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

Corina Cîrstea

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...
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 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
calgebraic 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 struc-
tures. 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 ap-
proach in [17] is here specialised in order to give a categorical account of equational
calgebraic approaches to specification\(^3\). A framework which unifies some of the
existing equational approaches to system specification, including [2,5,9,12], is in-
troduced 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
calgebras of the cosignature according to their particular interpretation for the
specified structure, and coequation over a cosignature, used to constrain the coal-
gebras 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, subcoalge-
bra, 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 $G : C \to C$ denote an arbitrary endofunctor. A $G$-coalgebra is a tuple $(C, \gamma)$, with $C$ a $C$-object and $\gamma : C \to GC$ a $C$-arrow, while a $G$-coalgebra homomorphism between $G$-coalgebras $(C, \gamma)$ and $(D, \delta)$ is a $C$-arrow $f : C \to D$ additionally satisfying $\delta \circ f = Gf \circ \gamma$. A $G$-bisimulation between $G$-coalgebras $(C, \gamma)$ and $(D, \delta)$ is a relation (see [1], p. 101) $(R, r_1, r_2)$ between $C$ and $D$ in $C$ such that there exists a $G$-coalgebra structure $(R, \rho)$ on $R$ making $r_1$ and $r_2$ into $G$-coalgebra homomorphisms from $(R, \rho)$ to $(C, \gamma)$ and $(D, \delta)$ respectively. The largest $4$ $G$-bisimulation between $(C, \gamma)$ and $(D, \delta)$, if it exists, is called $G$-bisimilarity. Also, a $G$-subcoalgebra of $(C, \gamma)$ is an object $(\langle D, \delta \rangle, d)$ of the category $\mathsf{Coalg}(G)/\gamma^4$, such that the $C$-arrow underlying $d$ is a monomorphism, while a homomorphic image of $(C, \gamma)$ is an object $(\langle D, \delta \rangle, e)$ of the category $\mathsf{Coalg}(G)/\gamma^5$, 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 $G : C \to C$, 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 $(\mathsf{Epi}(C), \mathsf{Mono}(C))$ as a factorisation system$^7$, and with $G : C \to C$ an endofunctor which preserves pullbacks$^8$ and limits of $\omega^{2b}$-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].

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

$^6$Coalg($G$)/$\gamma$ has objects given by pairs $(\langle D, \delta \rangle, e)$ with $(D, \delta)$ a $G$-coalgebra and $e : (C, \gamma) \to (D, \delta)$ a $G$-coalgebra homomorphism, and arrows from $(\langle D, \delta \rangle, e)$ to $(\langle D', \delta' \rangle, e')$ given by coalgebra homomorphisms $f : (D, \delta) \to (D', \delta')$ 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 $(\mathsf{Epi}(C), \mathsf{Mono}(C))$-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 $(\mathcal{C}, \mathcal{G})$-coalgebras and $(\mathcal{C}, \mathcal{G})$-coalgebra homomorphisms is denoted $\text{Coalg}(\mathcal{C}, \mathcal{G})$, while the functor taking $(\mathcal{C}, \mathcal{G})$-coalgebras to their carrier is denoted $\mathcal{U}_C : \text{Coalg}(\mathcal{C}, \mathcal{G}) \to \mathcal{C}$.

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

$$
\begin{align*}
G_0 &= \text{id}_C \\
G_1 &= \text{id}_C \times GG_0 \\
G_2 &= \text{id}_C \times GG_1 \\
&\vdots 
\end{align*}
$$

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 $\mathcal{U}_C$ has a right adjoint.

**Remark 1.5.** Destructor cosignatures over a set $V$ of visible sorts are defined in [2] as pairs $(H, \Delta)$, with $H$ a set of hidden sorts and $\Delta$ 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 $\delta : h \to s_1 \ldots s_n$ for $\delta \in \Delta_{h, s_1 \ldots s_n}$.

For a destructor cosignature $(H, \Delta)$ over $V$, a $\Delta_D$-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 $\delta : h \to s_1 \ldots s_n$ in $\Delta$, a function $\delta_A : A_h \to A_{s_1} + \ldots + A_{s_n}$.

