Simulations in Coalgebra

Bart Jacobs, Jesse Hughes

Department of Computer Science, University of Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
Email: {bart,jesseh}@cs.kun.nl *URL:*
http://www.cs.kun.nl/~{bart,jesseh}

Abstract

A new approach to simulations is proposed within the theory of coalgebras by taking a notion of order on a functor as primitive. Such an order forms a basic building block for a "lax relation lifting", or "relator" as used by other authors. Simulations appear as coalgebras of this lifted functor, and similarity as greatest simulation. Two-way similarity is then similarity in both directions. In general, it is different from bisimilarity (in the usual coalgebraic sense), but a sufficient condition is formulated (and illustrated) to ensure that bisimilarity and two-way similarity coincide. Also, a distributive law is identified which ensures that similarity on a final coalgebra forms a dcpo structure.

1 Introduction

Simulations are relations between one (dynamical) system and another, expressing that if one system can do a move, then the other can do a similar move. Simulations are heavily used for transition systems and automata (see *e.g.* [11]), especially for refinement proofs. Also, they are studied in modal logic [2], domain theory [12,5], category theory [16] (using spans, following earlier, unpublished work of Claudio Hermida on modules). Here we study simulations in a purely coalgebraic context, starting from a new, elementary notion of ordering on a functor, and using familiar techniques based on "relation lifting" or "relators".

The main contribution of the paper is systematisation, namely, systematisation of the definition, examples, results (for instance, about the properties of the order) and connections (*e.g.* between two-way similarity and bisimilarity). But many research issues remain.

The paper starts with our main definition, namely of order on a functor in Section 2. These orders are combined with ordinary relation lifting (recalled in Section 3) to form "lax relation liftings" in Section 4. Simulations then appear as coalgebras of such lax relation lifting functors. Similarity is the greatest

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

simulation, and two-way similarity is similarity in both directions. Its relation with ordinary bisimilarity is established in Section 6. Finally, Section 7 turns the similarity order on a final coalgebra into a dcpo structure in presence of a certain distributive law.

2 Orders on functors

We shall write **Sets** for the category of sets and functions, and **PreOrd** for the category of preorders $(X, \leq)$ (with $\leq$ a reflexive and transitive relation on X) and order-preserving (monotone) functions between them. There is an obvious forgetful functor **PreOrd** $\rightarrow$ **Sets** sending a preorder $(X, \leq)$ to its underlying set X. This functor will remain unnamed.

Definition 2.1 Let $F\colon \mathbf{Sets} \rightarrow \mathbf{Sets}$ be an arbitrary endofunctor on **Sets**. We define an **order on** F to be a functor $\sqsubseteq\colon \mathbf{Sets} \rightarrow \mathbf{PreOrd}$ making the following diagram commute.

$$\begin{tikzcd} & \mathbf{PreOrd} \arrow[d] \\ \mathbf{Sets} \arrow[ur, "\sqsubseteq"] \arrow[r, "F"'] & \mathbf{Sets} \end{tikzcd}$$

In this paper our examples are of a set-theoretic nature, so we restrict the above notion to endofunctors on sets, and we do not strive for the highest level of generality. But it is very easy to generalise it to other categories $\mathbb{C}$. The category **PreOrd** should then be suitably replaced by a category of preorders in $\mathbb{C}$ (or even a fibred category of preorder relations over $\mathbb{C}$ in some logic).

In concrete terms, an order $\sqsubseteq$ on a functor F, as just defined, consists of a collection of preorders $\sqsubseteq_X \subseteq F(X) \times F(X)$, for each set X, in such a way that $F(f)\colon F(X) \rightarrow F(Y)$ preserves the order, for each function $f\colon X \rightarrow Y$. Preorderedness seems to be the minimal requirement that one wishes to impose on such orders in the current setting.

Often, like in [12,5], notions of simulation are studied in an ordered setting, where the functor F acts on some category of dcpos. In that case each X and $F(X)$ is a dcpo and thus automatically carries on order. Our approach is minimal in a sense, because it only requires an order on the images $F(X)$ of F, and not on arbitrary objects.

Example 2.2 We illustrate the notion of order on a functor in the following examples.

(i) For each functor $F\colon \mathbf{Sets} \rightarrow \mathbf{Sets}$ we have both the discrete order (only equal elements are related) and the indiscrete one (any two elements are related).

(ii) Consider the functor $\mathcal{S}(X) = 1 + (A \times X)$ which adds a bottom element $*$ to a product set $A \times X$, where A is an arbitrary, fixed set. The behaviours of coalgebras of this functor consist of both finite and infinite sequences of elements of A. The sets $\mathcal{S}(X)$ carry the familiar "flat" order: for $u, v \in \mathcal{S}(X)$,

$$\begin{aligned} u \sqsubseteq v &\iff u \neq * \Rightarrow u = v \\ &\iff \forall a \in A. \forall x \in X. u = (a, x) \Rightarrow v = (a, x). \end{aligned}$$

(In this formulation we have left the coproduct coprojections $1 \xrightarrow{\kappa_1} 1 + (A \times X) \xleftarrow{\kappa_2} A \times X$ implicit.)

(iii) Next we consider the list (or free monoid) functor $\mathcal{L}(X) = X^\star$. A coalgebra of this functor maps an element to a finite list of successor states $\langle x_0, \ldots, x_{n-1} \rangle$, so that order and multiplicity of such states matter. Several orderings on $\mathcal{L}$ are possible, which may or may not take the order and multiplicity into account.

$$\begin{aligned} &\langle x_0, \ldots, x_{n-1} \rangle \sqsubseteq_1 \langle y_0, \ldots, y_{m-1} \rangle \\ &\iff \text{there is a strictly monotone function} \\ &\qquad\qquad \varphi\colon \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, m-1\} \\ &\qquad \text{with } x_i = y_{\varphi(i)}, \text{ for } i < m. \end{aligned}$$

Strict monotonicity means that $i < j$ implies $\varphi(i) < \varphi(j)$. As a result, φ is injective, and $n \leq m$. This order $\sqsubseteq_1$ basically says that the smaller sequence can be obtained by removing elements from the bigger one.

Our second ordering on $\mathcal{L}$ is much simpler, and ignores much of the existing structure:

$$\langle x_0, \ldots, x_{n-1} \rangle \sqsubseteq_2 \langle y_0, \ldots, y_{m-1} \rangle \iff \forall i < n. \exists j < m. x_i = y_j.$$

Thus, for different elements $x, y, z \in X$ we have $\langle x, z \rangle \sqsubseteq_i \langle x, x, y, z \rangle$ for both $i = 1, 2$. But $\langle y, x, x \rangle \sqsubseteq_i \langle x, y \rangle$ only holds for $i = 2$. Clearly, $\sqsubseteq_1 \subseteq \sqsubseteq_2$.

(iv) Our next example involves the related "bag" functor $\mathcal{B}$, capturing free commutative monoids (as algebras of the associated monad). It can be described as:

$$\mathcal{B}(X) = \{\alpha\colon X \to \mathbb{N} \mid \text{only finitely many } x \in X \text{ have } \alpha(x) \neq 0\}.$$

Often one says that such an α has "finite support". When using the bag instead of the list functor, we care about multiplicities $\alpha(x)$ of elements $x \in X$, but not about the order in which they occur. Like before we consider two orderings on the functor $\mathcal{B}$.

$$\alpha \sqsubseteq_1 \beta \iff \forall x \in X. \alpha(x) \leq \beta(x).$$

When we wish to ignore multiplicities and only consider occurrences we order as follows.

$$\alpha \sqsubseteq_2 \beta \iff \forall x \in X.\, \alpha(x) \neq 0 \Rightarrow \beta(x) \neq 0.$$

This says that if x occurs in α, then it should also occur in β, without requiring a relation between the multiplicities of x in α and in β (like in $\sqsubseteq_1$).

(v) Our final example involves the powerset functor $\mathcal{P}$ with a set A of "labels", in the functor $\mathcal{T}(X) = \mathcal{P}(X)^A \cong \mathcal{P}(A \times X)$. As is well-known, coalgebras of this functor are labeled transition systems. The obvious order on $\alpha, \beta \in \mathcal{T}(X)$ is pointwise inclusion:

$$\alpha \sqsubseteq \beta \iff \forall a \in A.\, \alpha(a) \subseteq \beta(a).$$

One could also consider the reverse inclusion $\supseteq$ instead of $\subseteq$.

