DOI 10.1016/S1571-0661(04)80908-3
11. Hansen, H.H., Kupke, C., Rutten, J.: Stream differential equations: Specification formats and solution methods. Logical Methods in Computer Science **13**(1) (2017). DOI 10.23638/LMCS-13(1:3)2017. URL https://doi.org/10.23638/LMCS-13(1:3)2017
12. Hughes, J., Jacobs, B.: Simulations in coalgebra. Theor. Comput. Sci. **327**(1-2), 71–108 (2004). DOI 10.1016/j.tcs.2004.07.022
13. Jacobs, B.: Introduction to Coalgebra: Towards Mathematics of States and Observation, *Cambridge Tracts in Theoretical Computer Science*, vol. 59. Cambridge University Press (2016). DOI 10.1017/CBO9781316823187
14. Jacobs, B., Rutten, J.: An introduction to (co)algebras and (co)induction. In: Advanced Topics in Bisimulation and Coinduction, pp. 38–99. Cambridge University Press (2011). DOI 10.1017/CBO9780511792588.003
15. Johnstone, P.T.: Adjoint lifting theorems for categories of algebras. Bulletin of the London Mathematical Society **7**(3), 294–297 (1975). DOI 10.1112/blms/7.3.294
16. Klin, B.: Structural operational semantics for weighted transition systems. In: J. Palsberg (ed.) Semantics and Algebraic Specification, *LNCS*, vol. 5700, pp. 121–139. Springer (2009). DOI 10/cxqzcf
17. Klin, B.: Bialgebras for structural operational semantics: An introduction. TCS **412**(38), 5043–5069 (2011). DOI 10.1016/j.tcs.2011.03.023
18. Klin, B., Nachyła, B.: Distributive laws and decidable properties of SOS specifications. In: J. Borgström, S. Crafa (eds.) Proc. EXPRESS/SOS 2014, *EPTCS*, vol. 160, pp. 79–93 (2014). DOI 10.4204/EPTCS.160.8
19. Klin, B., Nachyła, B.: Some undecidable properties of SOS specifications. J. Log. Algebr. Meth. Program. **87**, 94–109 (2017). DOI 10.1016/j.jlamp.2016.08.005
20. Lenisa, M., Power, J., Watanabe, H.: Distributivity for endofunctors, pointed and copointed endofunctors, monads and comonads. Electr. Notes Theor. Comput. Sci. **33**, 230–260 (2000). DOI 10.1016/S1571-0661(05)80350-0. URL https://doi.org/10.1016/S1571-0661(05)80350-0

21. Makkai, M., Paré, R.: Accessible Categories: The Foundations of Categorical Model Theory, *Contemporary mathematics - American Mathematical Society*, vol. 104. American Mathematical Society (1989). DOI 10.1090/conm/104
22. Power, J., Watanabe, H.: Combining a monad and a comonad. Theor. Comput. Sci. **280**(1-2), 137–162 (2002). DOI 10.1016/S0304-3975(01)00024-X
23. Rot, J.: Enhanced coinduction. Ph.D. thesis, Leiden University (2015)
24. Rot, J.: Distributive laws for monotone specifications. In: K. Peters, S. Tini (eds.) Proc. EXPRESS/SOS 2017, *EPTCS*, vol. 255, pp. 83–97 (2017). DOI 10.4204/EPTCS.255.6. URL https://doi.org/10.4204/EPTCS.255.6
25. Rot, J., Bonsangue, M.M.: Structural congruence for bialgebraic semantics. J. Log. Algebr. Meth. Program. **85**(6), 1268–1291 (2016). DOI 10.1016/j.jlamp.2016.08.001
26. Rutten, J.: Behavioural differential equations: a coinductive calculus of streams, automata, and power series. TCS **308**(1-3), 1–53 (2003)
27. Rutten, J.J.M.M.: Universal coalgebra: a theory of systems. TCS **249**(1), 3–80 (2000)
28. Rutten, J.J.M.M.: Elements of stream calculus (an extensive exercise in coinduction). Electr. Notes Theor. Comput. Sci. **45**, 358–423 (2001). DOI 10.1016/S1571-0661(04)80972-1. URL https://doi.org/10.1016/S1571-0661(04)80972-1
29. Staton, S.: General structural operational semantics through categorical logic. In: LICS, pp. 166–177. IEEE Computer Society (2008). DOI 10.1109/LICS.2008.43
30. Turi, D., Plotkin, G.: Towards a mathematical operational semantics. In: LICS, pp. 280–291. IEEE Computer Society (1997). DOI 10.1109/LICS.1997.614955