Also, a $\Delta_D$-homomorphism between $\Delta_D$-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 $[t_1 \circ f_{s_1}, \ldots, t_n \circ f_{s_n}](\delta_A(a)) = \delta_C(f_h(a))$ for each $\delta : h \to s_1 \ldots s_n$ in $\Delta$ and each $a \in A_h$, with $h \in H$ and $s_1, \ldots, s_n \in S$ (where $t_j : C_{s_j} \to C_{s_j} + \ldots + C_{s_n}$ for $j = 1, \ldots, n$ denote the coproduct injections). Then, the category $\text{Coalg}_D(\Delta)$ of $\Delta_D$-coalgebras and $\Delta_D$-homomorphisms is isomorphic to $\text{Coalg}(\text{Set}_D, G\Delta)$, with $\text{Set}_D^S$ 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 $G\Delta : \text{Set}_D^S \to \text{Set}_D^S$ being given by:

$$(G\Delta X)_S = \begin{cases} D_s & \text{if } s \in V \\
\prod_{a \in \Delta_{h, s_1 \ldots s_n}} (X_{s_1} + \ldots + X_{s_n}) & \text{if } s \in H, \ X \in [\text{Set}_D^S], \ s \in S. \end{cases}$$

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

$$(G_{ST} X)_{\text{Stack}} = (X_1 + X_{\text{Nat}}) \times (X_1 + X_{\text{Stack}}), \ X \in \text{Set}_D^{S_{ST}}.$$
The existence of a final object in $\mathcal{C}$ (see Def. 1.2) and of a right adjoint to $U_\mathcal{C}$ (see Rem. 1.4) result in the existence of final $(\mathcal{C}, G)$-coalgebras.

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

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

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

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

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

Propositions 1.7 and 1.9 now yield the following:

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

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

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

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

**Proposition 1.12.** Let $(\mathcal{C}, G)$ denote an abstract cosignature. Then, $U_\mathcal{C}$ 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 $U_\mathcal{C}$.

A consequence of $(\text{Epi}(\mathcal{C}), \text{Mono}(\mathcal{C}))$ defining a factorisation system for $\mathcal{C}$ and of the preservation of monomorphisms by $G$ is the existence in $\text{Coalg}(\mathcal{C}, G)$ 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 $f$. 

Definition 1.15. Let $(C, \gamma)$ denote an abstract cosignature. Also, let $f : \langle C, \gamma \rangle \to \langle D, \delta \rangle$ denote a $(C, G)$-coalgebra homomorphism, and let $U_C f = m \circ e$, with $e : C \to E$ and $m : E \to D$, denote the epi-mono factorisation of $U_C f$\(^9\). Then, there exists a unique $(C, G)$-coalgebra structure $(E, \eta)$ on $E$ which makes $e$ and $m$ into $(C, G)$-coalgebra homomorphisms.

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

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

Proof (sketch). By Proposition 1.13, $\text{Coalg}(C, G)$ 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 $\text{Coalg}(C, G)$.

Abstract cosignatures $(C, \gamma)$ induce natural transformations $\lambda : U_C \Rightarrow GU_C$, with $\lambda_{(C, \gamma)} : C \to GC$ being given by $\lambda_{(C, \gamma)} = \gamma$ for $(C, \gamma) \in |\text{Coalg}(C, G)|$. These natural transformations define basic observations on the carriers of $(C, G)$-coalgebras: $\lambda_{(C, \gamma)}$ uses the coalgebraic structure given by $\gamma$ 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 $G^{n-1}\lambda \circ \ldots \circ G\lambda \circ \lambda : U_C \Rightarrow G^nU_C$ with $n \in \mathbb{N}^\ast$. 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 $(C, \gamma)$ denote an abstract cosignature. A $(C, G)$-observer is a pair $(K, c)$, with $K : C \to C$ an endofunctor (called the type of the observer) which preserves monomorphisms, and with $c : U_C \Rightarrow KU_C$ a natural transformation.

