Theoretical Informatics and Applications
Theoret. Informatics Appl. **35** (2001) 1–29

INTEGRATING OBSERVATIONAL AND COMPUTATIONAL FEATURES IN THE SPECIFICATION OF STATE-BASED, DYNAMICAL SYSTEMS *

Corina Cîrstea[1]

Abstract. We present an abstract equational framework for the specification of systems having both observational and computational features. Our approach is based on a clear separation between the two categories of features, and uses algebra, respectively coalgebra to formalise them. This yields a coalgebraically-defined notion of observational indistinguishability, as well as an algebraically-defined notion of reachability under computations. The relationship between the computations yielding new system states and the observations that can be made about these states is specified using liftings of the coalgebraic structure of state spaces to a coalgebraic structure on computations over these state spaces. Also, correctness properties of system behaviour are formalised using equational sentences, with the associated notions of satisfaction abstracting away observationally indistinguishable, respectively unreachable states, and with the resulting proof techniques employing coinduction, respectively induction.

Mathematics Subject Classification. 18C10, 03C65, 68Q55, 68N30.

Introduction

State-based, dynamical systems comprise a computational aspect, concerned with the computations yielding new system states and with the reachability of states under computations, and an observational aspect, concerned with the observations that can be made about existing system states and with the indistinguishability of states by observations. These two aspects overlap, in that features concerned with the evolution of system states can be regarded both as a means

* *Research supported by St. John's College, Oxford.*

[1] Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.; e-mail: `corina.cirstea@comlab.ox.ac.uk`

to compute new states and as a means to observe existing states. There exist, however, system features whose nature is either purely computational or purely observational, with the construction of initial states and respectively the extraction of visible information from system states being instances of such features.

Existing approaches to system specification typically exploit the overlap between computational and observational features to employ either algebraic [7, 9, 10, 16] or coalgebraic techniques [5, 12, 13, 15] for specification and reasoning. (Exceptions include [11,14].) Such a choice limits the expressiveness of these formalisms w.r.t. either observational or computational features. In particular, observers with structured result types can not be accommodated by algebraic approaches, whereas constructors with structured argument types can not be accommodated by coalgebraic approaches. Furthermore, in the presence of constructors with multiple arguments, additional constraints are needed to guarantee that observational equivalence relations (defined in terms of subsignatures of observers) are preserved by such constructors. These constraints involve either restrictions on the algebras used to model the specified systems [7,10,11], or restrictions on the specifications used to describe system behaviour [16]. Finally, existing approaches to system specification (including [11,14]) do not consider ensuring that the system observers preserve the reachability of states under computations (with the notion of reachability being defined in terms of a subsignature of constructors). This is equally important, as one expects states of subsystems of a given system to be reachable whenever they are obtained by observing reachable system states.

The present paper[2] aims to fully exploit the expressive power of algebra and coalgebra when specifying purely computational and respectively purely observational structures, and to combine their complementary contributions when specifying structures that have both computational and observational features, in a manner which guarantees a certain compatibility between the two categories of features. A first step towards achieving the aim was taken in [2], where a coalgebraic, equational formalism for the specification of observational structures allowing for a choice in the result type of observations was developed. The duality between the structures considered in [2] and those specifiable in many-sorted algebra is reflected in the resulting formalism, which employs notions of covariable, coterm and coequation (dual to the standard ones of variable, term and equation) for specification and reasoning. The approach in [2] is here generalised to an abstract setting, with the resulting framework also subsuming other existing equational approaches to system specification, including [5, 12] and (a restricted version of) [9]. Furthermore, dualising this approach yields an abstract framework for the specification of structures that involve computation. The two approaches are then integrated in order to obtain a specification framework for systems having both computational and observational features. This integration builds on work in [18] on relating operational and denotational semantics. Following [18], liftings of the coalgebraic structure of state spaces to computations over these state spaces are used to interpret computations on the state spaces induced by

[2]An earlier version of this paper is [3].

the observational component. (A similar approach to integrating algebraic and coalgebraic features is considered in [6], where liftings of algebraic structures to transition systems over these structures are used to define transition structures on one-sorted algebras. The approach presented here is dual to the one in [6], but our setting is more abstract than the one of [6].) Equational sentences are then used to formalise correctness properties of system behaviour (referring either to the indistinguishability of computations by observations, or to the satisfaction of state invariants in states reachable under computations), with the associated proof techniques employing coinduction and respectively induction.

The paper is structured as follows. Section 1 introduces a coalgebraic equational framework for the specification of observational structures. Section 2 derives an (essentially dual) algebraic framework for the specification of computational structures. Section 3 integrates the two frameworks in order to account for structures having both an observational and a computational component. Section 4 briefly summarises the results presented. A specification of stacks of natural numbers is used as a running example.

1. Specifying observational structures

Reference [17] presents a general coalgebraic framework for the specification of state-based, dynamical systems, with arbitrary endofunctors on Set being used to specify system behaviour, and with coalgebras of such endofunctors providing (abstractions of) particular implementations of the specified behaviours. The approach in [17] is here specialised in order to give a categorical account of *equational* coalgebraic approaches to specification[3]. A framework which unifies some of the existing equational approaches to system specification, including [2, 5, 9, 12], is introduced in the following. The framework involves notions of *abstract cosignature*, used to specify particular kinds of observational structures, *coalgebra of a cosignature*, used to provide a particular interpretation for the structure specified by the cosignature, *observer over a cosignature*, used to extract information from the coalgebras of the cosignature according to their particular interpretation for the specified structure, and *coequation over a cosignature*, used to constrain the coalgebras of the cosignature by requiring different observers to yield similar (either equal or just observationally equal) results on the same coalgebra. In addition, a notion of (*horizontal*) *cosignature morphism* is used to specify a change in the type of information being observed, and this is shown to yield an institution w.r.t. the satisfaction (up to observability) of coequations by coalgebras.

We begin by noting that the Set-theoretic notions of bisimulation, subcoalgebra, homomorphic image and covariety generalise to endofunctors on arbitrary categories as follows.

[3]Our setting is, however, more abstract than the one in [17], as it involves endofunctors on arbitrary categories.

Definition 1.1. Let $\mathsf{G} : \mathsf{C} \to \mathsf{C}$ denote an arbitrary endofunctor. A **G-coalgebra** is a tuple $\langle C, \gamma \rangle$, with C a C-object and $\gamma : C \to \mathsf{G}C$ a C-arrow, while a **G-coalgebra homomorphism** between G-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ is a C-arrow $f : C \to D$ additionally satisfying $\delta \circ f = \mathsf{G}f \circ \gamma$. A **$\mathsf{G}$-bisimulation** between G-coalgebras $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ is a *relation* (see [1], p. 101) $\langle R, r_1, r_2 \rangle$ between C and D in C such that there exists a G-coalgebra structure $\langle R, \rho \rangle$ on R making r_1 and r_2 into G-coalgebra homomorphisms from $\langle R, \rho \rangle$ to $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$ respectively. The largest[4] G-bisimulation between $\langle C, \gamma \rangle$ and $\langle D, \delta \rangle$, if it exists, is called **G-bisimilarity**. Also, a **G-subcoalgebra** of $\langle C, \gamma \rangle$ is an object $\langle \langle D, \delta \rangle, d \rangle$ of the category $\mathsf{Coalg}(\mathsf{G})/\gamma$[5], such that the C-arrow underlying d is a monomorphism, while a **homomorphic image** of $\langle C, \gamma \rangle$ is an object $\langle \langle D, \delta \rangle, e \rangle$ of the category $\mathsf{Coalg}(\mathsf{G})\backslash\gamma$[6], such that the C-arrow underlying e is an epimorphism. Classes of G-coalgebras which are closed under subcoalgebras, homomorphic images and coproducts are called **covarieties**.

We also recall that, for an endofunctor $\mathsf{G} : \mathsf{C} \to \mathsf{C}$, the functor taking G-coalgebras to their carrier creates colimits, as well as any limits which are preserved by G.

We now introduce an abstract syntax for specifying observational structures.

Definition 1.2. An **(abstract) cosignature** is a pair (C, G), with C a category which is complete, cocomplete and has $(\mathsf{Epi}(\mathsf{C}), \mathsf{Mono}(\mathsf{C}))$ as a *factorisation system*[7], and with $\mathsf{G} : \mathsf{C} \to \mathsf{C}$ an endofunctor which preserves pullbacks[8] and limits of ω^{op}-chains.

Polynomial endofunctors on categories of sorted sets satisfy the conditions in the definition of abstract cosignatures. However, powerset endofunctors do not satisfy these conditions. (A generalisation of the notion of abstract cosignature which accounts for powerset endofunctors of bounded cardinality is discussed later in this section.)

Abstract cosignatures specify the type of information that can be observed about particular systems. The coalgebras of the endofunctors in question then provide (abstractions of) specific system implementations.

Definition 1.3. Let (C, G) denote an abstract cosignature. A **(C, G)-coalgebra** (**coalgebra homomorphism**) is a G-coalgebra (coalgebra homomorphism).

---

[4]w.r.t. an ordering $\leq$ on relations, see [1].

[5]$\mathsf{Coalg}(\mathsf{G})/\gamma$ has objects given by pairs $\langle \langle D, \delta \rangle, d \rangle$ with $\langle D, \delta \rangle$ a G-coalgebra and $d : \langle D, \delta \rangle \to \langle C, \gamma \rangle$ a G-coalgebra homomorphism, and arrows from $\langle \langle D, \delta \rangle, d \rangle$ to $\langle \langle D', \delta' \rangle, d' \rangle$ given by coalgebra homomorphisms $f : \langle D, \delta \rangle \to \langle D', \delta' \rangle$ satisfying $d = d' \circ f$.

[6]$\mathsf{Coalg}(\mathsf{G})\backslash\gamma$ has objects given by pairs $\langle \langle D, \delta \rangle, e \rangle$ with $\langle D, \delta \rangle$ a G-coalgebra and $e : \langle C, \gamma \rangle \to \langle D, \delta \rangle$ a G-coalgebra homomorphism, and arrows from $\langle \langle D, \delta \rangle, e \rangle$ to $\langle \langle D', \delta' \rangle, e' \rangle$ given by coalgebra homomorphisms $f : \langle D, \delta \rangle \to \langle D', \delta' \rangle$ satisfying $e' = f \circ e$.

[7]That is, every C-arrow f has a factorisation of form $f = m \circ e$ with e an epimorphism and m a monomorphism, and moreover, C has the *unique* $(\mathsf{Epi}(\mathsf{C}), \mathsf{Mono}(\mathsf{C}))$-*diagonalisation property*, that is, whenever the C-arrows e, m, f and g, with e an epimorphism and m a monomorphism, satisfy $m \circ f = g \circ e$, there exists a unique C-arrow d satisfying $d \circ e = f$ and $m \circ d = g$.

[8]Hence, G preserves monomorphisms.

The category of (C, G)-coalgebras and (C, G)-coalgebra homomorphisms is denoted $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$, while the functor taking (C, G)-coalgebras to their carrier is denoted $\mathsf{U}_\mathsf{C} : \mathsf{Coalg}(\mathsf{C}, \mathsf{G}) \to \mathsf{C}$.

Remark 1.4. Abstract cosignatures (C, G) induce comonads $(\mathsf{D}, \epsilon, \delta)$ on C, such that $\mathsf{Coalg}(\mathsf{C}, \mathsf{G}) \simeq \mathsf{Coalg}(\mathsf{D})$. Specifically, D is obtained as the limit object of the following ω^{op}-chain:

$$\mathsf{G}_0 = \mathsf{Id}_\mathsf{C} \xLeftarrow{g_0 = \pi_1} \mathsf{G}_1 = \mathsf{Id}_\mathsf{C} \times \mathsf{G}\mathsf{G}_0 \xLeftarrow{g_1 = 1_{\mathsf{Id}_\mathsf{C}} \times \mathsf{G}g_0} \mathsf{G}_2 = \mathsf{Id}_\mathsf{C} \times \mathsf{G}\mathsf{G}_1 \xLeftarrow{g_2 = 1_{\mathsf{Id}_\mathsf{C}} \times \mathsf{G}g_1} \cdots$$

Moreover, since G preserves monomorphisms, so does D. (See *e.g.* [4], 2.3.50 and 2.3.51 for proofs of these statements.) As a result, the functor U_C has a right adjoint.

Remark 1.5. *Destructor cosignatures* over a set V of *visible sorts* are defined in [2] as pairs (H, Δ), with H a set of *hidden sorts* and Δ an $H \times S^+$-sorted set of *operation symbols* (where $S = V \cup H$, and where S^+ denotes the set of finite, non-empty sequences of elements of S). One writes $\delta : h \to s_1 \dots s_n$ for $\delta \in \Delta_{h, s_1 \dots s_n}$. For a destructor cosignature (H, Δ) over V, a Δ_D*-coalgebra* (with D a fixed V-sorted set) is given by an S-sorted set A satisfying $A_v = D_v$ for each $v \in V$, together with, for each $\delta : h \to s_1 \dots s_n$ in Δ, a function $\delta_A : A_h \to A_{s_1} + \dots + A_{s_n}$. Also, a Δ_D*-homomorphism* between Δ_D-coalgebras A and C is given by an S-sorted function $f : A \to C$ additionally satisfying: $f_v = 1_{D_v}$ for each $v \in V$, and $[\iota_1 \circ f_{s_1}, \dots, \iota_n \circ f_{s_n}](\delta_A(a)) = \delta_C(f_h(a))$ for each $\delta : h \to s_1 \dots s_n$ in Δ and each $a \in A_h$, with $h \in H$ and $s_1, \dots, s_n \in S$ (where $\iota_j : C_{s_j} \to C_{s_1} + \dots + C_{s_n}$ for $j = 1, \dots, n$ denote the coproduct injections). Then, the category $\mathsf{Coalg}_D(\Delta)$ of Δ_D-coalgebras and Δ_D-homomorphisms is isomorphic to $\mathsf{Coalg}(\mathsf{Set}^S_D, \mathsf{G}_\Delta)$, with Set^S_D denoting the category of S-sorted sets whose V-sorted components are given by D and S-sorted functions whose V-sorted components are given by 1_D, and with $\mathsf{G}_\Delta : \mathsf{Set}^S_D \to \mathsf{Set}^S_D$ being given by:

$$(\mathsf{G}_\Delta X)_s = \begin{cases} D_s & \text{if } s \in V \\ \prod_{\delta \in \Delta_{s, s_1 \dots s_n}} (X_{s_1} + \dots + X_{s_n}) & \text{if } s \in H \end{cases}, \quad X \in |\mathsf{Set}^S_D|, \ s \in S.$$

Example 1.6. Stacks of natural numbers are specified using a destructor cosignature $\Delta_{\mathtt{ST}}$ over visible sorts `1` and `Nat` (with `1` and `Nat` denoting a one-element set and respectively the set of natural numbers), consisting of a hidden sort `Stack` together with operation symbols `top` : `Stack` → `1 Nat` and `rest` : `Stack` → `1 Stack`. The abstract cosignature associated to this destructor cosignature is $(\mathsf{Set}^{S_{\mathtt{ST}}}_D, \mathsf{G}_{\mathtt{ST}})$, with $S_{\mathtt{ST}} = \{\mathtt{1}, \mathtt{Nat}, \mathtt{Stack}\}$, with D denoting the $\{\mathtt{1}, \mathtt{Nat}\}$-sorted set whose components are given by $\{*\}$ and $\mathbb{N}$, and with the hidden-sorted component of $\mathsf{G}_{\mathtt{ST}}$ being given by:

$$(\mathsf{G}_{\mathtt{ST}} X)_{\mathtt{Stack}} = (X_{\mathtt{1}} + X_{\mathtt{Nat}}) \times (X_{\mathtt{1}} + X_{\mathtt{Stack}}), \quad X \in \mathsf{Set}^{S_{\mathtt{ST}}}_D.$$

The existence of a final object in C (see Def. 1.2) and of a right adjoint to U_C (see Rem. 1.4) result in the existence of final (C, G)-coalgebras.