A Naturality of biGSOS induced from ntree

We show that the induced ρ from an ntree specification (Example 3.6) is indeed natural. Thus, weneed to prove that for every map $f\colon X \to Y$:

$$\begin{tikzcd}
\Sigma B^\infty X \arrow[r, "\rho_X"] \arrow[d, "\Sigma B^\infty f"'] & (\mathcal{P}_f \Sigma^* X)^A \arrow[d, "(\mathcal{P}_f \Sigma^* f)^A"] \\
\Sigma B^\infty Y \arrow[r, "\rho_Y"] & (\mathcal{P}_f \Sigma^* Y)^A
\end{tikzcd}$$

commutes. This means concretely that for all $b_1, \dots, b_n \in B^\infty X$ and $a \in A$: (i) for all $u \in \rho_X(\sigma(b_1, \dots, b_n))(a)$, $\Sigma^* f(u) \in \rho_Y(\sigma(B^\infty f(b_1), \dots, B^\infty f(b_n)))(a)$, and (ii) for all $v \in \rho_Y(\sigma(B^\infty f(b_1), \dots, B^\infty f(b_n)))(a)$ there is $u \in \rho_X(\sigma(b_1, \dots, b_n))(a)$ such that $\Sigma^* f(u) = v$.

For (i), suppose $u = \Sigma^*(\epsilon_X \circ h)(t)$ for some t, h and rule of the form (5) s.t. $x_i \models_h b_i$ for all i. Since $B^\infty f$ is a B-coalgebra homomorphism (Example 2.2), the latter implies $x_i \models_{(B^\infty f) \circ h} B^\infty f(b_i)$ for all i. Further,

$$\Sigma^* f(u) = \Sigma^* f(\Sigma^*(\epsilon_X \circ h))(t) = \Sigma^*(f \circ \epsilon_X \circ h))(t) = \Sigma^*(\epsilon_Y \circ (B^\infty f) \circ h)(t)$$

by functoriality and naturality. Hence $\Sigma^* f(u) \in \rho_Y(\sigma(B^\infty f(b_1), \dots, B^\infty f(b_n)))(a)$ as desired.

For (ii), assume $v \in \rho_Y(\sigma(B^\infty f(b_1), \dots, B^\infty f(b_n)))(a)$, i.e., $v = \Sigma^*(\epsilon_Y \circ k)(t)$ for some $k\colon V \to B^\infty(Y)$ with $\forall i. x_i \models_k B^\infty f(b_i)$.

We define a map $h\colon V \to B^\infty X$, by induction on the depth d of the trees (recall that V is the set of nodes of a disjoint collection of trees), such that:

- $(B^\infty f) \circ h = k$, and
- for every node x at depth strictly below d, $x \xrightarrow{a} x'$ implies $h(x) \xrightarrow{a} h(x')$, and $x \overset{a}{\not\to}$ implies $h(x) \overset{a}{\not\to}$.

For $d = 0$ we look at the roots x_i, and define $h(x_i) = b_i$. The conditions are trivially satisfied; in particular $(B^\infty f) \circ h(x_i) = k(x_i)$ since $x_i \models B^\infty f(b_i)$.