At the end of this section we like to point out that our general notion of order on a functor, as given in Definition 2.1, allows us to formulate general results like: given a natural transformation $\sigma\colon F \Rightarrow G$, then an order $\sqsubseteq^G$ on G induces an order $\sqsubseteq^F \stackrel{\text{def}}{=} \sigma^*(\sqsubseteq^G)$ on F, namely as $u \sqsubseteq^F v \iff \sigma_X(u) \sqsubseteq^G \sigma_X(v)$, for $u, v \in F(X)$. In this way one can organise orders in a category which is fibred over a category of endofunctors.

Also, for a functor F with order $\sqsubseteq$ one can define a category $\mathrm{CoAlg}_{\sqsubseteq}(F)$ of F-coalgebras with "simulation mappings": a map f from $X \xrightarrow{c} F(X)$ to $Y \xrightarrow{d} F(Y)$ in $\mathrm{CoAlg}_{\sqsubseteq}(F)$ is then a function $f\colon X \to Y$ with $F(f)(c(x)) \sqsubseteq d(f(x))$ on $F(Y)$. Such a category is sometimes used for transition systems, if one wants maps to only preserve (and not reflect) transitions.

3 A recap on relation lifting and bisimulations

We shall write **Rel** for the category of binary relations. Its objects are arbitrary relations $R \subseteq X_1 \times X_2$; and its morphisms from $R \subseteq X_1 \times X_2$ to $S \subseteq Y_1 \times Y_2$ are pairs of functions $f_1\colon X_1 \to Y_1$, $f_2\colon X_2 \to Y_2$ between the underlying sets which preserve the relation, in the sense that $R(x_1, x_2) \Rightarrow S(f_1(x_1), f_2(x_2))$. There is then an obvious forgetful functor $\mathbf{Rel} \to \mathbf{Sets} \times \mathbf{Sets}$ mapping a relation to its underlying sets. Notice that there is a full and faithful embedding $\mathbf{PreOrd} \hookrightarrow \mathbf{Rel}$, describing preorders as a subcategory.

It is fairly standard in the theory of coalgebras [7,10] to associate with an endofunctor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ a **relation lifting** $\mathrm{Rel}(F)\colon \mathbf{Rel} \to \mathbf{Rel}$ in a diagram:

$$
\begin{tikzcd}
\mathbf{Rel} \arrow[r, "\mathrm{Rel}(F)"] \arrow[d] & \mathbf{Rel} \arrow[d] \\
\mathbf{Sets} \times \mathbf{Sets} \arrow[r, "F \times F"'] & \mathbf{Sets} \times \mathbf{Sets}
\end{tikzcd}
$$

For an arbitrary functor, this relation lifting $\mathrm{Rel}(F)$ can be defined on a relation $\langle r_1, r_2\rangle \colon R \hookrightarrow X_1 \times X_2$ by taking the image of the pair

$$\langle F(r_1), F(r_2)\rangle \colon F(R) \longrightarrow F(X_1) \times F(X_2),$$

see *e.g* [4,13]. In the language of fibred categories, then,

$$\mathrm{Rel}(F)(R) = \coprod\nolimits_{\langle Fr_1, Fr_2\rangle} F(R)$$

and in set-theoretic terms,

$$\begin{aligned}&\mathrm{Rel}(F)(R)\\ &\quad = \{\langle u, v\rangle \in FX_1 \times FX_2 \mid \exists w \in F(R).\, F(r_1)(w) = u \text{ and } F(r_2)(w) = v\}.\end{aligned}$$

For the special case of polynomially defined functors F, $\mathrm{Rel}(F)$ may equivalently be defined by induction the structure of F, see *e.g.* [10].

This relation lifting is assumed to satisfy the following properties.

(i) Equality is preserved: $\mathrm{Rel}(F)(=_X) = =_{F(X)}$.

(ii) Composition is preserved: for $R \subseteq X \times Y$ and $S \subseteq Y \times Z$, the relational composition $S \circ R = \{(x, z) \mid \exists y.\, R(x, y) \wedge S(y, z)\}$ satisfies: $\mathrm{Rel}(F)(S \circ R) = \mathrm{Rel}(F)(S) \circ \mathrm{Rel}(F)(R)$.

(iii) Inclusions are preserved: if $R \subseteq S$ then $\mathrm{Rel}(F)(R) \subseteq \mathrm{Rel}(F)(S)$.

(iv) Reversals are preserved: $\mathrm{Rel}(F)(R^{\mathrm{op}}) = \mathrm{Rel}(F)(R)^{\mathrm{op}}$.

(v) Inverse images (or substitution, or reindexing) is preserved: for functions $f_1 \colon X_1 \to Y_1$, $f_2 \colon X_2 \to Y_2$ and a relation $S \subseteq Y_1 \times Y_2$ we have:

$$\mathrm{Rel}(F)\Big((f_1 \times f_2)^{-1}(S)\Big) = (F(f_1) \times F(f_2))^{-1}\Big(\mathrm{Rel}(F)(S)\Big).$$

All these properties hold for functors F that preserve weak pullbacks.

For example, as a consequence, the graph relation

$$\mathsf{Graph}(f) = (f \times id)^*(=_Y) \subseteq X \times Y$$

of a function $f \colon X \to Y$ satisfies

$$\mathrm{Rel}(F)\big(\mathsf{Graph}(f)\big) = \mathsf{Graph}(F(f)).$$

A **bisimulation** is then just a $\mathrm{Rel}(F)$-coalgebra. It is a map in **Rel** over two maps in **Sets**, which are the underlying coalgebras. Concretely, in terms of such coalgebras $c\colon X \to F(X)$ and $d\colon Y \to F(Y)$ of the same functor F, a bisimulation (between c and d) is a relation $R \subseteq X \times Y$ satisfying for all $x \in X$ and $y \in Y$,

$$R(x, y) \Longrightarrow \mathrm{Rel}(F)(R)(c(x), d(y)).$$

Or, pictorially, as a map in **Rel**:

$$\begin{tikzcd} R \arrow[r, dashed] \arrow[d, tail] & \mathrm{Rel}(F)(R) \arrow[d, tail] \\ X \times Y \arrow[r, "c \times d"'] & F(X) \times F(Y) \end{tikzcd}$$

The next result mentions some standard properties (see *e.g.* [14]) that are relevant in the current setting. Proofs are omitted.

Proposition 3.1 *Let F be an endofunctor on* **Sets** *with a relation lifting functor functor Rel(F) as described above. Then, with respect to coalgebras $X \xrightarrow{c} FX$ and $Y \xrightarrow{d} FY$ one has that:*

(i) *Bisimulations are closed under arbitrary unions; as a result, there is a greatest bisimulation relation $\underline{\leftrightarrow} \subseteq X \times Y$, which is called* ***bisimilarity****.*

(ii) *The equality relation $=_X \subseteq X \times X$ is a bisimulation (for the single coalgebra c). Similarly, bisimilarity $\underline{\leftrightarrow} \subseteq X \times X$ is an equivalence relation.*

(iii) *An arbitrary function $f\colon X \to Y$ is a homomorphism of coalgebras (that is, satisfies $d \circ f = F(f) \circ c$) if and only if its graph relation $\mathsf{Graph}(f)$ is a bisimulation.*
Hence if f is a homomorphism, then $x \,\underline{\leftrightarrow}\, f(x)$.

(iv) *For a homomorphism $f\colon X \to Y$ and elements $x, x' \in X$ one has $x \,\underline{\leftrightarrow}\, x'$ iff $f(x) \,\underline{\leftrightarrow}\, f(x')$.*

(v) *If F has a final coalgebra $Z \xrightarrow{\cong} FZ$, then bisimilarity on Z is equality. Hence for $x \in X$ and $y \in Y$ one has $x \,\underline{\leftrightarrow}\, y$ iff $!(x) = !(y)$—where $!$ is the unique homomorphism to the final coalgebra.*

Example 3.2 We briefly describe bisimulations for the examples from the previous section.