Proposition 1.7. *Let* (C, G) *denote an abstract cosignature. Then,* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ *has a final object.*

As a result, a notion of *observability* of coalgebras can be defined.

Definition 1.8. Let (C, G) denote an abstract cosignature. A (C, G)-coalgebra is called **observable** if and only if the C-arrow underlying its unique homomorphism into the final (C, G)-coalgebra is a monomorphism.

Also, for an abstract cosignature (C, G), the existence of pullbacks in C together with the preservation of pullbacks by G result in the existence of pullbacks in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$.

Proposition 1.9. *Let* (C, G) *denote an abstract cosignature. Then,* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ *has pullbacks and* U_C *preserves them.*

Propositions 1.7 and 1.9 now yield the following:

Corollary 1.10. *For an abstract cosignature* (C, G)*,* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ *has finite limits.*

In particular, $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ has equalisers.

Remark 1.11. Since U_C preserves pullbacks, it follows that pullbacks in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ define (C, G)-bisimulations. Also, since $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ has a final object, largest bisimulations exist and are given by (the C-arrows underlying) the pullbacks of the unique (C, G)-coalgebra homomorphisms into the final (C, G)-coalgebra. And finally, since G preserves pullbacks, largest bisimulations carry *unique* coalgebra structures which make the C-arrows underlying the above-mentioned pullbacks into (C, G)-coalgebra homomorphisms.

The creation (and hence preservation) of kernel and respectively cokernel pairs by the functor U_C yields a characterisation of monomorphisms and respectively epimorphisms in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$.

Proposition 1.12. *Let* (C, G) *denote an abstract cosignature. Then,* U_C *preserves and reflects monomorphisms as well as epimorphisms.*

Proof (sketch). The fact that an arrow in a category is a monomorphism (respectively epimorphism) if and only if the identity relation (corelation) defines a kernel (cokernel) pair for it is used. □

It is worth noting that the preservation of pullbacks by G is crucial to the proof of preservation of monomorphisms by U_C.

A consequence of $(\mathsf{Epi}(\mathsf{C}), \mathsf{Mono}(\mathsf{C}))$ defining a factorisation system for C and of the preservation of monomorphisms by G is the existence in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ of factorisations of form $f = m \circ e$, with e defining a homomorphic image of the domain of f and with m defining a subcoalgebra of the codomain of f.

Proposition 1.13. *Let* (C, G) *denote an abstract cosignature. Also, let* $f : \langle C, \gamma \rangle \to \langle D, \delta \rangle$ *denote a* (C, G)*-coalgebra homomorphism, and let* $\mathsf{U_C} f = m \circ e$*, with* $e : C \to E$ *and* $m : E \to D$*, denote the epi-mono factorisation of* $\mathsf{U_C} f$[9]*. Then, there exists a unique* (C, G)*-coalgebra structure* $\langle E, \eta \rangle$ *on* E *which makes* e *and* m *into* (C, G)*-coalgebra homomorphisms.*

Proof (sketch). The unique $(\mathsf{Epi}(\mathsf{C}), \mathsf{Mono}(\mathsf{C}))$-diagonalisation property of C together with the preservation of monomorphisms by G yield a unique C-arrow $\eta : E \to \mathsf{G}E$ satisfying $\eta \circ e = \mathsf{G}e \circ \gamma$ and $\mathsf{G}m \circ \eta = \delta \circ m$. □

Corollary 1.14. *Let* (C, G) *denote an abstract cosignature. Then,* $(\mathcal{E}, \mathcal{M}) = (\mathsf{U_C}^{-1}(\mathsf{Epi}(\mathsf{C})), \mathsf{U_C}^{-1}(\mathsf{Mono}(\mathsf{C})))$ *is a factorisation system for* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$*.*

Proof (sketch). By Proposition 1.13, $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ has $(\mathcal{E}, \mathcal{M})$-factorisations. Also, the fact that G preserves monomorphisms can be used to show that the unique diagonalisation property of C lifts to $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$. □

Abstract cosignatures (C, G) induce natural transformations $\lambda : \mathsf{U_C} \Rightarrow \mathsf{G}\mathsf{U_C}$, with $\lambda_{\langle C, \gamma \rangle} : C \to \mathsf{G}C$ being given by $\lambda_{\langle C, \gamma \rangle} = \gamma$ for $\langle C, \gamma \rangle \in |\mathsf{Coalg}(\mathsf{C}, \mathsf{G})|$. These natural transformations define basic observations on the carriers of (C, G)-coalgebras: $\lambda_{\langle C, \gamma \rangle}$ uses the coalgebraic structure given by γ to extract information of type G from C. More complex observations on the carriers of (C, G)-coalgebras can also be defined, *e.g.* by considering natural transformations of form $\mathsf{G}^{n-1}\lambda \circ \ldots \circ \mathsf{G}\lambda \circ \lambda : \mathsf{U_C} \Rightarrow \mathsf{G}^n\mathsf{U_C}$ with $n \in \mathbb{N}^*$. The next definition formally captures such complex observations, as well as more general ones, by exploiting the fact that the result of an observation depends solely on the coalgebraic structure.

Definition 1.15. Let (C, G) denote an abstract cosignature. A (C, G)**-observer** is a pair (K, c), with $\mathsf{K} : \mathsf{C} \to \mathsf{C}$ an endofunctor (called the **type** of the observer) which preserves monomorphisms, and with $c : \mathsf{U_C} \Rightarrow \mathsf{K}\mathsf{U_C}$ a natural transformation.

(C, G)-observers are parameterised by (C, G)-coalgebras: a (C, G)-observer (K, c) specifies, for each (C, G)-coalgebra $\langle C, \gamma \rangle$, a C-arrow $c_\gamma : C \to \mathsf{K}C$, extracting information of type K from C according to γ. Moreover, the extraction of K-information from coalgebras commutes with coalgebra homomorphisms: if $f : \langle C, \gamma \rangle \to \langle D, \delta \rangle$ is a (C, G)-coalgebra homomorphism, then $c_\delta \circ f = \mathsf{K}f \circ c_\gamma$.

(C, G)-observers can be composed. Specifically, if (K, c) and (K', c') are (C, G)-observers, then so is $(\mathsf{K}\mathsf{K}', \mathsf{K}c' \circ c)$. (The preservation of monomorphisms by $\mathsf{K}\mathsf{K}'$ follows from their preservation by each of K, K'.) In particular, if $n \in \mathbb{N}^*$ and if $\lambda : \mathsf{U_C} \Rightarrow \mathsf{G}\mathsf{U_C}$ is as before, then $\mathsf{G}^{n-1}\lambda \circ \ldots \circ \mathsf{G}\lambda \circ \lambda : \mathsf{U_C} \Rightarrow \mathsf{G}^n\mathsf{U_C}$ is a (C, G)-observer.

Pairs of observers are here used to constrain system implementations, by requiring that the given observers yield similar results.

[9]This factorisation is unique up to isomorphism.

Definition 1.16. Let (C, G) denote an abstract cosignature. A (C, G)**-coequation** is a tuple (K, l, r), with (K, l) and (K, r) denoting (C, G)-observers. A (C, G)-coalgebra $\langle C, \gamma \rangle$ **satisfies** a (C, G)-coequation (K, l, r) (written $\langle C, \gamma \rangle \models (\mathsf{K}, l, r)$) if and only if $l_\gamma = r_\gamma$.

For a set E of (C, G)-coequations, the full subcategory of $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ whose objects satisfy the coequations in E is denoted $\mathsf{Coalg}(\mathsf{C}, \mathsf{G}, E)$.

Remark 1.17. For a destructor cosignature (H, Δ) and an S-sorted set $\mathcal{C}$ of *covariables*, the S-sorted set $T_\Delta[\mathcal{C}]$ of Δ-*coterms with covariables from* $\mathcal{C}$ is defined in [2] as the least S-sorted set satisfying:

1. $Z \in T_\Delta[\mathcal{C}]_s$ for $Z \in \mathcal{C}_s$, with $s \in S$;
2. $[t_1, \ldots, t_n]\delta \in T_\Delta[\mathcal{C}]_s$ for $\delta \in \Delta_{s,s_1 \ldots s_n}$ and $t_i \in T_\Delta[\mathcal{C}]_{s_i}$, $i = 1, \ldots, n$, with $s \in H$ and $s_1, \ldots, s_n \in S$.

A notion of substitution of coterms for covariables, similar to that of substitution of terms for variables in many-sorted algebra, can also be defined (see [2]).

Given a Δ_D-coalgebra A, a set $\{Z_1, \ldots, Z_n\}$ of covariables, with $Z_i : s_i$ for $i = 1, \ldots, n$, and a covariable $Z \in \{Z_1, \ldots, Z_n\}$, with $Z : s$, one writes $\iota_Z : A_s \to A_{s_1} + \ldots + A_{s_n}$ for the corresponding coproduct injection. Then, the interpretation provided by coalgebras to the operation symbols in destructor cosignatures extends naturally to an interpretation of coterms over these operation symbols. Specifically, the interpretation of a Δ-coterm $t \in T_\Delta[\mathcal{C}]$ in a Δ_D-coalgebra A, denoted t_A, is defined inductively as follows:

1. $Z_A = \iota_Z$ for $Z \in \mathcal{C}_s$, with $s \in S$;
2. $([t_1, \ldots, t_n]\delta)_A = [(t_1)_A, \ldots, (t_n)_A] \circ \delta_A$ for $\delta \in \Delta_{s,s_1 \ldots s_n}$ and $t_i \in T_\Delta[\mathcal{C}]_{s_i}$, $i = 1, \ldots, n$, with $s \in H$ and $s_1, \ldots, s_n \in S$.

Destructor cosignatures admit final coalgebras. Moreover, the hidden-sorted carriers of these coalgebras have elements given by mappings which *consistently* assign, to each coterm of the given sort, a covariable appearing in that coterm (specifying an evaluation path for the coterm), together with a value of the same sort as the covariable (specifying a result for the evaluation of the coterm), provided that this sort is a visible one. That is, the elements of final coalgebras are abstract behaviours, assigning visible results to all possible observations than can be made on the system states using the destructors provided.

Δ-*coequations* are used in [2] to constrain the interpretations of Δ-coterms by Δ_D-coalgebras. A Δ-*coequation* is given by a tuple (l, r, C) (alternatively denoted $l = r$ if C), with $l, r \in T_\Delta[\mathcal{C}]_h$ and C consisting of *conditions* $(t_1, \mathcal{C}'_1), \ldots, (t_n, \mathcal{C}'_n)$, with $t_i \in T_\Delta[\mathcal{C}_i]_h$ and $\mathcal{C}'_i \subseteq \mathcal{C}_i$ for $i = 1, \ldots, n$, for some $h \in H$. A Δ_D-coalgebra A *satisfies* a Δ-coequation e of the above form if and only if $l_A(a) = r_A(a)$ holds whenever $a \in A_h$ is such that $(t_i)_A(a) \in \iota_{Z_i}(A_{s_i})$ for some $Z_i \in (\mathcal{C}'_i)_{s_i}$, for $i = 1, \ldots, n$ (case in which a is said to satisfy the conditions C).

Δ-coequations are an instance of the abstract notion of coequation. For, given a Δ-coterm $t \in T_\Delta[\mathcal{C}]_s$ with $\mathcal{C}$ a finite set of covariables, together with some conditions C for the sort h, one can define a $(\mathsf{Set}^S_D, \mathsf{G}_\Delta)$-observer $(\mathsf{K}, \bar{t})$, with

$\mathsf{K} : \mathsf{Set}^S_D \to \mathsf{Set}^S_D$ being given by:

$$(\mathsf{K}X)_s = \begin{cases} D_s & \text{if } s \in V \\ 1 + \coprod\limits_{s' \in S} \coprod\limits_{Z \in \mathcal{C}_{s'}} X_{s'} & \text{if } s = h \\ 1 & \text{if } s \in H \setminus \{h\} \end{cases}, \quad X \in |\mathsf{Set}^S_D|,\ s \in S$$

and with $\bar{t} : \mathsf{U} \Rightarrow \mathsf{KU}$ (where $\mathsf{U} : \mathsf{Coalg}(\mathsf{Set}^S_D, \mathsf{G}_\Delta) \to \mathsf{Set}^S_D$ denotes the functor taking $(\mathsf{Set}^S_D, \mathsf{G}_\Delta)$-coalgebras to their carrier) being given by:

$$(\bar{t}_{\langle A,\alpha\rangle})_s(a) = \begin{cases} a & \text{if } s \in V \\ \iota_2(t_A(a)) & \text{if } s = h \text{ and } C \text{ holds in } a \\ \iota_1(*) & \text{if } s = h \text{ and } C \text{ does not hold in } a \\ * & \text{if } s \in H \setminus \{h\} \end{cases}, \quad a \in A_s,\ s \in S$$

for each $(\mathsf{Set}^S, \mathsf{G}_\Delta)$-coalgebra $\langle A, \alpha\rangle$ with A its corresponding Δ_D-coalgebra. (Note that K only depends on $\mathcal{C}$.) Then, if e denotes a Δ-coequation of form $l = r$ if C, and if $(\mathsf{K}, \bar{l})$ and $(\mathsf{K}, \bar{r})$ denote the $(\mathsf{Set}^S, \mathsf{G}_\Delta)$-observers induced by the Δ-coterms l and r together with the conditions C, then $A \models_\Delta e$ is equivalent to $\langle A, \alpha\rangle \models (\mathsf{K}, \bar{l}, \bar{r})$.