For the inductive case, suppose h is defined up to (and including) nodes at depth d satisfying the above properties, and let x be a node at depth d. If $x \overset{a}{\not\to}$ then $k(x) \overset{a}{\not\to}$, hence $B^\infty f(h(x)) = k(x) \overset{a}{\not\to}$ and hence $h(x) \overset{a}{\not\to}$, since $B^\infty f$ is a coalgebra morphism. Otherwise, suppose $x \xrightarrow{a} x'$. Then $k(x) \xrightarrow{a} k(x')$, and since $k(x) = (B^\infty f)(h(x))$ and $B^\infty f$ is a coalgebra morphism, there is a y such that $h(x) \xrightarrow{a} y$ and $B^\infty f(y) = k(x')$. Define $h(x') = y$. This is well-defined since x' only occurs once in the tree (here we use well-foundedness), and it satisfies the requirements.

This concludes the definition of the map h satisfying the above properties. It follows that $x_i \models_h b_i$. Hence $\Sigma^*(\epsilon_X \circ h)(t) \in \rho_X(\sigma(b_1, \dots, b_n))(a)$. But $\Sigma^* f \circ \Sigma^*(\epsilon_X \circ h) = \Sigma^*(f \circ \epsilon_X \circ h) = \Sigma^*(\epsilon_Y \circ B^\infty f \circ h) = \Sigma^*(\epsilon_Y \circ k) = v$, so we are done.

B Proofs of Section 7

In this section, we prove the results on countable presentability and accessibility that were used in Section 7. Parts are shown in the more general setting of κ-presentable categories; we recall a few basics here, but for understanding the proofs basic familiarity with those concepts is assumed. See [3] for an extensive treatment.

A category $\mathcal{C}$ is *locally κ-presentable*, for κ a regular cardinal, if it is locally small, cocomplete, and there is a set S of κ-presentable objects from C such that every object in $\mathcal{C}$ is a κ-filtered colimit of objects from S. For a locally κ-presentable category $\mathcal{C}$, we denote by $\mathcal{C}_\kappa$ the full subcategory of κ-presentable objects, with the inclusion $I\colon \mathcal{C}_\kappa \to \mathcal{C}$.

For arbitrary categories $\mathcal{C}, \mathcal{D}$ we denote the category of functors by $[\mathcal{C}, \mathcal{D}]$. For $\mathcal{C}$ and $\mathcal{D}$ both locally κ-presentable categories, we say $F\colon \mathcal{C} \to \mathcal{C}$ is κ-accessible if it preserves κ-filtered colimits. We denote the category of such κ-accessible functors by $[\mathcal{C}, \mathcal{D}]^\kappa$.

Let ω_1 be the first uncountable ordinal. A category is called *locally countably presentable* if it is locally ω_1-presentable, and a functor is called *countably accessible* if it is ω_1-accessible. The latter coincides with the concrete notion introduced in Section 7.

Let $\mathcal{C}, \mathcal{D}$ be locally κ-presentable categories. Let $I^* \colon [\mathcal{C}, \mathcal{D}]^\kappa \to [\mathcal{C}_\kappa, \mathcal{D}]$ be the functor that precomposes with the inclusion $I \colon \mathcal{C}_\kappa \to \mathcal{C}$. This functor has a left adjoint Lan_I, which is the *left Kan extension (along I)*; for a functor $G \colon \mathcal{C}_\kappa \to \mathcal{D}$, the functor $\mathsf{Lan}_I G$ is called the left Kan extension of G along I; such a left Kan extension is always κ-accessible [21, Proposition 2.4.3]. In fact, gives rise to an adjoint *equivalence*; this is a special case of [21, Corollary 2.1.9].

$$\begin{tikzcd}
{[\mathcal{C}_\kappa, \mathcal{D}]} \arrow[r, bend left, "\mathsf{Lan}_I"] \arrow[r, phantom, "\cong"] & {[\mathcal{C}, \mathcal{D}]^\kappa} \arrow[l, bend left, "I^*"]
\end{tikzcd} \tag{15}$$

The unit ι is given on a component G by the unit $\iota_G \colon G \to (\mathsf{Lan}_I G)I$ of the left Kan extension. Notice in particular that this unit ι_G is always an isomorphism.