$(C, G)$-observers are parameterised by $(C, G)$-coalgebras: a $(C, G)$-observer $(K, c)$ specifies, for each $(C, G)$-coalgebra $(C, \gamma)$, a $C$-arrow $c_{\gamma} : C \to KC$, extracting information of type $K$ from $C$ according to $\gamma$. 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 $c_\delta \circ f = Kf \circ c_\gamma$.

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

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

\*\*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 \(\langle C, \gamma \rangle \models (K, l, r)\)) if and only if \(l_\gamma = r_\gamma\).

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

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

1. \(Z \in T_\Delta[C]_s\) for \(Z \in C_s\), with \(s \in S\);
2. \([t_1, \ldots, t_n]\delta \in T_\Delta[C]_s\) for \(\delta \in \Delta_{s_1, \ldots, s_n}\) and \(t_i \in T_\Delta[C]_{s_i}\), \(i = 1, \ldots, n\), 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 in many-sorted algebra, can also be defined (see \([2]\)).

Given a \(\Delta_D\)-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 \(t_Z : A_s \rightarrow A_{s_1} + \ldots + A_{s_n}\) 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 \(\Delta\)-coterms \(t \in T_\Delta[C]\) in a \(\Delta_D\)-coalgebra \(A\), denoted \(t_A\), is defined inductively as follows:

1. \(Z_A = t_Z\) for \(Z \in C_s\), with \(s \in S\);
2. \(([t_1, \ldots, t_n]\delta)_A = ([t_1)_A, \ldots, (t_n)_A] \circ \delta_A\) for \(\delta \in \Delta_{s_1, \ldots, s_n}\) and \(t_i \in T_\Delta[C]_{s_i}\), \(i = 1, \ldots, n\), 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 coterms of the given sort, a covariable appearing in that coterms (specifying an evaluation path for the coterms), together with a value of the same sort as the covariable (specifying a result for the evaluation of the coterms), 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.

\(\Delta\)-coequations are used in \([2]\) to constrain the interpretations of \(\Delta\)-coterms by \(\Delta_D\)-coalgebras. A \(\Delta\)-coequation is given by a tuple \((l, r, C)\) (alternatively denoted \(l = r\) if \(C\)), with \(l, r \in T_\Delta[C]_h\) and \(C\) consisting of conditions \((t_1, C'_1), \ldots, (t_n, C'_n)\), with \(t_i \in T_\Delta[C]_h\) and \(C'_i \subseteq C_i\) for \(i = 1, \ldots, n\), for some \(h \in H\). A \(\Delta_D\)-coalgebra \(A\) satisfies a \(\Delta\)-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 \((t_i)_A(a) \in t_Z(A_{s_i})\) for some \(Z_i \in (C'_i)_h\), for \(i = 1, \ldots, n\) (case in which \(a\) is said to satisfy the conditions \(C\)).

\(\Delta\)-coequations are an instance of the abstract notion of coequation. For, given a \(\Delta\)-coterms \(t \in T_\Delta[C]_h\), with \(C\) a finite set of covariables, together with some conditions \(C\) for the sort \(h\), one can define a \((\text{Set}_D^S, G_\Delta)\)-observer \((K, l)\), with
\( K : \text{Set}_D^S \rightarrow \text{Set}_D^S \) being given by:

\[
(K_X)_s = \begin{cases} 
  D_s & \text{if } s \in V \\
  1 + \prod_{s' \in S} \prod_{Z \in \text{Coalg}_U} X_s' & \text{if } s = h \\
  1 & \text{if } s \in H \setminus \{h\} 
\end{cases}, \quad X \in |\text{Set}_D^S|, \ s \in S
\]

and with \( \bar{f} : U \Rightarrow KU \) (where \( U : \text{Coalg}(\text{Set}_D^S, G_\Delta) \rightarrow \text{Set}_D^S \) denotes the functor taking \((\text{Set}_D^S, G_\Delta)\)-coalgebras to their carrier) being given by:

\[
(\bar{f}_{(A, a)})_s(a) = \begin{cases} 
  a & \text{if } s \in V \\
  \nu_2(\nu_1(a)) & \text{if } s = h \text{ and } C \text{ holds in } a \\
  \nu_1(*) & \text{if } s = h \text{ and } C \text{ does not hold in } a \\
  * & \text{if } s \in H \setminus \{h\} 
\end{cases}, \quad a \in A_s, \ s \in S
\]