Example 1.18. Given the destructor cosignature of stacks in Example 1.6, a state invariant for stacks is captured by the coequations:

```
[Z,S]rest = [Z,S']rest if ([Z,N]top,Z)
[Z,N]top = [Z,N']top if ([Z,S]rest,Z)
```

formalising the fact that a stack is either empty, in which case it has neither a top nor a rest, or non-empty, in which case it has both a top and a rest.

The notion of satisfaction of Δ-coequations by Δ_D-coalgebras has the property of being preserved by homomorphisms whose underlying functions are surjective, and reflected by homomorphisms whose underlying functions are injective. The next result generalises this property to the abstract notion of satisfaction of coequations.

Proposition 1.19. *Let* (C, G) *denote an abstract cosignature, let* $f : \langle C, \gamma\rangle \to \langle D, \delta\rangle$ *denote a* (C, G)*-coalgebra homomorphism, and let* (K, l, r) *denote a* (C, G)*-coequation. Then, the following hold:*

1. *if* $\mathsf{U_C} f$ *is epi, then* $\langle C, \gamma\rangle \models (\mathsf{K}, l, r)$ *implies* $\langle D, \delta\rangle \models (\mathsf{K}, l, r)$*;*
2. *if* $\mathsf{U_C} f$ *is mono, then* $\langle D, \delta\rangle \models (\mathsf{K}, l, r)$ *implies* $\langle C, \gamma\rangle \models (\mathsf{K}, l, r)$*.*

Proof. 1 follows from: $\langle C, \gamma\rangle \models (\mathsf{K}, l, r) \Leftrightarrow l_\gamma = r_\gamma \Rightarrow \mathsf{KU_C} f \circ l_\gamma = \mathsf{KU_C} f \circ r_\gamma \Leftrightarrow l_\delta \circ \mathsf{U_C} f = r_\delta \circ \mathsf{U_C} f \Leftrightarrow l_\delta = r_\delta \Leftrightarrow \langle D, \delta\rangle \models (\mathsf{K}, l, r)$, while 2 follows from: $\langle D, \delta\rangle \models (\mathsf{K}, l, r) \Leftrightarrow l_\delta = r_\delta \Rightarrow l_\delta \circ \mathsf{U_C} f = r_\delta \circ \mathsf{U_C} f \Leftrightarrow \mathsf{KU_C} f \circ l_\gamma = \mathsf{KU_C} f \circ r_\gamma \Leftrightarrow l_\gamma = r_\gamma \Leftrightarrow \langle C, \gamma\rangle \models (\mathsf{K}, l, r)$. (The fact that K preserves monomorphisms is used in proving 2.) $\square$

The requirement that $\mathsf{U_C} f$ is epi (respectively mono) amounts to f defining a homomorphic image (respectively a subcoalgebra).

Another property of the abstract notion of satisfaction of coequations is that it is preserved by colimits in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$[10].

Proposition 1.20. *Let* (C, G) *denote an abstract cosignature. Also, let* $\mathsf{d} : \mathsf{D} \to \mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ *denote a diagram of shape* D *in* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$*, and let* $(\langle C, \gamma\rangle, (f_D : \mathsf{d}D \to \langle C, \gamma\rangle)_{D \in |\mathsf{D}|})$ *denote its colimit. Finally, let* (K, l, r) *denote a* (C, G)*-coequation. Then,* $\mathsf{d}D \models (\mathsf{K}, l, r)$ *for any* $D \in |\mathsf{D}|$ *implies* $\langle C, \gamma\rangle \models (\mathsf{K}, l, r)$*.*

Proof (sketch). The conclusion follows from: $l_\gamma \circ \mathsf{U_C} f_D = \mathsf{K}\mathsf{U_C} f_D \circ l_{\mathsf{d}D} = \mathsf{K}\mathsf{U_C} f_D \circ r_{\mathsf{d}D} = r_\gamma \circ \mathsf{U_C} f_D$ for $D \in |\mathsf{D}|$, together with the preservation of colimits by $\mathsf{U_C}$. □

Corollary 1.21. *Let* (C, G) *denote an abstract cosignature, and let* E *denote a set of* (C, G)*-coequations. Then,* $|\mathsf{Coalg}(\mathsf{C}, \mathsf{G}, E)|$ *is a covariety.*

Proof. The conclusion follows by Propositions 1.19 and 1.20. □

For a (C, G)-coalgebra $\langle C, \gamma\rangle$ and a (C, G)-coequation e (respectively a set of (C, G)-coequations E), the full subcategory of $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma$ whose objects satisfy the coequation e (the coequations in E) is denoted $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^e$ (respectively $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^E$). The observation that abstract cosignatures induce comonads (see Rem. 1.4) now results in the existence of final objects in $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^e$ and $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^E$.

Proposition 1.22. *Let* (C, G) *denote an abstract cosignature, let* $\langle C, \gamma\rangle$ *denote a* (C, G)*-coalgebra, and let* e *denote a* (C, G)*-coequation. Then,* $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^e$ *has a final object, which at the same time defines a* (C, G)*-subcoalgebra of* $\langle C, \gamma\rangle$*.*

Proof (sketch). Say e is of form (K, l, r). Let $(\mathsf{D}, \epsilon, \delta)$ denote the comonad induced by G, let $l^\flat_\gamma, r^\flat_\gamma : \langle C, \gamma\rangle \to \langle \mathsf{DK}C, \gamma'\rangle$ denote the unique (co)extensions of the C-arrows $l_\gamma, r_\gamma : C \to \mathsf{K}C$ to (C, G)-coalgebra homomorphisms, and let $\iota : \langle S, \xi\rangle \to \langle C, \gamma\rangle$ denote an equaliser for $l^\flat_\gamma, r^\flat_\gamma$ (see Cor. 1.10). Then, $\mathsf{U_C}\iota$ is a monomorphism[11], and hence $\langle\langle S, \xi\rangle, \iota\rangle$ defines a (C, G)-subcoalgebra of $\langle C, \gamma\rangle$. Moreover, $\langle\langle S, \xi\rangle, \iota\rangle$ is final in $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^e$ – the fact that $\langle S, \xi\rangle \models e$ follows from standard properties of adjunctions together with the observation that $\mathsf{K}\mathsf{U_C}\iota$ is a monomorphism, while finality of $\langle\langle S, \xi\rangle, \iota\rangle$ in $(\mathsf{Coalg}(\mathsf{C}, \mathsf{G})/\gamma)^e$ follows from standard properties of adjunctions together with the couniversality of ι. □

The existence of limits of ω^{op}-chains in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ (following from the existence of such limits in C together with their preservation by G) now yields the following.

[10] The existence of colimits in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ follows from the existence of colimits in C together with the creation of colimits by $\mathsf{U_C}$.

[11] This follows directly from the preservation of monomorphisms by $\mathsf{U_C}$. However, an alternative proof of this statement which does not make use of the preservation of pullbacks by G (and therefore carries over to the algebraic case, where preservation of pushouts by the endofunctors defining abstract signatures is not required) can also be given. The alternative proof uses the creation of factorisations and the reflection of epimorphisms by $\mathsf{U_C}$.

Proposition 1.23. *Let* (C,G) *denote an abstract cosignature, let* $\langle C,\gamma\rangle$ *denote a* (C,G)*-coalgebra, and let* E *denote an enumerable set of* (C,G)*-coequations. Then,* $(\mathsf{Coalg}(\mathsf{C},\mathsf{G})/\gamma)^E$ *has a final object, which at the same time defines a* (C,G)*-subcoalgebra of* $\langle C,\gamma\rangle$*.*

Proof. Say $E=\{e_i \mid i\in\mathbb{N}\}$. Now let $\langle C_0,\gamma_0\rangle=\langle C,\gamma\rangle$. Also, for $i\in\mathbb{N}$, let $\langle\langle C_{i+1},\gamma_{i+1}\rangle,c_{i+1}\rangle$ denote a final object in $(\mathsf{Coalg}(\mathsf{C},\mathsf{G})/\gamma_i)^{e_i}$. Finally, let $(\langle S,\xi\rangle,(\iota_i)_{i\in\mathbb{N}})$ define a limit for the ω^{op}-chain defined by $c_1,c_2,\ldots$

$$\begin{tikzcd}
& \langle S,\xi\rangle \arrow[dl, "\iota_0"'] \arrow[d, "\iota_1"] & \cdots \\
\langle C_0,\gamma_0\rangle & \langle C_1,\gamma_1\rangle \arrow[l, "c_1"'] & \cdots \arrow[l, "c_2"']
\end{tikzcd}$$

The fact that $\mathsf{U_C}$ creates limits of ω^{op}-chains together with the fact that each of $\mathsf{U_C}c_1,\mathsf{U_C}c_2,\ldots$ are monomorphisms result in each of $\mathsf{U_C}\iota_0,\mathsf{U_C}\iota_1,\ldots$ being monomorphisms. In particular, $\mathsf{U_C}\iota_0$ is a monomorphism, and hence $\langle\langle S,\xi\rangle,\iota_0\rangle$ defines a (C,G)-subcoalgebra of $\langle C,\gamma\rangle$. Moreover, $\langle S,\xi\rangle\models E$ – this follows by 2 of Proposition 1.19 from $\langle S,\xi\rangle$ being a subcoalgebra of each of $\langle C_1,\gamma_1\rangle,\langle C_2,\gamma_2\rangle,\ldots$ together with $\langle C_i,\gamma_i\rangle\models e_i$ for $i\in\mathbb{N}$. Also, the fact that $\langle\langle C_{i+1},\gamma_{i+1}\rangle,c_{i+1}\rangle$ is final in $(\mathsf{Coalg}(\mathsf{C},\mathsf{G})/\gamma_i)^{e_i}$ for $i\in\mathbb{N}$ results in $\langle\langle S,\xi\rangle,\iota_0\rangle$ being final in $(\mathsf{Coalg}(\mathsf{C},\mathsf{G})/\gamma)^E$. □

Corollary 1.24. *Let* (C,G) *denote an abstract cosignature and let* E *denote an enumerable set of* (C,G)*-coequations. Then,* $\mathsf{Coalg}(\mathsf{C},\mathsf{G},E)$ *has a final object.*

Proof (sketch). Proposition 1.7 and Proposition 1.23 are used. □

Example 1.25. The hidden-sorted carrier of the final coalgebra of the stack specification (see Ex. 1.18) is isomorphic to the set of stacks (either finite or infinite) of natural numbers. It is also worth noting that, in the absence of the two coequations, the final coalgebra also contains unwanted behaviours, allowing, for instance, for "stacks" which have no top element, but have a second element.

A notion of satisfaction of coequations up to bisimulation which unifies the various notions of satisfaction up to observability employed by existing equational specification formalisms, including [9, 13], can also be defined.

Definition 1.26. Let (C,G) denote an abstract cosignature. A (C,G)-coalgebra $\langle C,\gamma\rangle$ **satisfies** a (C,G)-coequation (K,l,r) **up to bisimulation** (written $\langle C,\gamma\rangle\models^{\mathsf{b}}(\mathsf{K},l,r)$) if and only if $\mathsf{KU_C}!\circ l_\gamma=\mathsf{KU_C}!\circ r_\gamma$, with $!:\langle C,\gamma\rangle\to\langle F,\zeta\rangle$ denoting the unique (C,G)-coalgebra homomorphism from $\langle C,\gamma\rangle$ to the final (C,G)-coalgebra.

Remark 1.27. If K preserves kernel pairs, then $\langle C,\gamma\rangle\models^{\mathsf{b}}(\mathsf{K},l,r)$ holds precisely when $\langle l_\gamma,r_\gamma\rangle$ factors through $\langle \mathsf{K}r_1,\mathsf{K}r_2\rangle$:

$$\begin{tikzcd}
C \arrow[r, "\langle l_\gamma,r_\gamma\rangle"] \arrow[dr, dashed, "c"'] & \mathsf{K}C\times\mathsf{K}C \\
& \mathsf{K}R \arrow[u, "\langle\mathsf{K}r_1,\mathsf{K}r_2\rangle"']
\end{tikzcd}$$

with $\langle R, r_1, r_2\rangle$ denoting (C,G)-bisimilarity on $\langle C,\gamma\rangle$. For, in this case, $\mathsf{K}r_1, \mathsf{K}r_2$ define a kernel pair for $\mathsf{KU_C}!$ (see also Rem. 1.11).

Standard satisfaction of (K, l, r) by $\langle C,\gamma\rangle$ implies its satisfaction up to bisimulation by $\langle C,\gamma\rangle$. And if, in addition, $\langle C,\gamma\rangle$ is observable, the converse also holds. For, in this case, $\mathsf{U_C}!$ is a monomorphism, while K preserves monomorphisms.

Satisfaction up to bisimulation can be expressed in terms of standard satisfaction. Specifically, if $! = m \circ e$ with $e : \langle C,\gamma\rangle \to \langle E,\eta\rangle$ and $m : \langle E,\eta\rangle \to \langle F,\zeta\rangle$ denotes the $(\mathsf{U_C}^{-1}(\mathsf{Epi}(\mathsf{C})), \mathsf{U_C}^{-1}(\mathsf{Mono}(\mathsf{C})))$-factorisation of the unique (C,G)-coalgebra homomorphism from $\langle C,\gamma\rangle$ to the final (C,G)-coalgebra, then $\langle C,\gamma\rangle \models^{\mathsf{b}} (\mathsf{K}, l, r)$ holds precisely when $\langle E,\eta\rangle \models (\mathsf{K}, l, r)$ does. This follows from $\mathsf{U_C}e$ being an epimorphism together with $\mathsf{KU_C}m$ being a monomorphism.

The notion of satisfaction of coequations up to bisimulation enjoys properties similar to those of standard satisfaction. In particular, Propositions 1.19, 1.20, 1.22 and 1.23, as well as Corollaries 1.21 and 1.24 hold; moreover, no restriction on the homomorphism f is required by 2 of Proposition 1.19.