Lemma B.1 *Let $\mathcal{C}$ be a locally κ-presentable category, with $I \colon \mathcal{C}_\kappa \to \mathcal{C}$ the subcategory of countably presentable objects. Let $F \colon \mathcal{C} \to \mathcal{C}$ be a κ-accessible functor, $G \colon \mathcal{C}_\kappa \to \mathcal{C}$ a functor and $\alpha \colon G \Rightarrow FI$ a natural isomorphism.*

$$\begin{tikzcd}
\mathcal{C}_\kappa \arrow[d, "I"'] \arrow[dr, "G"] & \\
\mathcal{C} \arrow[r, "F"'] \arrow[ur, phantom, "\cong" near start] & \mathcal{C}
\end{tikzcd}$$

Then (F, α) is a left Kan extension of G along I.

Proof By the universal property of the Kan extension $(\mathsf{Lan}_I G, \iota_G)$, there exists a unique natural transformation $\hat{\alpha}$ such that

$$\begin{tikzcd}
G \arrow[r, Rightarrow, "\iota_G"] \arrow[dr, Rightarrow, "\alpha"'] & \mathsf{Lan}_I GI \arrow[d, Rightarrow, "\hat{\alpha} I"] \\
& FI
\end{tikzcd}$$

Since ι_G and α are both iso (the former by (15)), $\hat{\alpha} I$ is as well. By the equivalence (15), it follows that $\hat{\alpha} \colon \mathsf{Lan}_I G \Rightarrow F$ is iso. It is now easy to show that (F, α) is a left Kan extension. □

Lemma B.2 *Let $\mathcal{C}$ be a locally κ-presentable category, with $I \colon \mathcal{C}_\kappa \to \mathcal{C}$ the subcategory of countably presentable objects. Any monad $(T_\kappa, \eta^\kappa, \mu^\kappa)$ on $\mathcal{C}_\kappa$ extends uniquely to a (κ-accessible) monad (T, η, μ) on $\mathcal{C}$, along $I \colon \mathcal{C}_\kappa \to \mathcal{C}$.*

$$\begin{tikzcd}
\mathcal{C}_\kappa \arrow[r, "T_\kappa"] \arrow[d, "I"'] \arrow[dr, phantom, "\cong"] & \mathcal{C}_\kappa \arrow[d, "I"] \\
\mathcal{C} \arrow[r, "T"'] & \mathcal{C}
\end{tikzcd}$$

Proof First of all, define T as the left Kan extension of IT_κ along I, with unit $\alpha \colon IT_\kappa \Rightarrow TI$ (this is an isomorphism (15)). By Lemma B.1 it follows that, for any i, T^i is a left Kan extension of IT_κ^i along I. In particular, $T^0 = \mathsf{Id}$ and T^2 are Kan extensions:

$$\begin{tikzcd}
\mathcal{C}_\kappa \arrow[r, "\mathsf{Id}"] \arrow[d, "I"'] \arrow[dr, phantom, "\cong \mathsf{id}"] & \mathcal{C}_\kappa \arrow[d, "I"] \\
\mathcal{C} \arrow[r, "\mathsf{Id}"'] & \mathcal{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{C}_\kappa \arrow[r, "T_\kappa"] \arrow[d, "I"'] \arrow[dr, phantom, "\cong \alpha"] & \mathcal{C}_\kappa \arrow[r, "T_\kappa"] \arrow[d, "I"] \arrow[dr, phantom, "\cong \alpha"] & \mathcal{C}_\kappa \arrow[d, "I"] \\
\mathcal{C} \arrow[r, "T"'] & \mathcal{C} \arrow[r, "T"'] & \mathcal{C}
\end{tikzcd}$$