(i) Consider two coalgebras $X \xrightarrow{c} \mathcal{S}(X)$, $Y \xrightarrow{d} \mathcal{S}(Y)$ of the sequence functor $\mathcal{S}(X) = 1 + (A \times X)$. A relation $R \subseteq X \times Y$ is a bisimulation iff for all $x \in X$ and $y \in Y$ with $R(x, y)$ we have either $c(x) = d(y) = *$, or $c(x) = (a, x')$ and $d(y) = (b, y')$ with $a = b$ and $R(x', y')$.

(ii) For two list-functor coalgebras $X \xrightarrow{c} X^\star$, $Y \xrightarrow{d} Y^\star$ we have $z \,\underline{\leftrightarrow}\, w$ iff there is a relation $R \subseteq X \times Y$ with $R(z, w)$ and for all elements $x \in X$ and $y \in Y$, if $R(x, y)$, then if $c(x) = \langle x_0, \ldots, x_{n-1}\rangle$ and if $d(y) = \langle y_0, \ldots, y_{m-1}\rangle$, then $n = m$ and $R(x_i, y_i)$ for all $i < n$.

(iii) For bag-coalgebras $X \xrightarrow{c} \mathcal{B}(X)$, $Y \xrightarrow{d} \mathcal{B}(Y)$ the situation is more complicated. A relation R is a bisimulation iff for all $x \in X$ and $y \in Y$ with $R(x,y)$ there is a $\gamma\colon R \to \mathbb{N}$ such that the following hold.
- $\gamma(x,y) = 0$ for all but finitely many x and y.
- $c(x)(x') = \sum_{y'}\{\gamma(x',y') \mid R(x',y')\}$
- $d(y)(y') = \sum_{x'}\{\gamma(x',y') \mid R(x',y')\}$.

(iv) Finally, for transition system coalgebras $X \xrightarrow{c} \mathcal{P}(X)^A$, $Y \xrightarrow{d} \mathcal{P}(Y)^A$, a relation $R \subseteq X \times Y$ is a bisimulation as defined above iff it is a (strong) bisimulation in the usual sense: if $R(x,y)$, then both:
- if $x \xrightarrow{a} x'$ (i.e., $x' \in c(x)(a)$), then there is an $y' \in Y$ with $y \xrightarrow{a} y'$ and $R(x',y')$.
- if $y \xrightarrow{a} y'$, then there is an $x' \in X$ with $x \xrightarrow{a} x'$ and $R(x',y')$.

4 Lax relation lifting and simulations

In the previous section we have seen how bisimulations were defined as coalgebras. We shall follow the same approach in this section for simulations. They are defined as coalgebras of a "lax relation lifting" functor $\mathrm{Rel}_{\sqsubseteq}(F)$ which is defined as a suitable combination of an order $\sqsubseteq$ on an endofunctor F and standard relation lifting.

Definition 4.1 For an endofunctor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ carrying a relation $\sqsubseteq$ (as in Definition 2.1) we define a lax relation operation $\mathrm{Rel}_{\sqsubseteq}(F)$ as:

$$\begin{aligned} R \longmapsto\; & \sqsubseteq \circ \mathrm{Rel}(F)(R) \circ \sqsubseteq \\ &= \{(u,v) \mid \exists u', v'.\, u \sqsubseteq u' \wedge (u',v') \in \mathrm{Rel}(F)(R) \wedge v' \sqsubseteq v\} \\ &= \{(u,v) \mid \exists w \in F(R).\, u \sqsubseteq F(r_1)(w) \wedge F(r_2)(w) \sqsubseteq v\}. \end{aligned}$$

In other terms,

$$\mathrm{Rel}_{\sqsubseteq}(F)(R) = \coprod\nolimits_{\langle \pi_1, \pi_3\rangle} \left(\langle \pi_1, F(r_1) \circ \pi_2\rangle^{-1}(\sqsubseteq_X) \wedge \langle F(r_2) \circ \pi_2, \pi_3\rangle^{-1}(\sqsubseteq_Y)\right),$$

as in the diagram below.

$$\begin{tikzcd} FX \times FX & FX \times FR \times FY \arrow[l, "\langle \pi_1, F(r_1) \circ \pi_2\rangle"'] \arrow[r, "\langle F(r_2) \circ \pi_2, \pi_3\rangle"] \arrow[d, "\langle \pi_1, \pi_3\rangle"] & FY \times FY \\ & FX \times FY & \end{tikzcd}$$

A **simulation** is then defined as a $\mathrm{Rel}_{\sqsubseteq}(F)$-coalgebra.

What we call lax relation lifting is called a relational extension in [8] and a (weak) relator in [15,2].

Lemma 4.2 *For F with order $\sqsubseteq$ as above we have:*

(i) *$Rel_{\sqsubseteq}(F)$ is a functor in commuting diagram:*

$$\begin{tikzcd}
\mathbf{Rel} \arrow[r, "Rel_{\sqsubseteq}(F)"] \arrow[d] & \mathbf{Rel} \arrow[d] \\
\mathbf{Sets} \times \mathbf{Sets} \arrow[r, "F \times F"'] & \mathbf{Sets} \times \mathbf{Sets}
\end{tikzcd}$$

(ii) $Rel_{\sqsubseteq}(F)(=) = \sqsubseteq$.

(iii) $R \subseteq S \Rightarrow Rel_{\sqsubseteq}(F)(R) \subseteq Rel_{\sqsubseteq}(F)(S)$.

(iv) $Rel_{\sqsubseteq}(F)(R^{op}) = Rel_{\sqsubseteq^{op}}(F)(R)^{op}$

(v) *Simulations are closed under arbitrary unions.*

(vi) *If R is a bisimulation, then both R and R^{op} are simulations.*

(vii) *For every $f\colon X \to Z$ and $g\colon Y \to W$,*

$$Rel_{\sqsubseteq}(F)\Big((f \times g)^{-1}(R)\Big) \subseteq (Ff \times Fg)^{-1}\Big(Rel_{\sqsubseteq}(F)(R)\Big).$$

(viii) *For every $f\colon X \to Z$ and $g\colon Y \to W$,*

$$\coprod_{Ff \times Fg}\Big(Rel_{\sqsubseteq}(F)(R)\Big) \subseteq Rel_{\sqsubseteq}(F)\Big(\coprod_{f \times g} R\Big).$$

Proof. We prove each claim in turn.

(i) Consider a morphism $R \to S$ in **Rel**, consisting of relations $R \subseteq X \times Y$ and $S \subseteq Z \times W$ with functions $f\colon X \to Z$ and $g\colon Y \to W$ between the underlying sets with $R(x, y) \Rightarrow S(f(x), g(y))$. Assuming $(u, v) \in \mathrm{Rel}_{\sqsubseteq}(F)(R)$ we have to prove that $(F(f)(u), F(g)(v)) \in \mathrm{Rel}_{\sqsubseteq}(F)(S)$. The assumption gives us $u' \in F(X)$ and $v' \in F(Y)$ with $u \sqsubseteq u'$, $(u', v') \in \mathrm{Rel}(F)(R)$ and $v' \sqsubseteq v$. Since $\sqsubseteq$ and $\mathrm{Rel}(F)$ are functors we then get $Ff(u) \sqsubseteq Ff(u')$, $(Ff(u'), Fg(v')) \in \mathrm{Rel}(F)(S)$ and $Fg(v') \sqsubseteq Fg(v)$. This establishes our goal.

(ii) Because:

$$\begin{aligned}
\mathrm{Rel}_{\sqsubseteq}(F)(=) &= \sqsubseteq \circ \mathrm{Rel}(F)(=) \circ \sqsubseteq \\
&= \sqsubseteq \circ = \circ \sqsubseteq \\
&= \sqsubseteq \circ \sqsubseteq \\
&= \sqsubseteq, \qquad \text{since } \sqsubseteq \text{ is transitive.}
\end{aligned}$$

(iii) Obvious, because ordinary relation lifting preserves inclusions.