Provided that the functors K used to define the types of coequations preserve kernel pairs, proofs of satisfaction of coequations up to bisimulation can benefit from the use of *generic bisimulations*, as defined below.

Definition 1.28. Let (C,G) denote an abstract cosignature, and let $\mathcal{C}$ denote a full subcategory of $\mathsf{Coalg}(\mathsf{C},\mathsf{G})$[12]. A **generic (C,G)-bisimulation on $\mathcal{C}$** is given by a tuple $\langle \mathsf{R}, \pi_1, \pi_2\rangle$ with $\mathsf{R} : \mathcal{C} \to \mathsf{C}$ and $\pi_1, \pi_2 : \mathsf{R} \Rightarrow \mathsf{U_C}\restriction_{\mathcal{C}}$, such that $\langle \mathsf{R}\gamma, \pi_{1,\gamma}, \pi_{2,\gamma}\rangle$ defines a (C,G)-bisimulation on $\langle C,\gamma\rangle$ for any $\langle C,\gamma\rangle \in |\mathcal{C}|$.

That is, a generic bisimulation on $\mathcal{C}$ associates to each coalgebra in $\mathcal{C}$ a bisimulation relation on it, with this association being functorial.

Then, proving that a (C,G)-coequation (K, l, r) with K preserving kernel pairs holds, up to bisimulation, in a full subcategory $\mathcal{C}$ of $\mathsf{Coalg}(\mathsf{C},\mathsf{G})$ can be reduced to exhibiting a generic (C,G)-bisimulation $\langle \mathsf{R}, \pi_1, \pi_2\rangle$ on $\mathcal{C}$, such that $\langle l_\gamma, r_\gamma\rangle$ factors through $\langle \mathsf{K}\pi_{1,\gamma}, \mathsf{K}\pi_{2,\gamma}\rangle$ for any $\langle C,\gamma\rangle \in |\mathcal{C}|$ (see also Rem. 1.27):

$$\begin{tikzcd} C \arrow[r, "\langle l_\gamma, r_\gamma\rangle"] \arrow[dr, dashed, "c"'] & \mathsf{K}C \times \mathsf{K}C \\ & \mathsf{KR}\gamma \arrow[u, "\langle \mathsf{K}\pi_{1,\gamma}, \mathsf{K}\pi_{2,\gamma}\rangle"'] \end{tikzcd}$$

Translations between abstract cosignatures, specifying a change in the type of information that can be observed about a system, are captured by *abstract cosignature morphisms*.

Definition 1.29. An **(abstract) cosignature morphism** between abstract cosignatures (C,G) and $(\mathsf{C}',\mathsf{G}')$ is a pair (U,η), with $\mathsf{U} : \mathsf{C}' \to \mathsf{C}$ a functor which

[12] $\mathcal{C}$ could, for instance, consist of all (C,G)-coalgebras satisfying (possibly only up to bisimulation) a given set of (C,G)-coequations.

preserves limits and has a right adjoint R[13], and with $\eta : \mathsf{UG}' \Rightarrow \mathsf{GU}$ a natural transformation.

An abstract cosignature morphism $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ induces a *reduct functor*[14] $\mathsf{U}_\eta : \mathsf{Coalg}(\mathsf{C}', \mathsf{G}') \to \mathsf{Coalg}(\mathsf{C}, \mathsf{G})$, with U_η taking a $(\mathsf{C}', \mathsf{G}')$-coalgebra $\langle D, \delta \rangle$ to the (C, G)-coalgebra $\langle \mathsf{U}D, \eta_D \circ \mathsf{U}\delta \rangle$, as well as a translation, itself denoted η, of (C, G)-observers to $(\mathsf{C}', \mathsf{G}')$-observers, with η taking a (C, G)-observer (K, c) to the $(\mathsf{C}', \mathsf{G}')$-observer $(\mathsf{RKU}, c^\flat_{\mathsf{U}_\eta})$:

$$\begin{tikzcd} \mathsf{U_C U_\eta} = \mathsf{UU_{C'}} \arrow[d, Rightarrow, "c_{\mathsf{U}_\eta}"'] \\ \mathsf{KU_C U_\eta} = \mathsf{KUU_{C'}} \end{tikzcd} \qquad \begin{tikzcd} \mathsf{U_{C'}} \arrow[d, Rightarrow, "c^\flat_{\mathsf{U}_\eta}"] \\ \mathsf{RKUU_{C'}} \end{tikzcd}$$

where:

$$\begin{tikzcd} \mathsf{Coalg}(\mathsf{C}, \mathsf{G}) \arrow[d, "\mathsf{U_C}"'] & \mathsf{Coalg}(\mathsf{C}', \mathsf{G}') \arrow[l, "\mathsf{U}_\eta"'] \arrow[d, "\mathsf{U_{C'}}"] \\ \mathsf{C} \arrow[r, dashed, shift right, "\mathsf{R}"'] & \mathsf{C}' \arrow[l, shift right, "\mathsf{U}"'] \end{tikzcd}$$

(The preservation of monomorphisms by RKU follows from the preservation of monomorphisms by each of U (as U preserves pullbacks), K and R. And if, in addition, K preserves kernel pairs, so does RKU (as both U and R do.)

The translation of (C, G)-observers into $(\mathsf{C}', \mathsf{G}')$-observers extends to a translation of (C, G)-coequations into $(\mathsf{C}', \mathsf{G}')$-coequations. As one would expect, this translation has the property that a (C, G)-coequation holds in the (C, G)-reduct of a $(\mathsf{C}', \mathsf{G}')$-coalgebra if and only if its translation along (U, η) holds in the given $(\mathsf{C}', \mathsf{G}')$-coalgebra. That is, the *satisfaction condition* of institutions (see [8]) holds.

Proposition 1.30. *Let $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ denote an abstract cosignature morphism, let $\langle D, \delta \rangle$ denote a $(\mathsf{C}', \mathsf{G}')$-coalgebra, and let e denote a (C, G)-coequation. Then, $\mathsf{U}_\eta \langle D, \delta \rangle \models e$ if and only if $\langle D, \delta \rangle \models \eta(e)$.*

Proof. If e is of form (K, l, r), then $\mathsf{U}_\eta \langle D, \delta \rangle \models e$ translates to $l_{\mathsf{U}_\eta \delta} = r_{\mathsf{U}_\eta \delta}$, while $\langle D, \delta \rangle \models \eta(e)$ translates to $l^\flat_{\mathsf{U}_\eta \delta} = r^\flat_{\mathsf{U}_\eta \delta}$. The conclusion then follows by R being a right adjoint to U. □

Remark 1.31. A *destructor cosignature morphism* [2] between destructor cosignatures (H, Δ) and (H', Δ') over V is given by a function $\phi : S \to S'$ satisfying $\phi \restriction_V = 1_V$ and $\phi(H) \subseteq H'$, together with an $H \times S^+$-sorted function $(\phi_{h,w})_{h \in H, w \in S^+}$, with $\phi_{h,w} : \Delta_{h,w} \to \Delta'_{\phi(h), \phi^+(w)}$ for $h \in H$ and $w \in S^+$ (where

[13] Consequently, U also preserves colimits.

[14] The terminology is borrowed from the theory of *institutions*, see *e.g.* [8].

ϕ^+ denotes the pointwise extension of $\phi : S \to S'$ to a function from S^+ to S'^+). Destructor cosignature morphisms $\phi : (H, \Delta) \to (H', \Delta')$ induce reduct functors $\mathsf{U}_\phi : \mathsf{Coalg}_D(\Delta') \to \mathsf{Coalg}_D(\Delta)$ (with U_ϕ taking Δ'_D-coalgebras C' to the Δ_D-coalgebras having carrier $C'\restriction_\Delta = (C'_{\phi(s)})_{s \in S}$ and operations given by $\delta_{C'\restriction_\Delta} = \phi(\delta)_{C'}$ for $\delta \in \Delta$), as well as translations of Δ-coequations to Δ'-coequations. Then, $\mathsf{U}_\phi : \mathsf{Coalg}_D(\Delta') \to \mathsf{Coalg}_D(\Delta)$ is naturally isomorphic to U_{η_ϕ}, with the hidden-sorted components of the natural transformation $\eta_\phi : \mathsf{UG}_{\Delta'} \Rightarrow \mathsf{G}_\Delta \mathsf{U}$ being given by:

$$(\eta_{\phi,X})_h((x_{\delta'})_{\delta' \in \Delta'_{\phi(h)}}) = (x_{\phi(\delta)})_{\delta \in \Delta_h},\ h \in H,\ X \in |\mathsf{Set}_D^{S'}|$$

and with $\mathsf{U} : \mathsf{Set}_D^{S'} \to \mathsf{Set}_D^S$ denoting the functor taking $A \in |\mathsf{Set}_D^{S'}|$ to $(A_{\phi(s)})_{s \in S} \in |\mathsf{Set}_D^S|$. Moreover, the translation along ϕ of Δ-coequations agrees with the translation along η_ϕ of the induced G_Δ-coequations.

Remark 1.32. If $(\mathsf{D}, \epsilon, \delta)$ and $(\mathsf{D}', \epsilon', \delta')$ denote the comonads induced by (C, G) and $(\mathsf{C}', \mathsf{G}')$ (see Rem. 1.4), then abstract cosignature morphisms $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ induce comonad morphisms[15] $(\mathsf{U}, \rho) : (\mathsf{D}, \epsilon, \delta) \to (\mathsf{D}', \epsilon', \delta')$, with the natural transformation $\rho : \mathsf{UD}' \Rightarrow \mathsf{DU}$ arising from the observation that DU is a limit object for the following ω^{op}-chain:

$$\mathsf{G}_0\mathsf{U} \xLeftarrow{(g_0)_\mathsf{U}} \mathsf{G}_1\mathsf{U} \xLeftarrow{(g_1)_\mathsf{U}} \mathsf{G}_2\mathsf{U} \xLeftarrow{(g_2)_\mathsf{U}} \cdots$$

whereas UD' is the object of a cone for this ω^{op}-chain. Moreover, if η is a natural monomorphism, then so is ρ. (The fact that U preserves limits is used to show this. See [4] (2.3.52 and 2.3.53) for proofs of these statements.)

The comonad morphism induced by an abstract cosignature morphism provides some information about the relationship between the notions of observability associated to the source and target of the given cosignature morphism. For, if $1'$ denotes a final C'-object, then the $1'$-component of the natural transformation ρ defining the comonad morphism gives the unique homomorphism from the (C, G)-reduct of the final $(\mathsf{C}', \mathsf{G}')$-coalgebra to the final (C, G)-coalgebra. Hence, the fact that the C-arrow $\rho_{1'} : \mathsf{UD}'1' \to \mathsf{DU}1'$ is a monomorphism reflects the fact that the notion of observability induced by target cosignature does not refine the notion of observability induced by the source cosignature[16].

Since the notion of observability associated to the target of a cosignature morphism is, in general, finer than the one associated to the source of the cosignature morphism, the notion of cosignature morphism does not give rise to an institution

[15] The notion of comonad morphism considered here generalises the standard one, as defined *e.g.* in [1] (4.5.8), being given by a pair (U, ρ) with $\mathsf{U} : \mathsf{C}' \to \mathsf{C}$ and $\rho : \mathsf{UD}' \Rightarrow \mathsf{DU}$ (subject to suitable constraints), rather than by a natural transformation $\rho : \mathsf{D}' \Rightarrow \mathsf{D}$. See [4] (2.4.15) for details.

[16] Informally, states in the final $(\mathsf{C}', \mathsf{G}')$-coalgebra are not identified by the unique homomorphism into the final (C, G)-coalgebra.

w.r.t. the satisfaction of coequations up to bisimulation. However, restricting attention to a certain category of cosignature morphisms does yield an institution, as shown in the following.

Definition 1.33. An abstract cosignature morphism $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ is **horizontal** if and only if η is a natural monomorphism.

Remark 1.34. If $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ denotes a horizontal abstract cosignature morphism, then the C-arrow underlying the unique (C, G)-coalgebra homomorphism ι from the (C, G)-reduct of the final $(\mathsf{C}', \mathsf{G}')$-coalgebra to the final (C, G)-coalgebra is a monomorphism. Consequently, if $\langle D, \delta \rangle$ denotes a $(\mathsf{C}', \mathsf{G}')$-coalgebra and (K, l, r) denotes a (C, G)-coequation, and if $! : \mathsf{U}_\eta \langle D, \delta \rangle \to \langle F, \zeta \rangle$ and $!' : \langle D, \delta \rangle \to \langle F', \zeta' \rangle$ denote the unique coalgebra homomorphisms from $\mathsf{U}_\eta \langle D, \delta \rangle$ and $\langle D, \delta \rangle$ to the final (C, G)- and respectively $(\mathsf{C}', \mathsf{G}')$-coalgebras, then $\mathsf{K}\mathsf{U}_\mathsf{C}! \circ l_{\mathsf{U}_\eta \delta} = \mathsf{K}\mathsf{U}_\mathsf{C}! \circ r_{\mathsf{U}_\eta \delta}$ is equivalent to $\mathsf{R}\mathsf{K}\mathsf{U}\mathsf{U}_{\mathsf{C}'}!' \circ l^\flat_{\mathsf{U}_\eta \delta} = \mathsf{R}\mathsf{K}\mathsf{U}\mathsf{U}_{\mathsf{C}'}!' \circ r^\flat_{\mathsf{U}_\eta \delta}$. (This follows by standard properties of adjunctions, using the observation that $\mathsf{U}_\mathsf{C} \iota$ is a monomorphism together with the preservation of monomorphisms by K.)

Proposition 1.35. *Let $(\mathsf{U}, \eta) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ denote a horizontal abstract cosignature morphism, let $\langle D, \delta \rangle$ denote a $(\mathsf{C}', \mathsf{G}')$-coalgebra, and let e denote a (C, G)-coequation of form (K, l, r). Then, $\mathsf{U}_\eta \langle D, \delta \rangle \models^{\mathsf{b}} e$ if and only if $\langle D, \delta \rangle \models^{\mathsf{b}} \eta(e)$.*

Proof. Similar to the proof of Proposition 1.30. Remark 1.34 is also used. □

Remark 1.36. If the destructor cosignature morphism $\phi : (H, \Delta) \to (H', \Delta')$ is such that $\delta' \in \Delta'_{\phi(h), w'}$ with $h \in H$ and $w' \in S'^{+}$ implies $\delta' = \phi(\delta)$ for some $\delta \in \Delta_{h,w}$ with $w \in S^+$, then the abstract cosignature morphism induced by ϕ (see Rem. 1.31) is horizontal.