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

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

\[
(Z,S)_{\text{rest}} = [Z,S']_{\text{rest}} \text{ if } ([Z,N]_{\text{top}}, Z) \\
(Z,N)_{\text{top}} = [Z,N']_{\text{top}} \text{ if } ([Z,S]_{\text{rest}}, Z)
\]

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 \( \Delta \)-coequations by \( \Delta_D \)-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 \( f : \langle C, \gamma \rangle \rightarrow \langle D, \delta \rangle \) denote a \((C, G)\)-coalgebra homomorphism, and let \((K, l, r)\) denote a \((C, G)\)-coequation. Then, the following hold:

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

**Proof.** 1 follows from: \( \langle C, \gamma \rangle \models (K, l, r) \) \( \Leftrightarrow \ l_\gamma = r_\gamma \Rightarrow KU_C f \circ l_\gamma = KU_C f \circ r_\gamma \Leftrightarrow l_\delta \circ U_C f = r_\delta \circ U_C f \Leftrightarrow l_\delta = r_\delta \Leftrightarrow \langle D, \delta \rangle \models (K, l, r) \), while 2 follows from: \( \langle D, \delta \rangle \models (K, l, r) \) \( \Leftrightarrow l_\delta = r_\delta \Rightarrow l_\delta \circ U_C f = r_\delta \circ U_C f \Leftrightarrow KU_C f \circ l_\gamma = KU_C f \circ r_\gamma \Leftrightarrow l_\gamma = r_\gamma \Leftrightarrow \langle C, \gamma \rangle \models (K, l, r) \). (The fact that \( K \) preserves monomorphisms is used in proving 2.) \( \square \)
The requirement that $U_C f$ 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 $\text{Coalg}(C, G)$\footnote{The existence of colimits in $\text{Coalg}(C, G)$ follows from the existence of colimits in $C$ together with the creation of colimits by $U_C$.}. 

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

_Proof (sketch)._ The conclusion follows from: $l_\gamma \circ U_C f_D = KU_C f_D \circ l_{4D} = KU_C f_D \circ r_{4D} = r_\gamma \circ U_C f_D$ for $D \in [D]$, together with the preservation of colimits by $U_C$. \hfill $\Box$

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

_Proof._ The conclusion follows by Propositions 1.19 and 1.20. \hfill $\Box$

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

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

_Proof (sketch)._ Say $\varepsilon$ is of form $(K, l, r)$. Let $(D, \epsilon, \delta)$ denote the monad induced by $G$, let $\ell_\gamma, r_\gamma : \langle C, \gamma \rangle \to \langle DKC, \gamma' \rangle$ denote the unique (co)extensions of the $C$-arrows $l_\gamma, r_\gamma : C \to KC$ to $(C, G)$-coalgebra homomorphisms, and let $\iota : \langle S, \xi \rangle \to \langle C, \gamma \rangle$ denote an equaliser for $\ell_\gamma, r_\gamma$ (see Cor. 1.10). Then, $U_C \iota$ is a monomorphism\footnote{This follows directly from the preservation of monomorphisms by $U_C$. 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 $U_C$.}, and hence $\langle S, \xi, \iota \rangle$ defines a $(C, G)$-subcoalgebra of $\langle C, \gamma \rangle$. Moreover, $\langle S, \xi, \iota \rangle$ is final in $(\text{Coalg}(C, G)/\gamma)^\varepsilon$ – the fact that $(S, \xi) \models \varepsilon$ follows from standard properties of adjunctions together with the observation that $KU_C \iota$ is a monomorphism, while finality of $\langle S, \xi, \iota \rangle$ in $(\text{Coalg}(C, G)/\gamma)^\varepsilon$ follows from standard properties of adjunctions together with the couniversality of $\iota$. \hfill $\Box$