(iv) Because:

$$
\begin{aligned}
&(u,v) \in \mathrm{Rel}_{\sqsubseteq}(F)(R^{\mathrm{op}}) \\
&\iff \exists u', v'.\, u \sqsubseteq u' \wedge (u',v') \in \mathrm{Rel}(F)(R^{\mathrm{op}}) \wedge v' \sqsubseteq v \\
&\iff \exists u', v'.\, u' \sqsubseteq^{\mathrm{op}} u \wedge (u',v') \in \mathrm{Rel}(F)(R)^{\mathrm{op}} \wedge v \sqsubseteq^{\mathrm{op}} v' \\
&\iff \exists u', v'.\, v \sqsubseteq^{\mathrm{op}} v' \wedge (v',u') \in \mathrm{Rel}(F)(R) \wedge u' \sqsubseteq^{\mathrm{op}} u \\
&\iff (v,u) \in \mathrm{Rel}_{\sqsubseteq^{\mathrm{op}}}(F)(R) \\
&\iff (u,v) \in \mathrm{Rel}_{\sqsubseteq^{\mathrm{op}}}(F)(R)^{\mathrm{op}}.
\end{aligned}
$$

(v) Since composition of relations and ordinary relation lifting preserve inclusions.

(vi) If R is a bisimulation then so is R^{op}, and hence R and R^{op} are simulations because $\sqsubseteq$ is reflexive.

(vii) Suppose that $\langle u,v\rangle \in \mathrm{Rel}_{\sqsubseteq}(F)((f\times g)^{-1}R)$. Then, there are u', v' such that

$$u \sqsubseteq u' \wedge \langle u',v'\rangle \in \mathrm{Rel}(F)((f\times g)^{-1}R) \wedge v' \sqsubseteq v.$$

Since relation lifting preserves inverse images, we see that

$$\langle u',v'\rangle \in (Ff \times Fg)^{-1}\mathrm{Rel}(F)(R),$$

i.e., $\langle Ff(u'), Fg(v')\rangle \in \mathrm{Rel}(F)(R)$. Thus,

$$Ff(u) \sqsubseteq Ff(u') \wedge \langle Ff(u'), Fg(v')\rangle \in \mathrm{Rel}(F)(R) \wedge Fg(v') \sqsubseteq Fg(v)$$

and so $\langle u,v\rangle \in (Ff\times Fg)^{-1}\mathrm{Rel}_{\sqsubseteq}(F)(R)$.

(viii) By (vii), we have

$$\mathrm{Rel}_{\sqsubseteq}(F)((f\times g)^{-1}\textstyle\coprod_{f\times g} R) \subseteq (Ff\times Fg)^{-1}\mathrm{Rel}_{\sqsubseteq}(F)(\textstyle\coprod_{f\times g} R),$$

and hence, since $\coprod_{f\times g} \dashv (f\times g)^{-1}$,

$$
\begin{aligned}
\textstyle\coprod_{Ff\times Fg} \mathrm{Rel}_{\sqsubseteq}(F)(R) &\subseteq \textstyle\coprod_{Ff\times Fg} \mathrm{Rel}_{\sqsubseteq}(F)((f\times g)^{-1}\textstyle\coprod_{f\times g}(R)) \\
&\subseteq \mathrm{Rel}_{\sqsubseteq}(F)(\textstyle\coprod_{f\times g} R).
\end{aligned}
$$

□

Example 4.3 We describe concrete simulations using the functors described in Examples 2.2 and 3.2.

(i) For two sequence coalgebras $X \xrightarrow{c} \mathcal{S}(X)$, $Y \xrightarrow{d} \mathcal{S}(Y)$ of the sequence functor $\mathcal{S}(X) = 1+(A\times X)$ a relation $R \subseteq X\times Y$ is a simulation iff for all $x\in X$ and $y \in Y$ with $R(x,y)$ we have $(c(x), d(y)) \in \mathrm{Rel}_{\sqsubseteq}(\mathcal{S})(R)$—where the order $\sqsubseteq$ is as described in Example 2.2 (ii). This means that there are u, v with $c(x) \sqsubseteq u$, $(u,v) \in \mathrm{Rel}(F)(R)$ and $v \sqsubseteq d(y)$. If $c(x) = *$ this yields no information, but if $c(x) = (a, x')$ we know that $u = (a, x')$, and so that $v = (a, y')$ with $R(x', y')$. But then $d(y) = (a, y')$. In conclusion, if $c(x) = (a, x')$ then $d(y) = (a, y')$ with $R(x', y')$.

(ii) For the list functor $\mathcal{L}(X) = X^\star$ we have seen two orderings $\sqsubseteq_1$ and $\sqsubseteq_2$ in Example 2.2 (iii). Hence for two list-functor coalgebras $X \xrightarrow{c} X^\star$, $Y \xrightarrow{d} Y^\star$ there are two associated notions of simulation. A relation $R \subseteq X \times Y$ is a simulation for $\sqsubseteq_1$ if $R(x, y)$ implies the following. If $c(x) = \langle x_0, \ldots, x_{n-1}\rangle$ and if $d(y) = \langle y_0, \ldots, y_{m-1}\rangle$, then there is a strictly monotone function $\varphi\colon \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, m-1\}$ with $R(x_i, y_{\varphi(i)})$ for each $i < n$.

For the second order $\sqsubseteq_2$ we would only have: $\forall i < n. \exists j < m. R(x_i, y_j)$.

(iii) For the bag functor $\mathcal{B}$ we only consider the first ordering $\sqsubseteq_1$ from Example 2.2 (iv). For two coalgebras $X \xrightarrow{c} \mathcal{B}(X)$, $Y \xrightarrow{d} \mathcal{B}(Y)$ a relation R is a simulation (wrt. $\sqsubseteq_1$) iff for all $x \in X$ and $y \in Y$ with $R(x, y)$, there is a $\gamma\colon R \to \mathbb{N}$ such that γ is zero almost everywhere and

- For each $x' \in X$, one has $c(x)(x') \leq \sum_{y'}\{\gamma(x', y') \mid R(x', y')\}$
- For each $y' \in Y$, one has $d(y)(y') \geq \sum_{x'}\{\gamma(x', y') \mid R(x', y')\}$.

(iv) Finally, for transition system coalgebras $X \xrightarrow{c} \mathcal{P}(A \times X)$, $Y \xrightarrow{d} \mathcal{P}(A \times Y)$, a relation $R \subseteq X \times Y$ is a simulation with respect to the inclusion iff it is a simulation in the usual sense: if $R(x, y)$, then $x \xrightarrow{a} x'$ implies there is an $y' \in Y$ with $y \xrightarrow{a} y'$ and $R(x', y')$.

5 Similarity

As a result of point (v) in Lemma 4.2 we can take, for given coalgebras, the union of all simulations. This relation is again a simulation, for which we shall write $\lesssim$. It will be called **similarity**.

Since the equality relation is a bisimulation, it is included in similarity. Hence similarity is a reflexive relation. In this section we shall look at properties (especially related to transitivity) and examples of similarity. The next section will concentrate on "two-way similarity", i.e., on $\lesssim \cap \lesssim^{\mathrm{op}}$.

Example 5.1 Transition system simulations, see Example 4.3 (iv), are related to trace inclusions in the following (standard) way. For a state x in a transition system with label set A we define:

$$\begin{aligned}
&\mathsf{trace}(x)\\
&\quad = \{\langle (x_0, a_0), (x_1, a_1), \ldots\rangle \in (X \times A)^\infty \mid x_0 = x \wedge \forall i \in \mathbb{N}. x_i \xrightarrow{a_i} x_{i+1}\}\\
&\mathsf{behtrace}(x)\\
&\quad = \{\pi_2^\infty(\sigma) \in A^\infty \mid \sigma \in \mathsf{trace}(x)\}.
\end{aligned}$$

Thus, the elements of $\mathsf{behtrace}(x)$ are the (finite or infinite) sequences of labels that may occur via transitions out of x.

Given a simulation R with $R(x, y)$, for each trace

$$\sigma = \langle (x_0, a_0), (x_1, a_1), \ldots \rangle \in \mathsf{trace}(x)$$

there is a $\tau = \langle (y_0, a_0), (y_1, a_1), \ldots \rangle \in \mathsf{trace}(y)$ with $R(x_i, y_i)$. We thus see that

$$x \lesssim y \Longrightarrow \mathsf{trace}(x) \subseteq \mathsf{trace}(y).$$

For this reason simulations form a standard ingredient of proofs of refinement (i.e., trace inclusion), where x is an initial state of an implementation, and y is an initial state of an abstract system (the specification) describing the appropriate behaviour.