We conclude this section with some remarks on the conditions defining abstract cosignatures. On the one hand, the requirement regarding the preservation of pullbacks by the endofunctors in question ensures that (finite) limits exist in the categories of coalgebras of abstract cosignatures. This is needed for instance to derive Proposition 1.22. However, many of the definitions and results in this section only use consequences of the above requirement (namely the preservation of monomorphisms by the endofunctors defining abstract cosignatures, and the construction of largest bisimulations as pullbacks in the underlying categories). For this, preservation of *weak pullbacks*[17] suffices. On the other hand, the requirement regarding the ω^{op}-continuity of the endofunctors defining abstract cosignatures ensures the existence of right adjoints to the functors taking coalgebras of abstract cosignatures to their carrier, and hence the existence of comonads induced by abstract cosignatures. Again, most of the definitions and results in this section only make use of the existence of such comonads, and not of the original requirement. The only places where this requirement is actually used are

[17] Weak pullbacks are defined similarly to standard pullbacks, except that the mediating arrow is not required to be unique.

the proof of Proposition 1.23, the proof of the fact that the induced comonads preserve monomorphisms[18], and the proof of the fact that the natural transformations defining the induced comonad morphisms are natural monomorphisms whenever the natural transformations defining the original cosignature morphisms are natural monomorphisms.

We can therefore infer that the requirements in the definition of abstract cosignatures can be relaxed in such a way that powerset endofunctors of bounded cardinality are also accounted for. Specifically, one can only require the preservation of weak pullbacks by the endofunctors defining abstract cosignatures, and the existence of right adjoints to the functors taking coalgebras of abstract cosignatures to their carrier. This also triggers a change in the definition of horizontal cosignature morphisms, which must now require that the $1'$-component of the natural transformation defining the comonad morphism induced by the cosignature morphism in question is a natural monomorphism. (The induced comonad morphism is still obtained by exploiting the couniversality of the comonad associated to the source cosignature; however, from the fact that the natural transformation defining a cosignature morphism is a natural monomorphism, one can not anymore infer that this is also the case for the natural transformation defining the induced comonad morphism.) Some of the preceding results, including those regarding the existence of limits in the categories of coalgebras of abstract cosignatures or the existence of largest subcoalgebras satisfying (sets of) coequations do not carry over to the new setting. However, the factorisation systems for the underlying categories still lift to the categories of coalgebras of abstract cosignatures, while sets of coequations still induce covarieties. Also, Propositions 1.30 and 1.35 still hold in the new setting.

2. Specifying computational structures

A framework for the specification of structures involving computation is obtained essentially by dualising the framework introduced in Section 1. However, since the abstract notion of signature that results from a complete dualisation of Definition 1.2 is too restrictive to allow for any interesting instantiations of the endofunctor involved (as polynomial endofunctors do not, in general, preserve pushouts), the conditions used in defining abstract signatures are not the exact duals of those used in defining abstract cosignatures. An outline of the resulting approach is given in the following.

Definition 2.1. An **(abstract) signature** is a pair (C, F), with C a category which is complete, cocomplete and has $(\mathsf{Epi}(\mathsf{C}), \mathsf{Mono}(\mathsf{C}))$ as a factorisation system, and with $\mathsf{F} : \mathsf{C} \to \mathsf{C}$ an endofunctor which preserves epimorphisms and colimits of ω-chains, and is such that the category $\mathsf{Alg}(\mathsf{F})$ has coequalisers[19]. An **(abstract) signature morphism** between abstract signatures (C, F) and $(\mathsf{C}', \mathsf{F}')$ is a pair

[18] This result is only needed in the dual case (namely for the proof of Prop. 3.7).

[19] The existence of coequalisers in $\mathsf{Alg}(\mathsf{C}, \mathsf{F})$ is only needed for the formulation of results dual to Propositions 1.22 and 1.23.

(U, ξ), with $\mathsf{U} : \mathsf{C}' \to \mathsf{C}$ a functor which preserves colimits and has a left adjoint L[20], and with $\xi : \mathsf{FU} \Rightarrow \mathsf{UF}'$ a natural transformation. If, in addition, ξ is a natural epimorphism, then (U, ξ) is called **horizontal**.

Definition 2.2. Let (C, F) denote an abstract signature. A (C, F)**-algebra** (respectively (C, F)**-algebra homomorphism**) is an F-algebra (F-algebra homomorphism).

The category of (C, F)-algebras and (C, F)-algebra homomorphisms is denoted $\mathsf{Alg}(\mathsf{C}, \mathsf{F})$, while the functor taking (C, F)-algebras to their carrier is denoted $\mathsf{U}_\mathsf{C} : \mathsf{Alg}(\mathsf{C}, \mathsf{F}) \to \mathsf{C}$.

Remark 2.3. Similarly to Remark 1.4, abstract signatures (C, F) induce monads (T, η, μ) on C such that $\mathsf{Alg}(\mathsf{C}, \mathsf{F}) \simeq \mathsf{Alg}(\mathsf{T})$. Moreover, since F preserves epimorphisms, so does T. Also, similarly to Remark 1.32, abstract signature morphisms $(\mathsf{U}, \xi) : (\mathsf{C}, \mathsf{F}) \to (\mathsf{C}', \mathsf{F}')$ induce monad morphisms $(\mathsf{U}, \nu) : (\mathsf{T}, \eta, \mu) \to (\mathsf{T}', \eta', \mu')$. And if, in addition, ξ is a natural epimorphism, then so is ν. (The fact that the C-arrow $\nu_{0'} : \mathsf{TU}0' \to \mathsf{UT}'0'$, with $0'$ denoting an initial C'-object, is an epimorphism formalises the fact that the notion of reachability associated to $(\mathsf{C}', \mathsf{F}')$ does not refine the notion of reachability associated to (C, F)[21].)

The existence of an initial object in C and of a left adjoint to U_C result in the existence of an initial object in $\mathsf{Alg}(\mathsf{C}, \mathsf{F})$. Hence, a notion of *reachability* of algebras can be defined.

Definition 2.4. Let (C, F) denote an abstract signature. A (C, F)-algebra $\langle A, \alpha \rangle$ is **reachable** if and only if the C-arrow underlying the unique F-algebra homomorphism from the initial (C, F)-algebra to $\langle A, \alpha \rangle$ is an epimorphism.

Also, for an abstract signature (C, F), $\mathsf{Alg}(\mathsf{C}, \mathsf{F})$ has factorisations of form $f = m \circ e$, with e defining a homomorphic image and with m defining a subalgebra. Moreover, $(\mathsf{U}_\mathsf{C}^{-1}(\mathsf{Epi}(\mathsf{C})), \mathsf{U}_\mathsf{C}^{-1}(\mathsf{Mono}(\mathsf{C})))$ defines a factorisation system for $\mathsf{Alg}(\mathsf{C}, \mathsf{F})$. (These statements follow similarly to Prop. 1.13 and respectively Cor. 1.14.)

Definition 2.5. Let (C, F) denote an abstract signature. A (C, F)**-constructor** is a pair (K, c), with $\mathsf{K} : \mathsf{C} \to \mathsf{C}$ a functor which preserves epimorphisms, and with $c : \mathsf{KU}_\mathsf{C} \Rightarrow \mathsf{U}_\mathsf{C}$ a natural transformation. A (C, F)**-equation** is a tuple (K, l, r) with (K, l) and (K, r) being (C, F)-constructors. A (C, F)-algebra $\langle A, \alpha \rangle$ **satisfies** a (C, F)-equation (K, l, r) (written $\langle A, \alpha \rangle \models (\mathsf{K}, l, r)$) if and only if $l_\alpha = r_\alpha$. Also, $\langle A, \alpha \rangle$ **satisfies** a (C, F)-equation (K, l, r) **up to reachability** (written $\langle A, \alpha \rangle \models^\mathsf{r} (\mathsf{K}, l, r)$) if and only if $l_\alpha \circ \mathsf{KU}_\mathsf{C}! = r_\alpha \circ \mathsf{KU}_\mathsf{C}!$, with $! : \langle I, \xi \rangle \to \langle A, \alpha \rangle$ denoting the unique (C, F)-algebra homomorphism from the initial (C, F)-algebra to $\langle A, \alpha \rangle$.

The notion of satisfaction of equations (up to reachability) by algebras of abstract signatures satisfies properties similar to those of the notion of satisfaction of coequations (up to bisimulation) by coalgebras of abstract cosignatures. In

[20] Consequently, U also preserves limits.

[21] Informally, states in the initial $(\mathsf{C}', \mathsf{F}')$-algebra are reachable from states in the initial (C, F)-algebra.

particular, it gives rise to an institution w.r.t. (horizontal) signature morphisms. Furthermore, results dual to Propositions 1.22 and 1.23 hold.

As one would expect, the many-sorted algebraic notions of signature, algebra, term, equation and equational satisfaction are instances of the abstract concepts thus obtained. Specifically, if (S, Σ) denotes a many-sorted signature and $\mathsf{F}_\Sigma : \mathsf{Set}^S \to \mathsf{Set}^S$ denotes the endofunctor given by:

$$(\mathsf{F}_\Sigma X)_s = \coprod_{\sigma \in \Sigma_{s_1 \ldots s_n, s}} (X_{s_1} \times \ldots \times X_{s_n}),\ X \in |\mathsf{Set}^S|,\ s \in S$$

then the category $\mathsf{Alg}(S, \Sigma)$ is isomorphic to $\mathsf{Alg}(\mathsf{Set}^S, \mathsf{F}_\Sigma)$. Also, Σ-terms $t \in T_\Sigma(\mathcal{V})_s$ with $\mathcal{V}$ consisting of variables $V_1 : s_1, \ldots, V_m : s_m$ induce $(\mathsf{Set}^S, \mathsf{F}_\Sigma)$-constructors $(\mathsf{K}, \bar{t})$, with $\mathsf{K} : \mathsf{Set}^S \to \mathsf{Set}^S$ being given by:

$$(\mathsf{K}X)_{s'} = \begin{cases} X_{s_1} \times \ldots \times X_{s_m} & \text{if } s' = s \\ \emptyset & \text{otherwise} \end{cases}$$

and with $\bar{t} : \mathsf{KU} \Rightarrow \mathsf{U}$ (where $\mathsf{U} : \mathsf{Alg}(\mathsf{Set}^S, \mathsf{F}_\Sigma) \to \mathsf{Set}^S$ denotes the functor taking $(\mathsf{Set}^S, \mathsf{F}_\Sigma)$-algebras to their carrier) being given by:

$$(\bar{t}_{\langle A, \alpha \rangle})_{s'} = \begin{cases} t_A & \text{if } s' = s \\ ! : \emptyset \to A_{s'} & \text{otherwise} \end{cases}$$

for each $(\mathsf{Set}^S, \mathsf{F}_\Sigma)$-algebra $\langle A, \alpha \rangle$ with A its associated Σ-algebra. Finally, unconditional Σ-equations e of form $(\forall\, \mathcal{V})\ l = r$ induce $(\mathsf{Set}^S, \mathsf{F}_\Sigma)$-equations $(\mathsf{K}, \bar{l}, \bar{r})$, with $\mathsf{K} : \mathsf{Set}^\mathsf{S} \to \mathsf{Set}^\mathsf{S}$ being as before, and with $\bar{l}, \bar{r} : \mathsf{KU} \Rightarrow \mathsf{U}$ being the $(\mathsf{Set}^S, \mathsf{F}_\Sigma)$-constructors associated to l and r. Moreover, $A \models_\Sigma e$ is equivalent to $\langle A, \alpha \rangle \models (\mathsf{K}, \bar{l}, \bar{r})$.

Remark 2.6. Similarly to Remarks 1.5 and 1.17, many-sorted signatures whose sets of sorts have been classified into visible and hidden ones and whose operation symbols have a hidden result type (to be referred to as *constructor signatures* in what follows) are an instance of the abstract notion of signature, while equations of hidden sort over constructor signatures are an instance of the abstract notion of equation.

Example 2.7. Stacks of natural numbers are specified using a constructor signature $\Sigma_{\mathtt{ST}}$ over visible sorts $\mathtt{1}$[22] and Nat, consisting of a hidden sort Stack together with operation symbols `empty` $: \to$ `Stack`, `push` : `Stack Nat` $\to$ `Stack` and `pop` : `Stack` $\to$ `Stack`. The abstract signature associated to this constructor signature is $(\mathsf{Set}_D^{S_{\mathtt{ST}}}, \mathsf{F}_{\mathtt{ST}})$, with $S_{\mathtt{ST}}$ and D being as in Example 1.6, and with the hidden-sorted component of $\mathsf{F}_{\mathtt{ST}}$ being given by:

$$(\mathsf{F}_{\mathtt{ST}} X)_{\mathtt{Stack}} = X_1 + (X_{\mathtt{Stack}} \times X_{\mathtt{Nat}}) + X_{\mathtt{Stack}},\quad X \in \mathsf{Set}_D^{S_{\mathtt{ST}}}.$$

[22]For consistency with Example 1.6.

The hidden-sorted carrier of the initial algebra of the stack signature is isomorphic to the set of $\Sigma_{\mathtt{ST}}$-terms over D, of sort `Stack`.

Also, the correctness of stack implementations is formalised by the equations:

```
pop(empty) = empty
pop(push(s,n)) = s.
```

The hidden-sorted carrier of the initial algebra of the resulting stack specification is isomorphic to the set of $\Sigma_{\mathtt{ST}} \setminus \{\mathtt{pop}\}$-terms over D, of sort `Stack`. (The satisfaction of the two equations by this algebra results in any $\Sigma_{\mathtt{ST}}$-term over D, of sort `Stack`, being identified with some $\Sigma_{\mathtt{ST}}$-term containing no occurrence of `pop`.)

3. Specifying combined structures

Sections 1 and 2 have illustrated how coalgebra and algebra can be used to specify and reason about structures involving observation, respectively computation. The resulting frameworks are here integrated in order to account for the relationship between observations and computations in structures having both an observational and a computational component.