The existence of limits of $\omega^{sp}$-chains in $\text{Coalg}(C, G)$ (following from the existence of such limits in $C$ together with their preservation by $G$) now yields the following.
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, \((\text{Coalg}(C, G)/\gamma)^E\) 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 \(\langle C_{i+1}, \gamma_{i+1}, c_{i+1} \rangle\) denote a final object in \((\text{Coalg}(C, G)/\gamma_i)^E\). Finally, let \((\langle S, \xi \rangle, \langle \xi_i \rangle)_{i \in \mathbb{N}}\) define a limit for the \(\omega^0\)-chain defined by \(c_1, c_2, \ldots\)

\[
\begin{array}{c}
\langle S, \xi \rangle \\
\langle C_0, \gamma_0 \rangle \\
\langle C_1, \gamma_1 \rangle \\
\langle C_2, \gamma_2 \rangle \\
\vdots
\end{array}
\]

\[\xi_n \rightarrow_{i_n} \cdots \rightarrow_{i_1} \xi_0 \] The fact that \(U_C\) creates limits of \(\omega^0\)-chains together with the fact that each of \(U_{C1}, U_{C2}, \ldots\) are monomorphisms result in each of \(U_{Ct_0}, U_{Ct_1}, \ldots\) being monomorphisms. In particular, \(U_{Ct_0}\) is a monomorphism, and hence \((\langle S, \xi \rangle, \xi_0)\) defines a \((C, G)\)-subcoalgebra of \(\langle C, \gamma \rangle\). Moreover, \(\langle S, \xi \rangle \models E - \text{this follows by 2 of Proposition 1.19 from} (S, \xi)\) 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 \((\langle C_{i+1}, \gamma_{i+1}, c_{i+1} \rangle)\) is final in \((\text{Coalg}(C, G)/\gamma)^E\) for \(i \in \mathbb{N}\) results in \((\langle S, \xi \rangle, \xi_0)\) being final in \((\text{Coalg}(C, G)/\gamma)^E\).

Corollary 1.24. Let \((C, G)\) denote an abstract cosignature and let \(E\) denote an enumerable set of \((C, G)\)-coequations. Then, \(\text{Coalg}(C, G, E)\) 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 \(\langle C, \gamma \rangle \models^b (K, l, r)\)) if and only if \(\text{KU}_C! \circ l_\gamma = \text{KU}_C! \circ r_\gamma\), with \(!: \langle C, \gamma \rangle \rightarrow \langle F, \xi \rangle\) 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 \(\langle C, \gamma \rangle \models^b (K, l, r)\) holds precisely when \(\langle l_\gamma, r_\gamma \rangle\) factors through \((\text{Kr}_1, \text{Kr}_2)\):

\[
\begin{array}{ccc}
C & \xrightarrow{(l_\gamma, r_\gamma)} & \text{KC} \times \text{KC} \\
\downarrow \varepsilon & & \downarrow \text{Kr}_1, \text{Kr}_2 \\
\text{KR} & \xrightarrow{} & \text{KR}
\end{array}
\]
with \( (R, r_1, r_2) \) denoting \((C, G)\)-bisimilarity on \( (C, \gamma) \). For, in this case, \( Kr_1, Kr_2 \) define a kernel pair for \( KU_C \) (see also Rem. 1.11).

Standard satisfaction of \((K, l, r)\) by \( (C, \gamma) \) implies its satisfaction up to bisimulation by \( (C, \gamma) \). And if, in addition, \( (C, \gamma) \) is observable, the converse also holds. For, in this case, \( U_C \) 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 \( e : (C, \gamma) \rightarrow (E, \eta) \) and \( m : (E, \eta) \rightarrow (F, \zeta) \) denotes the \((U^{-1}_C(Epi(C)), U^{-1}_C(Mono(C)))\)-factorisation of the unique \((C, G)\)-coalgebra homomorphism from \( (C, \gamma) \) to the final \((C, G)\)-coalgebra, then \( (K, \gamma) \models^b (K, l, r) \) holds precisely when \( (E, \eta) \models (K, l, r) \) does. This follows from \( U_C e \) being an epimorphism together with \( KU_C m \) 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 \textit{generic bisimulations}, as defined below.