What is special about the approach in this paper is that we take orderings on functors as primitive, and define lax relation lifting in terms of this order (and ordinary relation lifting, which is seen as canonical and taken for granted). In [8] such a lifting (or relational extension, as it is called there) is taken as primitive, subject to certain requirements. For a comparison we recall this approach. A relational extension (for a given endofunctor F) is a mapping G sending a relation $R \subseteq X \times Y$ to a relation $GR \subseteq FX \times FY$ such that:

(i) $=_{FX} \subseteq G(=_X)$

(ii) $R \subseteq S \Rightarrow GR \subseteq GS$

(iii) $GR \circ GS = G(R \circ S)$

(iv) "functoriality"

This last requirement is written out in detail, but amounts to the property that G is a functor $\mathbf{Rel} \to \mathbf{Rel}$ as in Lemma 4.2 (i). Interestingly, a "normal form" is proven in [8] (Lemma 1) showing that each relator can be described as a composite like in Definition 4.1, where the order $\sqsubseteq$ is $G(=)$. This shows that our approach—with a defined operation $\mathrm{Rel}_{\sqsubseteq}(F)$ instead of an assumed G—is more primitive.

However, the third requirement about preservation of composition is not automatic in our approach.

Definition 5.2 A functor F with ordering $\sqsubseteq$ is **composition-preserving** if the associated lax relation lifting $\mathrm{Rel}_{\sqsubseteq}(F)$ preserves composition of relations:

$$\mathrm{Rel}_{\sqsubseteq}(F)(R \circ S) = \mathrm{Rel}_{\sqsubseteq}(F)(R) \circ \mathrm{Rel}_{\sqsubseteq}(F)(S).$$

(The inclusion $\subseteq$ always holds, because ordinary relation lifting preserves compositions, and $\sqsubseteq$ is reflexive.)

A direct consequence of this requirement is that lax relation lifting restricts to a functor $\mathrm{Rel}_{\sqsubseteq}(F) \colon \mathbf{PreOrd} \to \mathbf{PreOrd}$.

In [15], it was shown that a functor F with order $\sqsubseteq$ is composition-preserving if, for all $f\colon Y \to X$, $g\colon Z \to X$,

$$\begin{aligned}\mathrm{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(=_X)) &= (Ff \times Fg)^{-1}\mathrm{Rel}_{\sqsubseteq}(F)(=_X) \\ &= (Ff \times Fg)^{-1}(\sqsubseteq_X)\end{aligned}$$

Note that, from Proposition 5.3 (vii), the inclusion $\subseteq$ always holds.

Here are some consequences of preservation of composition.

Proposition 5.3 *Let F be a functor with a composition-preserving ordering $\sqsubseteq$. Then:*

(i) *Simulations are closed under composition.*

(ii) *Similarity is a transitive relation.*

(iii) *For homomorphisms f, g between coalgebras,*

$$x \lesssim y \Longleftrightarrow f(x) \lesssim g(y).$$

(iv) *The similarity $\lesssim$ on the final coalgebra is the final $\mathrm{Rel}_{\sqsubseteq}(F)$-coalgebra.*

Proof. We prove each in turn.

(i) Obvious, because relation composition preserves inclusions.

(ii) Suppose $x \lesssim y$ and $y \lesssim z$. Then there are simulations R, S with $R(x, y)$ and $S(y, z)$. Hence $(S \circ R)(x, z)$, and so $x \lesssim z$ because $S \circ R$ is a simulation by (i). This shows $x \lesssim z$.

(iii) Since f is a homomorphism of coalgebras, its graph relation $\mathsf{Graph}(f)$ is a bisimulation. Hence both $\mathsf{Graph}(f)$ and $\mathsf{Graph}(f)^{\mathrm{op}}$ are simulations. This means that both $x \lesssim f(x)$ and $f(x) \lesssim x$. Similarly, $y \lesssim g(y)$ and $g(y) \lesssim y$. Hence we can easily prove the third point in the proposition, using the second:

($\Rightarrow$) If $x \lesssim y$, then $f(x) \lesssim x \lesssim y \lesssim g(y)$ so that $f(x) \lesssim g(y)$.

($\Leftarrow$) If $f(x) \lesssim g(y)$, then $x \lesssim f(x) \lesssim g(y) \lesssim y$, so that $x \lesssim y$.

(iv) Let R be a simulation over $A \xrightarrow{\alpha} F(A)$ and $B \xrightarrow{\beta} F(B)$ and let $!_A\colon A \to Z$ and $!_B\colon B \longrightarrow Z$ be the unique homomorphisms into the final F-coalgebra $\zeta\colon Z \xrightarrow{\cong} F(Z)$ and consider the diagram in Figure 1. By (iii), there is an arrow $R \to \lesssim$ in **Rel**, as shown. One must show that this arrow is a $\mathrm{Rel}_{\sqsubseteq}(F)$-homomorphism, i.e., that the top trapezoid commutes. This follows by the fact that $\mathrm{Rel}_{\sqsubseteq}(F)(\lesssim) \to FZ \times FZ$ is monic.

□

Example 5.4 We recall that the final coalgebra for the sequence functor $\mathcal{S}(X) = 1 + (A \times X)$ is the set A^∞ of finite and infinite sequences with

$$
\begin{tikzcd}
R \arrow[rrr] \arrow[dddd, tail] \arrow[dr, dotted] & & & \mathrm{Rel}_{\sqsubseteq}(F)(R) \arrow[dddd, tail] \arrow[dl, dotted] \\
& \lesssim \arrow[r] \arrow[dd, tail] & \mathrm{Rel}_{\sqsubseteq}(F)(\lesssim) \arrow[dd, tail] & \\
& & & \\
& Z \times Z \arrow[r] & FZ \times FZ & \\
A \times B \arrow[rrr] \arrow[ur] & & & FA \times FB \arrow[ul]
\end{tikzcd}
$$

Fig. 1. Similarity on the final coalgebra is the final $\mathrm{Rel}_{\sqsubseteq}(F)$-coalgebra

coalgebra structure $A^\infty \xrightarrow{\cong} 1 + (A \times A^\infty)$ given by

$$\sigma \longmapsto \begin{cases} * & \text{if } \sigma \text{ is the empty sequence } \langle\rangle \\ (a, \sigma') & \text{if } \sigma = a \cdot \sigma' \text{ with head } a \text{ and tail } \sigma'. \end{cases}$$

This set of sequences A^∞ carries the usual "prefix" order:

$$\sigma \leq \tau \iff \sigma \cdot \rho = \tau, \text{ for some } \rho \in A^\infty.$$

We claim that this prefix order is the same as similarity.

The inclusion $\leq \subseteq \lesssim$ is easy, because $\leq$ is a simulation: if $\sigma \leq \tau$, say via $\sigma \cdot \rho = \tau$, and $\sigma = a \cdot \sigma'$, then $\tau = a \cdot \tau'$ where $\sigma' \cdot \rho = \tau'$. This shows $\sigma' \leq \tau'$.

For the reverse inclusion $\lesssim \subseteq \leq$ we assume $\sigma \lesssim \tau$, say via a simulation $R \subseteq A^\infty \times A^\infty$ with $R(\sigma, \tau)$. We determine elements $a_0, a_1, \ldots \in A$ and $\sigma_0, \sigma_1, \ldots \in A^\infty$ with for each n, $\sigma = a_0 \cdot a_1 \cdots a_n \cdot \sigma_n$. By induction we find $\tau_0, \tau_1, \ldots \in A^\infty$ with for each n, $\tau = a_0 \cdot a_1 \cdots a_n \cdot \tau_n$. There are two cases:

- σ is finite, say, $\sigma = a_0 \cdots a_n$. Then $\tau = \sigma \cdot \tau_n$, so that $\sigma \leq \tau$.
- σ is infinite. Then $\sigma = \tau$, and thus also $\sigma \leq \tau$.