Our approach builds on the functorial approach to operational semantics of [18]. Specifically, we use natural transformations of form $\sigma : \mathsf{TU_C} \Rightarrow \mathsf{GTU_C}$, with $\mathsf{U_C} : \mathsf{Coalg}(\mathsf{C},\mathsf{G}) \to \mathsf{C}$ denoting the functor taking (C,G)-coalgebras to their carrier, and with (T, η, μ) denoting the monad induced by the abstract signature (C,F), to define liftings of the monad T to the category $\mathsf{Coalg}(\mathsf{C},\mathsf{G})$.

Definition 3.1. An **(abstract) lifted signature** is a tuple $(\mathsf{C},\mathsf{G},\mathsf{F},\sigma)$, with (C,G) an abstract cosignature, (C,F) an abstract signature, and $\sigma : \mathsf{TU_C} \Rightarrow \mathsf{GTU_C}$ a natural transformation, such that the following diagram commutes:

$$\begin{tikzcd}
\mathsf{T}^2\mathsf{U_C} \arrow[r, Rightarrow, "\sigma_{\mathsf{T}_\sigma}"] \arrow[d, Rightarrow, "\mu_{\mathsf{U_C}}"'] & \mathsf{GT}^2\mathsf{U_C} \arrow[d, Rightarrow, "\mathsf{G}\mu_{\mathsf{U_C}}"] \\
\mathsf{TU_C} \arrow[r, Rightarrow, "\sigma"] & \mathsf{GTU_C} \\
\mathsf{U_C} \arrow[u, Rightarrow, "\eta_{\mathsf{U_C}}"] \arrow[r, Rightarrow, "\lambda"'] & \mathsf{GU_C} \arrow[u, Rightarrow, "\mathsf{G}\eta_{\mathsf{U_C}}"']
\end{tikzcd}$$

where the natural transformation $\lambda : \mathsf{U_C} \Rightarrow \mathsf{GU_C}$ is given by: $\lambda_{\langle C,\gamma\rangle} = \gamma$ for $\langle C,\gamma\rangle \in |\mathsf{Coalg}(\mathsf{G})|$, while the functor $\mathsf{T}_\sigma : \mathsf{Coalg}(\mathsf{G}) \to \mathsf{Coalg}(\mathsf{G})$ is given by:

- $\mathsf{T}_\sigma\langle C,\gamma\rangle = \langle \mathsf{T}C, \sigma_\gamma\rangle$ for $\langle C,\gamma\rangle \in |\mathsf{Coalg}(\mathsf{G})|$
- $\mathsf{U_C}\mathsf{T}_\sigma f = \mathsf{TU_C} f$ for $f \in \parallel \mathsf{Coalg}(\mathsf{G}) \parallel$[23]

(and consequently $\mathsf{U_C}\mathsf{T}_\sigma = \mathsf{TU_C}$).

An **(abstract) lifted signature morphism** from $(\mathsf{C},\mathsf{G},\mathsf{F},\sigma)$ to $(\mathsf{C}',\mathsf{G}',\mathsf{F}',\sigma')$ is a tuple (U,τ,ξ) with $(\mathsf{U},\tau) : (\mathsf{C},\mathsf{G}) \to (\mathsf{C}',\mathsf{G}')$ an abstract cosignature morphism

[23]Naturality of σ ensures that $\mathsf{TU_C} f$ defines a G-coalgebra homomorphism.

and $(\mathsf{U}, \xi) : (\mathsf{C}, \mathsf{F}) \to (\mathsf{C}', \mathsf{F}')$ a horizontal abstract signature morphism, such that the following diagram commutes:

$$\begin{tikzcd}
\mathsf{T}\mathsf{U}_{\mathsf{C}}\mathsf{U}_{\tau} \arrow[rr, Rightarrow, "\sigma_{\mathsf{U}_{\tau}}"] \arrow[d, equal] & & \mathsf{G}\mathsf{T}\mathsf{U}_{\mathsf{C}}\mathsf{U}_{\tau} \arrow[d, equal] \\
\mathsf{T}\mathsf{U}\mathsf{U}_{\mathsf{C}'} \arrow[d, Rightarrow, "\nu_{\mathsf{U}_{\mathsf{C}'}}"'] & & \mathsf{G}\mathsf{T}\mathsf{U}\mathsf{U}_{\mathsf{C}'} \arrow[d, Rightarrow, "\mathsf{G}\nu_{\mathsf{U}_{\mathsf{C}'}}"] \\
\mathsf{U}\mathsf{T}'\mathsf{U}_{\mathsf{C}'} \arrow[r, Rightarrow, "\mathsf{U}\sigma'"'] & \mathsf{U}\mathsf{G}'\mathsf{T}'\mathsf{U}_{\mathsf{C}'} \arrow[r, Rightarrow, "\tau_{\mathsf{T}'\mathsf{U}_{\mathsf{C}'}}"'] & \mathsf{G}\mathsf{U}\mathsf{T}'\mathsf{U}_{\mathsf{C}'}
\end{tikzcd}$$

where $(\mathsf{U}, \nu) : (\mathsf{T}, \eta, \mu) \to (\mathsf{T}', \eta', \mu')$ is the monad morphism induced by the signature morphism (U, ξ).

The components of the natural transformation σ used to define a lifted signature $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ define (C, G)-coalgebra structures on (the carriers of) the free T-algebras over (the carriers of) (C, G)-coalgebras. Moreover, the constraints defining lifted signatures ensure that, for any (C, G)-coalgebra $\langle C, \gamma \rangle$, the C-arrows $\eta_C : C \to \mathsf{T}C$ and $\mu_C : \mathsf{T}^2 C \to C$ define G-coalgebra homomorphisms. This results in the tuple $(\mathsf{T}_{\sigma}, \eta, \mu)$ defining a monad on $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$. An algebra of this monad is given by a (C, G)-coalgebra $\langle C, \gamma \rangle$ together with a (C, G)-coalgebra homomorphism $\alpha : \mathsf{T}_{\sigma}\langle C, \gamma \rangle \to \langle C, \gamma \rangle$, additionally satisfying: $\alpha \circ \eta_C = 1_C$ and $\alpha \circ \mu_C = \alpha \circ \mathsf{T}\alpha$. Equivalently, a T_{σ}-algebra is given by a C-object C carrying both a G-coalgebra structure $\langle C, \gamma \rangle$ and a T-algebra structure $\langle C, \alpha \rangle$, such that α defines a G-coalgebra homomorphism from $\langle \mathsf{T}C, \sigma_{\gamma} \rangle$ to $\langle C, \gamma \rangle$. Similarly, a T_{σ}-algebra homomorphism from $\langle \langle C, \gamma \rangle, \alpha \rangle$ to $\langle \langle D, \delta \rangle, \beta \rangle$ is given by a C-arrow $f : C \to D$ defining both a G-coalgebra homomorphism from $\langle C, \gamma \rangle$ to $\langle D, \delta \rangle$ and a T-algebra homomorphism from $\langle C, \alpha \rangle$ to $\langle D, \beta \rangle$.

The models of a lifted signature $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ are taken to be the algebras of the lifted monad T_{σ}. The functor taking T_{σ}-algebras to their carrier is denoted $\mathsf{U}_{\mathsf{Coalg}(\mathsf{C},\mathsf{G})} : \mathsf{Alg}(\mathsf{T}_{\sigma}) \to \mathsf{Coalg}(\mathsf{C}, \mathsf{G})$.

The constraints defining a lifted signature morphism $(\mathsf{U}, \tau, \xi) : (\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma) \to (\mathsf{C}', \mathsf{G}', \mathsf{F}', \sigma')$ ensure that, for any $(\mathsf{C}', \mathsf{G}')$-coalgebra $\langle C', \gamma' \rangle$, the $(\mathsf{C}', \mathsf{G}')$-coalgebra structure induced by σ' on $\mathsf{T}'C'$ agrees with the (C, G)-coalgebra structure induced by σ on $\mathsf{T}\mathsf{U}C'$. This results in lifted signature morphisms $(\mathsf{U}, \tau, \xi) : (\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma) \to (\mathsf{C}', \mathsf{G}', \mathsf{F}', \sigma')$ inducing reduct functors $\mathsf{U}_{(\tau,\xi)} : \mathsf{Alg}(\mathsf{T}'_{\sigma'}) \to \mathsf{Alg}(\mathsf{T}_{\sigma})$, with $\mathsf{U}_{(\tau,\xi)}$ taking a $\mathsf{T}'_{\sigma'}$-algebra $\langle \langle C', \gamma' \rangle, \alpha' \rangle$ to the T_{σ}-algebra $\langle \langle \mathsf{U}C', \tau_{C'} \circ \mathsf{U}\gamma' \rangle, \mathsf{U}\alpha' \circ \nu_{C'} \rangle$.

Remark 3.2. In [18], monads T and comonads D are used to specify syntax and respectively behaviour, and *distributive laws* (defined as natural transformations $\lambda : \mathsf{T}\mathsf{D} \Rightarrow \mathsf{D}\mathsf{T}$ subject to certain compatibility conditions) are used to relate the two. Also, (T, D)-*bialgebras* (defined as pairs consisting of a T-algebra and a D-coalgebra with the same carrier) are used to interpret such specifications, and a notion of satisfaction of distributive laws by (T, D)-bialgebras is introduced. It is then shown in [18] that distributive laws $\lambda : \mathsf{T}\mathsf{D} \Rightarrow \mathsf{D}\mathsf{T}$ are in one-to-one correspondence with liftings $\overline{\mathsf{T}}$ of T to D-coalgebras, and moreover, the category

$\mathsf{Alg}(\overline{\mathsf{T}})$ is isomorphic to the category of (T, D)-bialgebras satisfying λ. Thus, the models of a lifted signature $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ are in one-to-one correspondence with the (T, D)-bialgebras satisfying some distributive law λ (with T and D denoting the monad and respectively comonad induced by F and respectively G).

Remark 3.3. The natural transformation $\sigma : \mathsf{TU_C} \Rightarrow \mathsf{GTU_C}$ required by the definition of lifted signatures can be given in terms of a natural transformation $\rho : \mathsf{FU_C} \Rightarrow \mathsf{G}(\mathsf{U_C}{+}\mathsf{FU_C})$ (defining the one-step observations of the results yielded by one-step computations as either zero- or one-step computations on observations of their arguments)[24]. Also, if $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ and $(\mathsf{C}', \mathsf{G}', \mathsf{F}', \sigma')$ are the lifted signatures induced by the natural transformations $\rho : \mathsf{FU_C} \Rightarrow \mathsf{G}(\mathsf{U_C} + \mathsf{FU_C})$ and respectively $\rho' : \mathsf{F}'\mathsf{U_{C'}} \Rightarrow \mathsf{G}'(\mathsf{U_{C'}} + \mathsf{F}'\mathsf{U_{C'}})$, and if $(\mathsf{U}, \tau) : (\mathsf{C}, \mathsf{G}) \to (\mathsf{C}', \mathsf{G}')$ is a cosignature morphism and $(\mathsf{U}, \xi) : (\mathsf{C}, \mathsf{F}) \to (\mathsf{C}', \mathsf{F}')$ is a horizontal signature morphism, such that the following diagram commutes:

$$\begin{tikzcd}
\mathsf{FUU_{C'}} \arrow[r, equal] \arrow[dd, Rightarrow, "\xi_{\mathsf{U_{C'}}}"'] & \mathsf{FU_C U_\tau} \arrow[r, Rightarrow, "\rho_{\mathsf{U}_\tau}"] & \mathsf{G}(\mathsf{U_C U_\tau} + \mathsf{FU_C U_\tau}) \arrow[r, equal] & \mathsf{G}(\mathsf{UU_{C'}} + \mathsf{FUU_{C'}}) \arrow[d, Rightarrow, "\mathsf{G}(1_{\mathsf{UU_{C'}}} + \xi_{\mathsf{U_{C'}}})"] \\
& & & \mathsf{G}(\mathsf{UU_{C'}} + \mathsf{UF'U_{C'}}) \arrow[d, Rightarrow, "\mathsf{G}[\mathsf{U}\iota_1', \mathsf{U}\iota_2']"] \\
\mathsf{UF'U_{C'}} \arrow[r, Rightarrow, "\mathsf{U}\rho'"'] & \mathsf{UG}'(\mathsf{U_{C'}} + \mathsf{F'U_{C'}}) \arrow[rr, Rightarrow, "\tau_{\mathsf{U_{C'}} + \mathsf{F'U_{C'}}}"'] & & \mathsf{GU}(\mathsf{U_{C'}} + \mathsf{F'U_{C'}})
\end{tikzcd}$$

then (U, τ, ξ) defines a lifted signature morphism from $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ to $(\mathsf{C}', \mathsf{G}', \mathsf{F}', \sigma')$. (The above constraint on ρ, ρ', ξ and τ ensures that σ, σ', ν and τ satisfy the constraint required by Def. 3.1.) See [4] for proofs of the above statements.