Now, to define the unit and multiplication of T, we use the universal property of the above left Kan extensions respectively on $\alpha \circ \eta^\kappa \colon I \Rightarrow TI$ and $\alpha \circ I\mu^\kappa \colon IT_\kappa T_\kappa \Rightarrow TI$ to obtain a unique η and μ making the following diagrams commute:

$$\begin{tikzcd}
I \arrow[r, "\mathsf{id}"] \arrow[d, "I\eta^\kappa"'] & I \arrow[d, "\eta I"] \\
IT_\kappa \arrow[r, "\alpha"'] & TI
\end{tikzcd}
\qquad
\begin{tikzcd}
IT_\kappa T_\kappa \arrow[r, "\alpha T_\kappa"] \arrow[d, "I\mu^\kappa"'] & TIT_\kappa \arrow[r, "T\alpha"] & TTI \arrow[d, "\mu I"] \\
IT_\kappa \arrow[rr, "\alpha"'] & & TI
\end{tikzcd}
\tag{16}$$

Once we show that (T, η, μ) is a monad, the above diagrams assert that (T, η, μ) is an extension of $(T_\kappa, \eta^\kappa, \mu^\kappa)$.

For the monad axioms, first consider the following diagram:

$$\begin{tikzcd}
IT_\kappa \arrow[rr, "\alpha"] \arrow[d, "IT_\kappa\eta^\kappa"] \arrow[dd, equal, bend right=60] & & TI \arrow[dl, "TI\eta^\kappa"'] \arrow[d, "T\eta I"] \\
IT_\kappa T_\kappa \arrow[r, "\alpha T_\kappa"] \arrow[d, "I\mu^\kappa"] & TIT_\kappa \arrow[r, "T\alpha"'] & TTI \arrow[d, "\mu I"] \\
IT_\kappa \arrow[rr, "\alpha"'] & & TI
\end{tikzcd}$$

The crescent commutes by a monad axiom, the upper left part by naturality of α, and the triangle and lower rectangle by 16. Hence $\mu I \circ T\eta I \circ \alpha = \alpha = \mathsf{id} \circ \alpha$, which means $\mu \circ T\eta = \mathsf{id}$ by the universal property of the Kan extension (T, α). In a similar way we obtain $\mu I \circ \eta TI \circ \alpha = \alpha$, so that $\mu \circ \eta T = \mathsf{id}$.

Finally, for the associativity consider:

$$\begin{tikzcd}
TTTI \arrow[d, "T\mu I"] & TTIT_\kappa \arrow[l, "TT\alpha"'] & TIT_\kappa T_\kappa \arrow[l, "T\alpha T_\kappa"'] \arrow[dl, "TI\mu^\kappa"'] & IT_\kappa T_\kappa T_\kappa \arrow[l, "\alpha T_\kappa T_\kappa"'] \arrow[r, "\alpha T_\kappa T_\kappa"] \arrow[dl, "IT_\kappa\mu^\kappa"'] \arrow[dr, "I\mu^\kappa T_\kappa"] & TIT_\kappa T_\kappa \arrow[r, "T\alpha T_\kappa"] & TTIT_\kappa \arrow[r, "TT\alpha"] \arrow[d, "\mu IT_\kappa"] & TTTI \arrow[d, "\mu TI"] \\
TTI \arrow[d, "\mu I"] & TIT_\kappa \arrow[l, "T\alpha"'] & IT_\kappa T_\kappa \arrow[l, "\alpha T_\kappa"'] \arrow[dr, "I\mu^\kappa"'] & & IT_\kappa T_\kappa \arrow[r, "\alpha T_\kappa"] \arrow[dl, "I\mu^\kappa"] & TIT_\kappa \arrow[r, "T\alpha"] & TTI \arrow[d, "\mu I"] \\
TI & & & IT_\kappa \arrow[lll, "\alpha"'] \arrow[rrr, "\alpha"] & & & TI
\end{tikzcd}$$

which commutes by (16), a monad axiom (associativity μ^κ) and naturality. Hence,

$$\mu I \circ T\mu I \circ TT\alpha \circ T\alpha T_\kappa \circ \alpha T_\kappa T_\kappa = \mu I \circ \mu TI \circ TT\alpha \circ T\alpha T_\kappa \circ \alpha T_\kappa T_\kappa$$

so that $\mu \circ T\mu = \mu \circ \mu T$, since $(T^3, TT\alpha \circ T\alpha T_\kappa \circ \alpha T_\kappa T_\kappa)$ is a left Kan extension of IT^3 along I. □

We now move to countably accessible functors B on Set, to prove the main lifting theorem.