As a consequence of Proposition 5.3 (iii) we now have for arbitrary sequence coalgebras $X \xrightarrow{c} \mathcal{S}(X)$, $Y \xrightarrow{d} \mathcal{S}(Y)$ and elements $x \in X, y \in Y$,

$$x \lesssim y \iff !(x) \leq !(y),$$

where ! is the unique homomorphism to the final coalgebra and $\leq$ is its prefix order.

6 Two-way similarity

Having seen similarity $\lesssim$, we define **two-way similarity** as $\eqsim = \lesssim \cap \lesssim^{\mathrm{op}}$, i.e., as:

$$x \eqsim y \overset{\text{def}}{\iff} x \lesssim y \text{ and } y \lesssim x.$$

An immediate consequence of Lemma 4.2 (vi) is that bisimilarity implies two-way similarity: $\underline{\leftrightarrow} \subseteq \eqsim$. In this section we are interested in the converse, i.e., in whether or not $\eqsim \subseteq \underline{\leftrightarrow}$. The next examples show that this may or may not be the case.

Example 6.1 We give an example in which $\eqsim \subseteq \underline{\leftrightarrow}$, and one in which the inclusion fails.

(i) Let's consider the sequence example, with two coalgebras $X \xrightarrow{c} 1+(A\times X)$ and $Y \xrightarrow{d} 1+(A\times Y)$. Assume $x \eqsim y$. Then there are simulations $R \subseteq X \times Y$ and $S \subseteq Y \times X$ with $R(x,y)$ and $S(y,x)$. The fact that R and S are simulations means that for all $z \in X, w \in Y$:
(a) $R(z,w)$ and $c(z) = (a,z')$ implies $d(w) = (a,w')$ with $R(z',w')$.
(b) $S(w,z)$ and $d(w) = (a,w')$ implies $c(z) = (a,z')$ with $S(w',z')$.
We claim that $T = (R \cap S^{\mathrm{op}}) \subseteq X \times Y$ is a bisimulation with $T(x,y)$. The last point is obvious. In order to show that T is a bisimulation, assume $T(z,w)$. Then:
- If $c(z) = *$ but $d(w) = (a,w')$, then we get a contradiction by (2) above. Hence $d(w) = *$. The reverse implication is obtained similarly.
- If $c(z) = (a,z')$, then $d(w) = (a,w')$ with $R(z',w')$, by (1). Applying (2) yields that $c(z) = (a,z'')$ with $S(w',z'')$. But then we get $z' = z''$, so that $T(z',w')$, as required.

(ii) Here is a simple variation on the previous example. Let $F(X) = X + X$ with order $\sqsubseteq$ given by:

$$u \sqsubseteq v \iff \forall x \in X.\, u = \kappa_2(x) \Rightarrow v = \kappa_2(x).$$

Notice that no relation is required in case u is in the first (left) component of $X + X$.

The associated notion of similarity says, for given coalgebras $c\colon X \to X + X$ and $d\colon Y \to Y + Y$, that $R \subseteq X \times Y$ is a simulation if for each x, y with $R(x,y)$ one has that if $c(x) = \kappa_2(x')$, then $d(y)$ must be of the form $\kappa_2(y')$ with $R(x',y')$. In case we have a two-way similarity there must also be a relation S with $S(y,x)$ implies that $d(y) = \kappa_2(y')$, then $c(x) = \kappa_2(x')$ with $S(y',x')$.

But this is not the same as bisimilarity for this functor, because then we must also have a relation in the first components of the coproduct $+$: $R \subseteq X \times Y$ is a bisimulation if $R(x,y)$ implies both:
- if $c(x) = \kappa_1(x')$, then $d(y) = \kappa_1(y')$ with $R(x',y')$;
- if $c(x) = \kappa_2(x')$, then $d(y) = \kappa_2(y')$ with $R(x',y')$.

In the second example we see that there is something missing from the relation $\sqsubseteq$ that ensures that two-way similarity implies bisimilarity. The following result gives a sufficient condition.

Theorem 6.2 *Let F be a functor with a relation $\sqsubseteq$ such that the associated relation liftings satisfy the condition:*

$$Rel_{\sqsubseteq}(F)(R_1) \cap Rel_{\sqsubseteq^{op}}(F)(R_2) \subseteq Rel(F)(R_1 \cap R_2).$$

Then two-way similarity (for coalgebras of this functor) is the same as bisimilarity:

$$x \underline{\leftrightarrow} y \Longleftrightarrow x \precsim\!\!\!\succsim y.$$

Proof. We only need to prove the direction ($\Leftarrow$), and so we assume $x \precsim\!\!\!\succsim y$, say via simulations R, S with $R(x, y)$ and $S(y, x)$. The fact that R, S are simulations says that $R \subseteq (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(R)\big)$ and $S \subseteq (d \times c)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(S)\big)$. We take as new relation $T = (R \cap S^{\mathrm{op}})$, like in Example 6.1 (i). Clearly, $T(x, y)$. We are done if we can show that T is a bisimulation, i.e., satisfies $T \subseteq (c \times d)^{-1}\big(\mathrm{Rel}(F)(T)\big)$. But:

$$\begin{aligned} S^{\mathrm{op}} &\subseteq (d \times c)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(S)\big)^{\mathrm{op}} \\ &= (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(S)^{\mathrm{op}}\big) \\ &= (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq^{\mathrm{op}}}(F)(S^{\mathrm{op}})\big) \qquad \text{by Lemma 4.2 (iv).} \end{aligned}$$

Hence:

$$\begin{aligned} T \;&=\; (R \cap S^{\mathrm{op}}) \\ &\subseteq (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(R)\big) \,\cap\, (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq^{\mathrm{op}}}(F)(S^{\mathrm{op}})\big) \\ &= (c \times d)^{-1}\big(\mathrm{Rel}_{\sqsubseteq}(F)(R)\big) \,\cap\, \mathrm{Rel}_{\sqsubseteq^{\mathrm{op}}}(F)(S^{\mathrm{op}})\big) \\ &\subseteq (c \times d)^{-1}\big(\mathrm{Rel}(F)(T)\big). \end{aligned}$$

The last step uses the condition of the theorem. $\square$

Notice that the condition in this theorem can be formulated because we take an order $\sqsubseteq$ on a functor as primitive (and not the resulting relation lifting): this allows us to change the order (by taking the opposite $\sqsubseteq^{\mathrm{op}}$) and consider the associated lifting.

Example 6.3 In this example we show that the first ordering $\sqsubseteq_1$ for the list functor $\mathcal{L}$ in Example 2.2 satisfies the condition of the previous theorem.

Assume two sequences $u = \langle x_0, \ldots, x_{n-1}\rangle$ and $v = \langle y_0, \ldots, y_{m-1}\rangle$ satisfy $(u, v) \in \mathrm{Rel}_{\sqsubseteq_1}(\mathcal{L})(R_1) \cap \mathrm{Rel}_{\sqsubseteq_1^{\mathrm{op}}}(\mathcal{L})(R_2)$. This means that there are strictly monotone functions

$$\begin{aligned} &\varphi\colon \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, m-1\}, \\ &\psi\colon \{0, 1, \ldots, m-1\} \to \{0, 1, \ldots, n-1\} \end{aligned}$$

with $R_1(x_i, y_{\varphi(i)})$ and $R_2(x_{\psi(j)}, y_j)$. But this can only happen if $n = m$ and $\varphi = \psi = id$. Hence $(R_1 \cap R_2)(x_i, y_i)$, so that $(u, v) \in \mathrm{Rel}(\mathcal{L})(R_1 \cap R_2)$.

Example 6.4 For (labeled) transition systems it is *not* the case that two-way similarity is the same as bisimilarity. Here is a simple (unlabeled) example.

$$\begin{tikzcd}
& 1 \arrow[dl] \arrow[dr] & & a \arrow[d] \\
2 \arrow[d] & & 3 & b \arrow[d] \\
4 & & & c
\end{tikzcd}$$

A simulation from left to right is:

$$R = \{(1, a), (2, b), (3, b), (4, c)\}.$$

Indeed, $R(x, y)$ and $x \longrightarrow x'$ implies $y \longrightarrow y'$ for some y' with $R(x', y')$.