Example 3.4. The relationship between computing stack states and observing them is specified using a lifted signature $\mathtt{STACK} = (\mathsf{Set}_D^{S_{\mathtt{ST}}}, \mathsf{G}_{\mathtt{ST}}, \mathsf{F}_{\mathtt{ST}}, \sigma)$, with $S_{\mathtt{ST}}$, D, $\mathsf{G}_{\mathtt{ST}}$ and $\mathsf{F}_{\mathtt{ST}}$ being as in Examples 1.6 and respectively 2.7, and with the natural transformation $\sigma : \mathsf{T}_{\mathtt{ST}}\mathsf{U} \Rightarrow \mathsf{G}_{\mathtt{ST}}\mathsf{T}_{\mathtt{ST}}\mathsf{U}$ (where $\mathsf{T}_{\mathtt{ST}}$ denotes the monad induced by $\mathsf{F}_{\mathtt{ST}}$, and where $\mathsf{U} : \mathsf{Coalg}(\mathsf{Set}_D^{S_{\mathtt{ST}}}, \mathsf{G}_{\mathtt{ST}}) \to \mathsf{Set}_D^{S_{\mathtt{ST}}}$ denotes the functor taking $(\mathsf{Set}_D^{S_{\mathtt{ST}}}, \mathsf{G}_{\mathtt{ST}})$-coalgebras to their carrier) being induced by the natural transformation $\rho : \mathsf{F}_{\mathtt{ST}}\mathsf{U} \Rightarrow \mathsf{G}_{\mathtt{ST}}(\mathsf{U} + \mathsf{F}_{\mathtt{ST}}\mathsf{U})$ whose hidden-sorted component is given by:

$$(\rho_{\langle C,\gamma\rangle})_{\mathtt{Stack}}(\iota_1(*)) = \langle \iota_1(*_D), \iota_1(*_D)\rangle$$

$$(\rho_{\langle C,\gamma\rangle})_{\mathtt{Stack}}(\iota_2(\langle c, d\rangle)) = \langle \iota_2(d), \iota_2(c)\rangle$$

$$\pi_1((\rho_{\langle C,\gamma\rangle})_{\mathtt{Stack}}(\iota_3(c))) = \begin{cases} \iota_1(*_D) & \text{if } [\iota_1, \mathtt{top}_C](\mathtt{rest}_C(c)) \in \iota_1(C_1) \\ \iota_2(d) & \text{if } [\iota_1, \mathtt{top}_C](\mathtt{rest}_C(c)) = \iota_2(d) \in \iota_2(C_{\mathtt{Nat}}) \end{cases}$$

$$\pi_2((\rho_{\langle C,\gamma\rangle})_{\mathtt{Stack}}(\iota_3(c))) = \begin{cases} \iota_1(*_D) & \text{if } [\iota_1, \mathtt{rest}_C](\mathtt{rest}_C(c)) \in \iota_1(C_1) \\ \iota_2(c') & \text{if } [\iota_1, \mathtt{rest}_C](\mathtt{rest}_C(c)) = \iota_2(c') \in \iota_2(C_{\mathtt{Stack}}) \end{cases}$$

[24]This observation is based on the approach in [18].

for $\langle C, \gamma\rangle \in |\mathsf{Coalg}(\mathsf{Set}_D^{S_{\mathsf{ST}}}, \mathsf{G}_{\mathsf{ST}})|$ with C its associated Δ_{ST}-coalgebra. The natural transformation ρ can alternatively be specified using the following *constraints*[25]:

```
[Z,N]top.empty = Z.*
[Z,S]rest.empty = Z.*

[Z,N]top.push(s,n) = N.n
[Z,S]rest.push(s,n) = S.s

[Z,N]top.pop(s) = Z.* if [Z,[Z,N]top]rest.s = Z.z
[Z,N]top.pop(s) = N.n if [Z,[Z,N]top]rest.s = N.n
[Z,S]rest.pop(s) = Z.* if [Z,[Z,S]rest]rest.s = Z.z
[Z,S]rest.pop(s) = S.s' if [Z,[Z,S]rest]rest.s = S.s'
```

defining the results of Δ_{ST}-observers on Σ_{ST}-constructors in terms of particular observations on the arguments to these constructors. In order to allow specific observations on the arguments to the constructors to be used in defining the value of observers on the results yielded by constructors, a form of case analysis (on the types of the results yielded by observations) is incorporated in constraints. For instance, when defining the value of the observer `top` on the result yielded by the constructor `pop`, a distinction is made between stacks containing at most one element and stacks containing at least two elements. For, in the first case, the resulting stack is empty, and therefore observing it using `[Z,N]top` yields a result of type `1`, whereas in the second case the resulting stack contains at least one element, and hence observing it using `[Z,N]top` yields the same result (of type `Nat`) as when observing the original stack using `[Z,[Z,N]top]rest`. This is captured by the use of the variable `n` both in the condition of the second constraint defining the value of `top` on the result yielded by `pop` (where a value is provided for `n`), and in the rhs of this constraint (where the value of `n` is used to define the value of the lhs of the constraint).

For a lifted signature $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$, the existence of finite limits in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ (see Cor. 1.10) results in the existence of finite limits in $\mathsf{Alg}(\mathsf{T}_\sigma)$.

Proposition 3.5. *Let* $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ *denote a lifted signature. Then,* $\mathsf{Alg}(\mathsf{T}_\sigma)$ *has finite limits and* $\mathsf{U}_{\mathsf{Coalg}(\mathsf{C},\mathsf{G})}$ *preserves them.*

Proof. The conclusion follows from Corollary 1.10 together with the observation that $\mathsf{U}_{\mathsf{Coalg}(\mathsf{C},\mathsf{G})}$ creates limits. $\square$

In particular, $\mathsf{Alg}(\mathsf{T}_\sigma)$ has a final object, given by the T_σ-algebra $\langle\langle F, \zeta\rangle, !_{\sigma_\zeta}\rangle$, with $\langle F, \zeta\rangle$ denoting a final (C, G)-coalgebra, and with $!_{\sigma_\zeta} : \langle \mathsf{T}F, \sigma_\zeta\rangle \to \langle F, \zeta\rangle$ denoting the unique (C, G)-coalgebra homomorphism of $\langle \mathsf{T}F, \sigma_\zeta\rangle$ into $\langle F, \zeta\rangle$. The final T_σ-algebra provides an interpretation of arbitrary computations on abstract states.

Also, kernel pairs exist in $\mathsf{Alg}(\mathsf{T}_\sigma)$ and are created by $\mathsf{U}_{\mathsf{Coalg}(\mathsf{C},\mathsf{G})}$. This yields a T_σ-algebra structure on (C, G)-bisimilarity on the underlying coalgebra of a T_σ-algebra, in such a way that the coalgebra homomorphisms defining the bisimilarity

[25] See [4] for a formal definition of the notion of constraint and of its associated notion of satisfaction.

relation become T_σ-algebra homomorphisms. (Recall from Rem. 1.11 that bisimilarity is given by the kernel pair of the unique coalgebra homomorphism into the final coalgebra.) That is, (C, G)-bisimilarity on the underlying coalgebra of a T_σ-algebra is preserved by the T-algebra structure.

On the other hand, the existence of an initial object in $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ (following from the existence of an initial object in C) results in the existence of an initial T_σ-algebra.

Proposition 3.6. *Let* $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ *denote a lifted signature. Then,* $\mathsf{Alg}(\mathsf{T}_\sigma)$ *has an initial object.*

The initial object is given by the T_σ-algebra $\langle \mathsf{T}_\sigma\langle 0, !_{\mathsf{G}0}\rangle, \mu_0\rangle = \langle\langle \mathsf{T}0, \sigma_{!_{\mathsf{G}0}}\rangle, \mu_0\rangle$, where 0 denotes an initial C-object (and consequently $!_{\mathsf{G}0} : 0 \to \mathsf{G}0$ defines an initial (C, G)-coalgebra). This T_σ-algebra provides an observational structure on ground computations.

Now recall from Corollary 1.14 that $(\mathsf{U}_\mathsf{C}^{-1}(\mathsf{Epi}(\mathsf{C})), \mathsf{U}_\mathsf{C}^{-1}(\mathsf{Mono}(\mathsf{C})))$ defines a factorisation system for $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$. Also, since T preserves epimorphisms (see Rem. 2.3) and since U_C preserves as well as reflects epimorphisms (see Prop. 1.12), it follows that T_σ also preserves epimorphisms. These observations result in the existence in $\mathsf{Alg}(\mathsf{T}_\sigma)$ of factorisations of form $f = m \circ e$, with e defining a T_σ-homomorphic image and with m defining a T_σ-subalgebra.

Proposition 3.7. *Let* $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ *denote a lifted signature, let* f *denote a* T_σ*-algebra homomorphism, and let* $\mathsf{U}_{\mathsf{Coalg}(\mathsf{C},\mathsf{G})} f = m \circ e$ *denote the factorisation of* f *resulting from Proposition 1.13. Then,* e *defines a* T_σ*-homomorphic image of the domain of* f*, while* m *defines a* T_σ*-subalgebra of the codomain of* f*.*

Proof. Similar to the proof of Proposition 1.13. □

Corollary 3.8. *The factorisation system for* $\mathsf{Coalg}(\mathsf{C}, \mathsf{G})$ *given by Corollary 1.14 lifts to a factorisation system for* $\mathsf{Alg}(\mathsf{T}_\sigma)$*.*

Proof. Similar to the proof of Corollary 1.14. □

By taking the T_σ-algebra homomorphism f in the statement of Proposition 3.7 to be the unique homomorphism from the initial T_σ-algebra to an arbitrary one, one obtains, for each T_σ-algebra $\langle\langle C, \gamma\rangle, \alpha\rangle$, a T_σ-subalgebra of $\langle\langle C, \gamma\rangle, \alpha\rangle$ which is reachable. This T_σ-algebra will be referred to as the *reachable subalgebra* of $\langle\langle C, \gamma\rangle, \alpha\rangle$. (Its uniqueness up to isomorphism is guaranteed by Cor. 1.14.)

Once the relationship between computations and observations has been specified by means of a lifted signature, abstract equations and coequations can be used to formalise correctness properties of the specified structures. Specifically, high-level requirements referring to the equivalence of computations can be captured by equations, whereas low-level requirements regarding system invariants can be captured by coequations. Since the interest is in the *observable* result of *ground* computations, the associated notions of satisfaction abstract away bisimilar, respectively unreachable behaviours.

Definition 3.9. Let $(\mathsf{C},\mathsf{G},\mathsf{F},\sigma)$ denote a lifted signature. A T_σ-algebra $\langle\langle C,\gamma\rangle,\alpha\rangle$ **satisfies** a (C,G)-coequation (K,l,r) **up to reachability** (written $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} (\mathsf{K},l,r)$) if and only if $l_\gamma \circ \mathsf{U_C U_{Coalg(C,G)}}! = r_\gamma \circ \mathsf{U_C U_{Coalg(C,G)}}!$, with ! denoting the unique T_σ-algebra homomorphism from the initial T_σ-algebra to $\langle\langle C,\gamma\rangle,\alpha\rangle$.

Also, $\langle\langle C,\gamma\rangle,\alpha\rangle$ **satisfies** a (C,F)-equation (K',l',r') **up to bisimulation** (written $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} (\mathsf{K}',l',r')$) if and only if $\mathsf{U_C U_{Coalg(C,G)}}!' \circ l'_{\alpha'} = \mathsf{U_C U_{Coalg(C,G)}}!' \circ r'_{\alpha'}$, with !′ denoting the unique T_σ-algebra homomorphism from $\langle\langle C,\gamma\rangle,\alpha\rangle$ to the final T_σ-algebra, and with $\langle C,\alpha'\rangle$ denoting the (C,F)-algebra induced by the T-algebra $\langle C,\alpha\rangle$.

Standard satisfaction of coequations (respectively equations) by the underlying coalgebra (algebra) of a T_σ-algebra implies their satisfaction up to reachability (up to bisimulation) by the T_σ-algebra in question. That is, $\langle C,\gamma\rangle \models (\mathsf{K},l,r)$ implies $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} (\mathsf{K},l,r)$, while $\langle C,\alpha'\rangle \models (\mathsf{K}',l',r')$ implies $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} (\mathsf{K}',l',r')$. Moreover, if the underlying algebra (coalgebra) is reachable (observable), then the converse also holds. For, in this case, $\mathsf{U_C U_{Coalg(C,G)}}!$ (respectively $\mathsf{U_C U_{Coalg(C,G)}}!'$) is an epimorphism (monomorphism).

Since $\mathsf{U_C}$ preserves kernel pairs, it follows that $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} (\mathsf{K}',l',r')$ holds if and only if $\langle l'_{\alpha'}, r'_{\alpha'}\rangle$ factors through $\langle r_1,r_2\rangle$:

$$\begin{tikzcd}
\mathsf{K}C \arrow[r, "\langle l'_{\alpha'}, r'_{\alpha'}\rangle"] \arrow[dr, dashed, "c"'] & C \times C \\
& R \arrow[u, "\langle r_1, r_2\rangle"']
\end{tikzcd}$$

with $\langle R,r_1,r_2\rangle$ denoting (C,G)-bisimilarity on $\langle C,\gamma\rangle$.

The maximality of bisimilarity amongst the bisimulations on a given coalgebra yields a coinductive technique for proving the satisfaction of equations up to bisimulation. Specifically, proving that a (C,F)-equation holds, up to bisimulation, in a full subcategory $\mathcal{A}$ of $\mathsf{Alg}(\mathsf{T}_\sigma)$ can be reduced to exhibiting a generic (C,G)-bisimulation[26] $\langle \mathsf{R},\pi_1,\pi_2\rangle$ on $\mathsf{U_{Coalg(C,G)}}(\mathcal{A})$, such that $\langle l'_{\alpha'}, r'_{\alpha'}\rangle$ factors through $\langle \pi_{1,\gamma},\pi_{2,\gamma}\rangle$ for any $\langle\langle C,\gamma\rangle,\alpha\rangle \in |\mathcal{A}|$.

$$\begin{tikzcd}
\mathsf{K}C \arrow[r, "\langle l'_{\alpha'}, r'_{\alpha'}\rangle"] \arrow[dr, dashed, "c"] \arrow[ddr, dashed, "d"'] & C \times C \\
& R \arrow[u] \\
& \mathsf{R}\gamma \arrow[u, dashed, "e"'] \arrow[uu, bend right=60, "\langle \pi_{1,\gamma}, \pi_{2,\gamma}\rangle"']
\end{tikzcd}$$

For this, in turn, yields a C-arrow $d : \mathsf{K}C \to \mathsf{R}\gamma$ such that $\langle \pi_{1,\gamma},\pi_{2,\gamma}\rangle \circ d = \langle l'_{\alpha'}, r'_{\alpha'}\rangle$. Also, the maximality of (C,G)-bisimilarity on $\langle C,\gamma\rangle$ yields a C-arrow $e : \mathsf{R}\gamma \to R$. Then, c is taken to be $e \circ d$.

[26] See Definition 1.28.

Example 3.10. The technique previously described can be used to show that the equations given in Example 2.7 hold, up to bisimulation, in all the algebras of the lifted signature of stacks. Specifically, the satisfaction of these equations can be inferred by exhibiting a generic bisimulation on the coalgebras underlying `STACK`-algebras, which, in addition, relates the lhs and rhs of each of the two equations. For a `STACK`-algebra $C = \langle\langle C, \gamma\rangle, \alpha\rangle$, let R_C denote the binary relation on C whose visible-sorted components are given by the equality relations, and whose hidden-sorted component is the least binary relation on $C_{\mathtt{Stack}}$ satisfying:

- $c \; R_{C,\mathtt{Stack}} \; c$, for each $c \in C_{\mathtt{Stack}}$;
- $\mathtt{pop}_C(\mathtt{empty}_C) \; R_{C,\mathtt{Stack}} \; \mathtt{empty}_C$;
- $\mathtt{pop}_C(\mathtt{push}_C(c,d)) \; R_{C,\mathtt{Stack}} \; c$, for each $c \in C_{\mathtt{Stack}}$ and each $d \in C_{\mathtt{Nat}}$.