Proof (Theorem 7.4) The first two points are instances of Lemma B.2. For the second point, we use that $\mathsf{coalg}(B)$ is locally countably presentable, with $\mathsf{coalg}_c(B)$ the set of countably presentable objects, since $B \colon \mathsf{Set} \to \mathsf{Set}$ is a countably presentable Set endofunctor [2]. (This is a non-trivial result, and $\mathsf{coalg}(B)$ is not finitely presentable in general even for finitary functors B, as shown as well in [2].)

It remains to show that $(\overline{T}, \overline{\eta}, \overline{\mu})$ is a lifting of (T, η, μ). First, we instantiate the equivalence (15) to:

$$\begin{tikzcd}
{[\mathsf{coalg}_c(B), \mathsf{Set}]} \arrow[r, bend left, "\mathsf{Lan}_{\overline{I}}"] \arrow[r, phantom, "\cong"] & {[\mathsf{coalg}(B), \mathsf{Set}]^c} \arrow[l, bend left, "\overline{I}^*"]
\end{tikzcd}
\tag{17}$$

where $[\mathsf{coalg}(B), \mathsf{Set}]^c$ is the category of countably accessible functors from $\mathsf{coalg}(B)$ to Set.

Next, we observe there is an isomorphism as in the horizontal path below, constructed from the (unnamed) assumed isomorphisms (some are actually equalities) in the cube (drawn with double lines):

$$\begin{tikzcd}
TU\overline{I} \arrow[r, equals, "T\cong"] \arrow[rrrrr, equals, bend right=12, "\alpha\overline{I}"'] & TIU \arrow[r, equals, "\cong U"] & IT_cU \arrow[r, equals, "I\cong"] & IU\overline{T}_c \arrow[r, equals, "\cong\overline{T}_c"] & U\overline{I}\overline{T}_c \arrow[r, equals, "U\cong"] & U\overline{T}\,\overline{I}
\end{tikzcd} \tag{18}$$

The functors T and $\overline{T}$ are both accessible by Lemma B.2. The forgetful functor $U\colon \mathsf{coalg}(B) \to \mathsf{Set}$ always preserves colimits, so it is accessible as well. Hence TU and $U\overline{T}$ are accessible. Now, by the equivalence (17), it is not difficult to show that there is a natural isomorphism α making the above crescent commute.

This proves that $\overline{T}$ is a lifting of T. It remains to show that the monad $(\overline{T}, \overline{\eta}, \overline{\mu})$ is a lifting of (T, η, μ). That follows from the equivalence (17) once we show that the following two diagrams commute:

$$\begin{tikzcd}
TU\overline{I} \arrow[rr, equals, "\alpha\overline{I}"] & & U\overline{T}\,\overline{I} \\
& U\overline{I} \arrow[ul, Rightarrow, "\eta U\overline{I}"] \arrow[ur, Rightarrow, "U\overline{\eta}\overline{I}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
TTU\overline{I} \arrow[r, equals, "T\alpha\overline{I}"] \arrow[d, Rightarrow, "\mu U\overline{I}"'] & TU\overline{T}\,\overline{I} \arrow[r, equals, "\alpha\overline{T}\,\overline{I}"] & U\overline{T}\,\overline{T}\,\overline{I} \arrow[d, Rightarrow, "U\overline{\mu}\overline{I}"] \\
TU\overline{I} \arrow[rr, equals, "\alpha\overline{I}"'] & & U\overline{T}\,\overline{I}
\end{tikzcd}$$

Commutativity essentially follows from the lifting of (T_c, η^c, μ^c) to $(\overline{T}_c, \overline{\eta}^c, \overline{\mu}^c)$ and the two extensions, and a straightforward but tedious computation, by unfolding $\alpha\overline{I}$ as in (18). $\square$