And a simulation from right to left is:

$$S = \{(a, 1), (b, 2), (c, 4)\}.$$

This shows that $1 \eqsim a$. But we do not have $1 \underline{\leftrightarrow} a$.

7 Dcpo structure by finality

In Example 5.4 we have seen that similarity on the final coalgebra of sequences coincides with the prefix order. The latter happens to provide a dcpo structure: every directed subset has a join. In this section we shall see that this dcpo structure results from a distributive law between the sequence functor and the free dcpo monad on preorders. We start with the latter.

We write **Dcpo** for the category of directed complete preorders. It comes with a forgetful functor $\mathcal{U} \colon \mathbf{Dcpo} \to \mathbf{PreOrd}$. This functor has a left adjoint, for which we write $\mathcal{D}$. It maps a preorder to its directed downsets, ordered by inclusion. The join in $\mathcal{D}(X)$ of a directed collection $(U_i)_{i \in I}$ of directed downsets U_i is then simply their union $\bigcup_{i \in I} U_i$. The adjunction induces a monad on **PreOrd**, for which we shall also write $\mathcal{D}$, with:

$$\begin{array}{rll}
\text{unit:} & \eta_X \colon X \to \mathcal{D}(X) & x \longmapsto {\downarrow} x \\
\text{multiplication:} & \mu_X \colon \mathcal{D}^2(X) \to \mathcal{D}(X) & (U_i)_{i \in I} \longmapsto \bigcup_{i \in I} U_i.
\end{array}$$

The following result is standard.

Lemma 7.1 *For a preorder X the following are equivalent.*

(i) *X is a dcpo;*

(ii) *X carries an (Eilenberg-Moore) algebra structure for the monad $\mathcal{D}$;*

(iii) *the unit $\eta_X \colon X \to \mathcal{D}(X)$ has a left adjoint.*

The structure map in (ii) and (iii) is of course the join operation

$$\bigvee : \mathcal{D}(X) \to X.$$

Successive left adjoints to the unit are studied in [9] and describe continuity and algebraicity in the dcpo.

Theorem 7.2 *Let* $F\colon \mathbf{Sets} \to \mathbf{Sets}$ *have a composition-preserving order* $\sqsubseteq$ *for which there is a distributive law*

$$\begin{tikzcd} \mathbf{PreOrd} \arrow[r, "\mathcal{D}"] \arrow[d, "Rel_{\sqsubseteq}(F)"'] \arrow[dr, phantom, "\nearrow\!\!\!\nearrow"] & \mathbf{PreOrd} \arrow[d, "Rel_{\sqsubseteq}(F)"] \\ \mathbf{PreOrd} \arrow[r, "\mathcal{D}"'] & \mathbf{PreOrd} \end{tikzcd}$$

consisting of monotone functions $\tau_X\colon \mathcal{D}(FX) \to F(\mathcal{D}(X))$*, where* $\mathcal{D}(FX)$ *carries the inclusion order* $\subseteq$ *on the completion* $\mathcal{D}(FX) = \mathcal{D}(FX, Rel_{\sqsubseteq}(F)(\leq))$*, and* $F(\mathcal{D}(X))$ *carries the lifting* $Rel_{\sqsubseteq}(F)(\subseteq)$ *of the inclusion order* $\subseteq$ *on* $\mathcal{D}(X) = \mathcal{D}(X, \leq)$*. This distributive law is required to make the following two diagrams commute.*

$$\begin{tikzcd} F(X) \arrow[r, "\eta_{F(X)}"] \arrow[dr, "F(\eta_X)"'] & \mathcal{D}(F(X)) \arrow[d, "\tau_X"] \\ & F(\mathcal{D}(X)) \end{tikzcd} \qquad \begin{tikzcd} \mathcal{D}^2(F(X)) \arrow[r, "\mathcal{D}(\tau_X)"] \arrow[d, "\mu_{F(X)}"'] & \mathcal{D}(F(\mathcal{D}(X))) \arrow[r, "\tau_{\mathcal{D}(X)}"] & F(\mathcal{D}^2(X)) \arrow[d, "F(\mu_X)"] \\ \mathcal{D}(F(X)) \arrow[rr, "\tau_X"'] & & F(\mathcal{D}(X)) \end{tikzcd}$$

If F *has a final coalgebra, then it forms with its similarity order a dcpo.*

Such a dcpo structure can be used in a denotational semantics of a programming language, to give meaning to constructs like loops or recursion.

Proof. Let $\zeta\colon Z \xrightarrow{\cong} F(Z)$ be the final coalgebra. We define an (Eilenberg-Moore) algebra structure $\bigvee\colon \mathcal{D}(Z) \to Z$ by finality, in:

$$\begin{tikzcd} F(\mathcal{D}(Z)) \arrow[r, "F(\bigvee)"] & F(Z) \\ \mathcal{D}(F(Z)) \arrow[u, "\tau_Z"] & \\ \mathcal{D}(Z) \arrow[u, "\mathcal{D}(\zeta)"] \arrow[r, "\bigvee"'] & Z \arrow[uu, "\cong"', "\zeta"] \end{tikzcd}$$

Note that $\bigvee$ is monotone by Proposition 5.3. We have to verify the laws for Eilenberg-Moore algebras: $\bigvee \circ \eta_Z = id$ and $\bigvee \circ \mu_Z = \bigvee \circ \mathcal{D}(\bigvee)$. Both equations follow from uniqueness. The first one holds because the unit is a homomorphism of coalgebras:

$$\begin{tikzcd}
F(Z) \arrow[r, "F(\eta_Z)"] \arrow[dr, "\eta_{F(Z)}"'] & F(\mathcal{D}(Z)) \\
 & \mathcal{D}(F(Z)) \arrow[u, "\tau_Z"'] \\
Z \arrow[uu, "\zeta"] \arrow[r, "\eta_Z"'] & \mathcal{D}(Z) \arrow[u, "\mathcal{D}(\zeta)"']
\end{tikzcd}$$

The lower-left diagram commutes by naturality, and the upper-right one is the first distributivity requirement mentioned above. The composite $\bigvee \circ \eta_Z$ is then a homomorphism $\zeta \to \zeta$, and must thus be the identity.

In a similar way one proves that both $\bigvee \circ \mu_Z$ and $\bigvee \circ \mathcal{D}(\bigvee)$ are coalgebra homomorphisms from the coalgebra $\tau_{F(Z)} \circ \mathcal{D}(\tau_Z) \circ \mathcal{D}^2(\zeta) \colon \mathcal{D}^2(Z) \to F(\mathcal{D}^2(Z))$ to the final coalgebra. $\square$

The dcpo structure on sequences in Example 5.4 indeed comes from a distributive law as above. We define a transformation

$$\mathcal{D}(1 + (A \times X)) \xrightarrow{\tau_X} 1 + (A \times \mathcal{D}(X))$$

as follows. Let S be a directed downset of $1 + (A \times X)$. Then

$$\tau_X(S) = \begin{cases} (a, \{x \mid (a, x) \in S\}) & \text{if } \{x \mid (a, x) \in S\} \neq \emptyset \\ * & \text{else.} \end{cases}$$

It follows from directedness that this is well-defined. We omit the proof that the necessary diagrams commute.

Actually, the definition via finality of the join for sequences occurs already in [6], but here we put this definition in a wider context via distributive laws.

We consider another such example.

Example 7.3 We fix a set V, and think of its elements as variables. We use V in the functor $\mathcal{T}_V \colon \mathbf{Sets} \to \mathbf{Sets}$ given by

$$\mathcal{T}_V(X) = 1 + \Big(V^\star \times V \times X^\star\Big)$$

We shall write the final coalgebra as $\zeta \colon \mathsf{BT} \xrightarrow{\cong} 1 + (V^\star \times V \times \mathsf{BT}^\star)$. Its elements will be considered as (abstract) Böhm trees, see [3]. For $A \in \mathsf{BT}$ we can write:

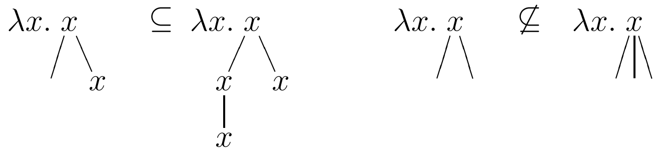
$$\zeta(A) = * \qquad \text{or} \qquad \zeta(A) = \begin{pmatrix} \lambda x_1 \ldots x_n.\, y & & \\ \zeta(A_1) & \cdots & \zeta(A_m) \end{pmatrix}$$

where, on the right, $\zeta(A) = (\langle x_1, \ldots, x_n \rangle, y, \langle A_1, \ldots, A_m \rangle)$. The '$\lambda$' is just syntactic sugar, used to suggest the analogy with the standard notation for Böhm trees [3]. The elements of BT are thus finitely branching, possibly infinite rooted trees, with labels of the form $\lambda x_1 \ldots x_n.\, y$, for variables $x_i, y \in V$.