The relations R_C define a generic bisimulation on the coalgebras underlying `STACK`-algebras – this follows from the definition of ρ in Example 3.4, or, alternatively, from the satisfaction of the constraints given in Example 3.4 by `STACK`-algebras. Moreover, for any `STACK`-algebra C, R_C relates the interpretations in C of the lhs and rhs of the two equations. It therefore follows by the previous remarks that $C \models^{\mathsf{b}}$ `pop(empty) = empty` and $C \models^{\mathsf{b}}$ `pop(push(s,n)) = s` hold for any `STACK`-algebra C.

Example 3.11. The coequations formalising the stack invariant (see Ex. 1.18) hold, up to reachability, in all `STACK`-algebras. However, proving this requires further insights into the notion of reachability under `empty`, `push` and `pop`. For, the fact that these coequations hold in a state `s` does not guarantee that they hold in `pop(s)`, and therefore straightforward induction can not be used to show that the coequations hold in all reachable states. However, the observation that the equations: `pop(empty) = empty` and `pop(push(s,n)) = s` hold, up to bisimulation, in all `STACK`-algebras (see Ex. 3.10) allows one to reduce proving that the stack invariant holds up to reachability under `empty`, `push` and `pop` to proving that the stack invariant holds up to reachability under `empty` and `push` only. For, from the satisfaction up to bisimulation of the above equations, one can infer that any stack state reachable under `empty`, `push` and `pop` is bisimilar to a stack state reachable under `empty` and `push` only. Then, the satisfaction of the stack invariant follows from this invariant holding in `empty` and being preserved by `push`, together with the observation that the coequations defining the invariant hold in a state `s` whenever they hold in a state bisimilar to `s`.

The following result further justifies the use of inductive and coinductive techniques in proving the satisfaction of coequations up to reachability, and respectively of equations up to bisimulation, by T_σ-algebras.

Proposition 3.12. *Let* $(\mathsf{C}, \mathsf{G}, \mathsf{F}, \sigma)$ *denote a lifted signature. The following hold:*

1. *A* (C, G)*-coequation is satisfied (up to reachability) by the initial* T_σ*-algebra precisely when it is satisfied up to reachability by any* T_σ*-algebra;*
2. *A* (C, F)*-equation is satisfied (up to bisimulation) by the final* T_σ*-algebra precisely when it is satisfied up to bisimulation by any* T_σ*-algebra.*

Proof. The conclusion follows immediately using the naturality of (C,G)-observers and respectively of (C,F)-constructors. $\square$

In assigning suitable denotations to lifted signatures, neither final nor initial algebras seem appropriate – the former are not reachable, whereas the underlying coalgebras of the latter are not observable. However, according to Proposition 3.7, the reachable subalgebra of the final T_σ-algebra is reachable, while its underlying coalgebra is observable[27]. Moreover, this algebra has the property that it satisfies (in the standard sense) precisely those equations which are satisfied up to bisimulation by the initial algebra, and precisely those coequations which are satisfied up to reachability by the final coalgebra.

Proposition 3.13. *Let* $(\mathsf{C},\mathsf{G},\mathsf{F},\sigma)$ *denote a lifted signature, and let* $! = m \circ e$*, with* $e : \langle\langle \mathsf{T}0, \sigma_{!_{\mathsf{G}0}}\rangle, \mu_0\rangle \to \langle\langle R,\gamma\rangle,\alpha\rangle$ *and* $m : \langle\langle R,\gamma\rangle,\alpha\rangle \to \langle\langle F,\zeta\rangle, !_{\sigma_\zeta}\rangle$*, denote the factorisation of the unique* T_σ*-algebra homomorphism from the initial* T_σ*-algebra to the final one resulting from Proposition 3.7. Then, the following hold:*

1. *if* (K,l,r) *denotes a* (C,G)*-coequation, then* $\langle\langle F,\zeta\rangle, !_{\sigma_\zeta}\rangle \models^{\mathsf{r}} (\mathsf{K},l,r)$ *is equivalent to* $\langle\langle R,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} (\mathsf{K},l,r)$*, as well as to* $\langle R,\gamma\rangle \models (\mathsf{K},l,r)$*;*
2. *if* (K',l',r') *denotes a* (C,F)*-equation, then* $\langle\langle \mathsf{T}0, \sigma_{!_{\mathsf{G}0}}\rangle, \mu_0\rangle \models^{\mathsf{b}} (\mathsf{K}',l',r')$ *is equivalent to* $\langle\langle R,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} (\mathsf{K}',l',r')$*, as well as to* $\langle R,\alpha\rangle \models (\mathsf{K}',l',r')$*.*

Proof (sketch). By Propositions 3.7 and 1.12, $\mathsf{U}_{\mathsf{C}}\mathsf{U}_{\mathsf{Coalg(C,G)}}e$ is an epimorphism, while $\mathsf{U}_{\mathsf{C}}\mathsf{U}_{\mathsf{Coalg(C,G)}}m$ is a monomorphism. The conclusion then follows from the definitions of $\models^{\mathsf{b}}$ and respectively $\models^{\mathsf{r}}$, using the naturality of l and r. $\square$

The requirement that the signature morphisms underlying lifted signature morphisms are horizontal results in the notions of reachability associated to the source and target of such morphisms being essentially the same (see Rem. 2.3). This, in turn, yields an institution w.r.t. the satisfaction of coequations up to reachability by algebras of lifted signatures.

Theorem 3.14. *Let* $(\mathsf{U},\tau,\xi) : (\mathsf{C},\mathsf{G},\mathsf{F},\sigma) \to (\mathsf{C}',\mathsf{G}',\mathsf{F}',\sigma')$ *denote a lifted signature morphism, let* $\langle\langle C,\gamma\rangle,\alpha\rangle$ *denote a* $\mathsf{T}'_{\sigma'}$*-algebra, and let* (K,l,r) *denote a* (C,G)*-coequation. Then,* $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} \tau(\mathsf{K},l,r)$ *if and only if* $\mathsf{U}_{(\tau,\xi)}\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} (\mathsf{K},l,r)$*.*

Proof. Let $! : \mathsf{T}0 \to \mathsf{U}C$ and $!' : \mathsf{T}'0' \to C$ denote the C- and respectively C'-arrows underlying the unique T_σ- and $\mathsf{T}'_{\sigma'}$-algebra homomorphisms from the initial

[27] Proposition 1.12 is also used here.

T_σ- and $\mathsf{T}'_{\sigma'}$-algebras to $\mathsf{U}_{(\tau,\xi)}\langle\langle C,\gamma\rangle,\alpha\rangle$ and $\langle\langle C,\gamma\rangle,\alpha\rangle$. Then, the following holds:

$$\begin{aligned}
\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} \tau(\mathsf{K},l,r) &\Leftrightarrow \text{(definition of } \models^{\mathsf{r}})\\
l^{\flat}_{\mathsf{U}_\tau\gamma}\circ !' = r^{\flat}_{\mathsf{U}_\tau\gamma}\circ !' &\Leftrightarrow (\mathsf{U} \dashv \mathsf{R})\\
l_{\mathsf{U}_\tau\gamma}\circ \mathsf{U}!' = r_{\mathsf{U}_\tau\gamma}\circ \mathsf{U}!' &\Leftrightarrow (\nu_{0'} \text{ is epi})\\
l_{\mathsf{U}_\tau\gamma}\circ \mathsf{U}!'\circ\nu_{0'} = r_{\mathsf{U}_\tau\gamma}\circ \mathsf{U}!'\circ\nu_{0'} &\Leftrightarrow (\mathsf{U}0'=0,\ \mathsf{U}!'\circ\nu_{0'}=!)\\
l_{\mathsf{U}_\tau\gamma}\circ ! = r_{\mathsf{U}_\tau\gamma}\circ ! &\Leftrightarrow \text{(definition of } \models^{\mathsf{r}})\\
\mathsf{U}_{(\tau,\xi)}\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{r}} (\mathsf{K},l,r).&
\end{aligned}$$

□

A similar condition needs to be imposed to the cosignature morphisms underlying lifted signature morphisms in order to obtain an institution w.r.t. the satisfaction of equations up to bisimulation by algebras of lifted signatures.

Definition 3.15. A lifted signature morphism (U,τ,ξ) is **horizontal** if and only if (U,τ) is horizontal.

The notions of observability associated to the source and target signatures of horizontal lifted signature morphisms are essentially the same (see Rem. 1.34). This, in turn, yields an institution.

Theorem 3.16. *Let* $(\mathsf{U},\tau,\xi) : (\mathsf{C},\mathsf{G},\mathsf{F},\sigma) \to (\mathsf{C}',\mathsf{G}',\mathsf{F}',\sigma')$ *denote a horizontal lifted signature morphism, let* $\langle\langle C,\gamma\rangle,\alpha\rangle$ *denote a* $\mathsf{T}'_{\sigma'}$*-algebra, and let* (K,l,r) *denote a* (C,F)*-equation. Then,* $\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} \xi(\mathsf{K},l,r)$ *if and only if* $\mathsf{U}_{(\tau,\xi)}\langle\langle C,\gamma\rangle,\alpha\rangle \models^{\mathsf{b}} (\mathsf{K},l,r)$.

Proof. Similar to the proof of Theorem 3.14. □

4. Summary of results

A framework for the specification of state-based, dynamical systems has been obtained by suitably integrating two separate frameworks for the specification of observational and respectively computational structures. First, abstract notions of cosignature, observer and coequation have been used to specify particular kinds of observational structures and to further constrain these structures. Coequations have been shown to induce subcoalgebras of given coalgebras on the one hand and covarieties on the other, while the resulting specification logic has been shown to be an institution. Similar results have been formulated for a notion of satisfaction of coequations up to bisimulation. Next, an algebraic framework for the specification of computational structures has been obtained essentially by dualising the previously-obtained coalgebraic framework. Finally, the two frameworks have been integrated in order to account for systems having both an observational and a computational component. Following [18], liftings of monads induced by abstract signatures to categories of coalgebras of abstract cosignatures have been used to interpret computations on the semantic domains induced by the observational component. In particular, such an approach resulted in a compatibility between observational and computational features. Abstract equations and coequations have been used to formalise the equivalence of computations and respectively invariants on the structure of systems, with the associated notions of satisfaction abstracting away observationally indistinguishable and respectively unreachable behaviours. Suitable choices for the notion of signature morphism have been shown to yield institutions w.r.t. both notions of satisfaction. The use of coinductive and respectively inductive techniques for correctness proofs has also been illustrated.

References

[1] F. Borceux, *Handbook of Categorial Algebra*, Vol. II. CUP, Cambridge (1994).

[2] C. Cîrstea, A Coequational Approach to Specifying Behaviours, edited by B. Jacobs and J. Rutten, *Coalgebraic Methods in Computer Science*. Elsevier Science, *Electron. Notes Theor. Comput. Sci.* **19** (1999) 173-194.

[3] C. Cîrstea, An Algebra-Coalgebra Framework for System Specification, edited by H. Reichel, *Coalgebraic Methods in Computer Science*. Elsevier Science, *Electron. Notes Theor. Comput. Sci.* **33** (2000) 81-112.

[4] C. Cîrstea, *Integrating Observations and Computations in the Specification of State-Based, Dynamical Systems*, Ph.D. Thesis. University of Oxford (2000) `http://www.comlab.ox.ac.uk/oucl/work/corina.cirstea/thesis.html`

[5] A. Corradini, A Completeness Result for Equational Deduction in Coalgebraic Specification, edited by F. Parisi–Presicce, *Recent Trends in Algebraic Development Techniques*. Springer, *Lecture Notes in Comput. Sci.* **1376** (1998) 190-205.

[6] A. Corradini, R. Heckel and U. Montanari, From SOS Specifications to Structured Coalgebras: How to Make Bisimulation a Congruence, edited by B. Jacobs and J. Rutten, *Coalgebraic Methods in Computer Science*. Elsevier Science, *Electron. Notes Theor. Comput. Sci.* **19** (1999) 149-172.

[7] R. Diaconescu, *Behavioural Coherence in Object-Oriented Algebraic Specification*, Technical Report IS-RR-98-0017F. Japan Advanced Institute for Science and Technology (1998).

[8] J. Goguen and R. Burstall, Institutions: Abstract Model Theory for Specification and Programming. *J. ACM* **39** (1992) 95-146.

[9] J. Goguen and G. Malcolm, A Hidden Agenda. *Theoret. Comput. Sci.* **245** (2000) 55-101.

[10] R. Hennicker and M. Bidoit, Observational Logic, in *Proc. AMAST '98*. Springer, *Lecture Notes in Comput. Sci.* **1548** (1999) 263-277.

[11] R. Hennicker and A. Kurz, (Ω, Ξ)-Logic: On the Algebraic Extension of Coalgebraic Specifications, edited by B. Jacobs, L. Moss, H. Reichel and J. Rutten, *Coalgebraic Methods in Computer Science*. Elsevier Science, *Electron. Notes Theor. Comput. Sci.* **19** (1999) 195-211.

[12] B. Jacobs, Objects and Classes, Coalgebraically, edited by B. Freitag, C.B. Jones, C. Lengauer and H.-J. Schek, *Object Orientation with Parallelism and Persistence*. Kluwer Academic Publishers (1996) 83-103.

[13] B. Jacobs, Invariants, Bisimulations and the Correctness of Coalgebraic Refinements, edited by M. Johnson, *Algebraic Methodology and Software Technology*. Springer, *Lecture Notes in Comput. Sci.* **1349** (1997) 276-291.

[14] P. Padawitz, Swinging Types = Functions + Relations + Transition Systems. *Theoret. Comput. Sci.* **243** (2000) 93-165.

[15] H. Reichel, An Approach to Object Semantics Based on Terminal Coalgebras. *Math. Structures Comput. Sci.* **5** (1995) 129-152.

[16] G. Roşu and J. Goguen, Hidden Congruent Deduction, edited by R. Caferra and G. Salzer, *Automated Deduction in Classical and Non-Classical Logics*. Springer, *Lecture Notes in Comput. Sci.* **1761** (2000) 251-266.
[17] J. Rutten, Universal Coalgebra: A Theory of Systems. *Theoret. Comput. Sci.* **249** (2000) 3-80.
[18] D. Turi and G. Plotkin, Towards a Mathematical Operational Semantics, in *Proc. LICS* (1997) 280-291.

Accepted April 25, 2001.