The order considered on Böhm trees as formulated in [3, §10.2] is:

$$A \subseteq B \iff A \text{ results from } B \text{ by cutting of some subtrees}$$

This description is fairly informal. The question is how to make it precise, via an order on the functor $\mathcal{T}_V$. Two possible orders come to mind: the flat order from Example 2.2 (ii) or the precise order $\sqsubseteq_1$ on the list functor from Example 2.2 (iii). The following illustrations from [3, §10.2] help.



These pictures show that "cutting off subtrees" should be interpreted as: replacing a node by $*$. Thus, the order $\sqsubseteq$ that we consider on the functor $\mathcal{T}_V$ is simply the flat order, like for sequences in Example 2.2 (ii): $u \sqsubseteq v$ iff $u \neq * \Rightarrow u = v$.

The induced similarity order $\lesssim$ on BT is then the above order $\subseteq$. The previous theorem allows us to conclude that it is a dcpo.

These and other examples can be readily constructed, using the following simple results, showing that distributive laws can be constructed by induction on the structure of polynomial functors.

- For any dcpo $(A, \leq_A)$, the constant functor $FX = A$ with order $\sqsubseteq_{FX} = \leq_A$ has a distributive law given by $\bigvee \colon \mathcal{D}A \to A$. In particular, this applies when we take $\leq_A$ to be $=_A$.
- For the identity functor $FX = X$ with the discrete order, the identity transformation $\mathcal{D}X \to \mathcal{D}X$ is a distributive law.
- Suppose that functors F_1 and F_2 have distributive laws τ_1 and τ_2, and define an order $\sqsubseteq$ on $F_1 \times F_2$ by taking the orders on F_1 and F_2 component-wise. Then

$$((\tau_1)_X \circ \mathcal{D}\pi_1, (\tau_2)_X \circ \mathcal{D}\pi_2) \colon \mathcal{D}(F_1 \times F_2)X \to (F_1 \times F_2)\mathcal{D}X$$

  is a distributive law.
- Let F_1 and F_2 be as above, and define $\sqsubseteq_{F_1+F_2}$ as the disjoint union of $\sqsubseteq_{F_1}$

and $\sqsubseteq_{F_2}$. Then,

$$\begin{aligned} D(F_1+F_2)X &\longrightarrow (F_1+F_2)DX \\ S &\longmapsto \begin{cases} (\tau_1)_X S \text{ if } S \subseteq F_1 X \\ (\tau_2)_X S \text{ else.} \end{cases} \end{aligned}$$

defines a distributive law for $F_1 + F_2$ with the given order. More relevant for our examples (especially when $F_1 = 1$, like for sequences), if we define $\sqsubseteq'_{F_1+F_2}$ so that

$$a \sqsubseteq'_{F_1+F_2} b \text{ iff } a \sqsubseteq_{F_1+F_2} b \text{ or } a \in F_1 X \text{ and } b \in F_2 X,$$

then there is a related distributive law given by

$$\begin{aligned} D(F_1+F_2)X &\longrightarrow (F_1+F_2)DX \\ S &\longmapsto \begin{cases} (\tau_1)_X S & \text{if } S \subseteq F_1 X \\ (\tau_2)_X (S \cap F_2 X) & \text{else.} \end{cases} \end{aligned}$$

8 Conclusion and open questions

This paper contains a fresh, systematic approach to simulations within the theory of coalgebras. It contains basic notions, examples, and results. There are many remaining open research questions, of which we list a few.

(i) The final coalgebras of sequences and Böhm trees are not just dcpos, but *algebraic* dcpos, where each element is a directed join of the "finite" elements below it. Does this algebraicity also follow from a general requirement? See also [1], where a special order on a final coalgebra (as limit) is introduced. Relations to our work should be elaborated.

(Finiteness of states of a coalgebra can be defined in general via least fixed point of predicate lifting, as described in [10].)

(ii) Simulations have been studied in relation to modal logics, see *e.g.* [2]. What is then the relation between ordering on functors as introduced here and results in the modal logic associated with these functors?

(iii) It is not clear if the preservation of composition property, as described in Definition 5.2, is a real requirement. We are not aware of a non-example.

(iv) Given an arbitrary Böhm tree $A \in \mathsf{BT}$, we can easily get a new tree $\lambda x.\, A \in \mathsf{BT}$ via λ-abstraction:

$$\lambda x.\, A = \begin{cases} \zeta^{-1}(*) & \text{if } \zeta(A) = \bot \\ \zeta^{-1}(x \cdot \alpha, y, \beta) & \text{if } \zeta(A) = (\alpha, y, \beta). \end{cases}$$

But the question is how to define the associated application operation $\cdot : \mathsf{BT} \times \mathsf{BT} \to \mathsf{BT}$ in a smooth way. In [3, Definition 18.3.2] a trick is used by first translating finite Böhm trees into λ-terms, applying them as terms, and then translating them back into Böhm trees. For arbitrary Böhm trees, application is then defined in such a way via their finite approximants. The underlying problem is that application of Böhm trees may involve β-reduction.

References

[1] J. Adámek. Final Coalgebras are Ideal Completions of Initial Algebras. *Journ. of Logic and Computation*, 12(2):217–242, 2002.

[2] A. Baltag. A logic for coalgebraic simulation. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, number 33 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2000.
`www.elsevier.nl/locate/entcs/volume33.html`.

[3] H.P. Barendregt. *The Lambda Calculus. Its Syntax and Semantics.* North-Holland, Amsterdam, 2^{nd} rev. edition, 1984.

[4] A. Carboni, G.M. Kelly, and R.J. Wood. A 2-categorical approach to change of base and geometric morphisms I. Report 90-1, Dep. of Pure Mathematics, Univ. of Sydney, 1990.

[5] M.P. Fiore. A coinduction principle for recursive data types based on bisimulation. *Inf. & Comp.*, 127(2):186–198, 1996.

[6] U. Hensel and B. Jacobs. Coalgebraic theories of sequences in PVS. *Journ. of Logic and Computation*, 9(4):463–500, 1999.

[7] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. & Comp.*, 145:107–152, 1998.

[8] W.H. Hesselink and A. Thijs. Fixpoint semantics and simulation. *Theor. Comp. Sci.*, 238:275–311, 2000.

[9] B. Jacobs. Coalgebras and approximation. In A. Nerode and Yu. V. Matiyasevich, editors, *Logical Foundations of Computer Science*, number 813 in Lect. Notes Comp. Sci., pages 173–183. Springer, Berlin, 1994.

[10] B. Jacobs. Exercises in coalgebraic specification. In R. Crole R. Backhouse and J. Gibbons, editors, *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction*, number 2297 in Lect. Notes Comp. Sci., pages 237–280. Springer, Berlin, 2002.

[11] N. Lynch and F. Vaandrager. Forward and backward simulations. I. Untimed systems. *Inf. & Comp.*, 121(2):214–233, 1995.

[12] A.M. Pitts. A co-induction principle for recursively defined domains. *Theor. Comp. Sci.*, 124(2):195–219, 1994.

[13] J. Rutten. Relators and metric bisimulations. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, number 11 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 1998. `www.elsevier.nl/locate/entcs/volume11.html`.

[14] J. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.

[15] A. Thijs. *Simulation and Fixpoint Semantics*. PhD thesis, Univ. Groningen, 1996.

[16] K. Worytkiewicz. Paths and simulations. In *Category Theory and Computer Science 2002*, number 69 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2002.
`www.elsevier.nl/locate/entcs/volume69.html`.