\textbf{Definition 1.28.} Let \((C, G)\) denote an abstract cosignature, and let \( C \) denote a full subcategory of \( \text{Coalg}(C, G) \). A \textbf{generic \((C, G)\)-bisimulation on \( C \)} is given by a tuple \( (R, \pi_1, \pi_2) \) with \( R : C \rightarrow C \) and \( \pi_1, \pi_2 : R \Rightarrow U_C | \_C \), such that \((R \gamma, \pi_1 \gamma, \pi_2 \gamma)\) defines a \((C, G)\)-bisimulation on \( (C, \gamma) \) for any \( (C, \gamma) \in \_C \).

That is, a generic bisimulation on \( C \) associates to each coalgebra in \( C \) 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 \( C \) of \( \text{Coalg}(C, G) \) can be reduced to exhibiting a generic \((C, G)\)-bisimulation \((R, \pi_1, \pi_2)\) on \( C \), such that \((l_\gamma, r_\gamma)\) factors through \((K \pi_1 \gamma, K \pi_2 \gamma)\) for any \( (C, \gamma) \in \_C \) (see also Rem. 1.27):

\[
\begin{array}{ccc}
C & \overset{(l_\gamma, r_\gamma)}{\rightarrow} & KC \times KC \\
c & \downarrow & \downarrow (K \pi_1 \gamma, K \pi_2 \gamma) \\
\text{KR}_\gamma & \rightarrow & \text{KR}_\gamma
\end{array}
\]

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

\textbf{Definition 1.29.} An \textbf{(abstract) cosignature morphism} between abstract cosignatures \((C, G)\) and \((C', G')\) is a pair \((U, \eta)\), with \( U : C' \rightarrow C \) a functor which
preserves limits and has a right adjoint $R^{13}$, and with $\eta : UG' \Rightarrow GU$ a natural transformation.

An abstract cosignature morphism $(U, \eta) : (C, G) \to (C', G')$ induces a reduct functor$^{14}$ $U_\eta : \text{Coalg}(C', G') \to \text{Coalg}(C, G)$, with $U_\eta$ taking a $(C', G')$-coalgebra $(D, \delta)$ to the $(C, G)$-coalgebra $(UD, \eta_D \circ U\delta)$, as well as a translation, itself denoted $\eta$, of $(C, G)$-observers to $(C', G')$-observers, with $\eta$ taking a $(C, G)$-observer $(K, c)$ to the $(C', G')$-observer $(RKU, c_{U_\eta})$:

\[
\begin{array}{c}
KU_\eta U = KUU_
\end{array}
\]

where:

\[
\begin{array}{ccc}
\text{Coalg}(C, G) & \xrightarrow{U} & \text{Coalg}(C', G') \\
\text{Coalg}(C, G) & \xrightarrow{U_\eta} & \text{Coalg}(C', G') \\
C & \xrightarrow{u} & C'
\end{array}
\]

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

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

**Proof.** If $e$ is of form $(K, l, r)$, then $U_\eta(D, \delta) \models e$ translates to $l_{U_\eta, \delta} = r_{U_\eta, \delta}$, while $(D, \delta) \models \eta(e)$ translates to $l_{\eta_\delta} = r_{\eta_\delta}$. The conclusion then follows by $R$ being a right adjoint to $U$. \[\square\]

**Remark 1.31.** A destructor cosignature morphism [2] between destructor cosignatures $(H, \Delta)$ and $(H', \Delta')$ over $V$ is given by a function $\phi : S \to S'$ satisfying $\phi \mid_V = 1_V$ 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].
\( \phi^+ \) 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 \( U_\phi : \text{Coalg}_{D'}(\Delta') \to \text{Coalg}_{D}(\Delta) \) (with \( U_\phi \) taking \( \Delta \text{D}_G \)-coalgebras \( C' \) to the \( \Delta \text{D}_G \)-coalgebras having carrier \( C'_\Delta = (C'_\phi(s))_{s \in S} \) and operations given by \( \delta_{\phi(s)} = \phi(\delta)s' \) for \( s' \in \Delta \)), as well as translations of \( \Delta \)-coequations to \( \Delta' \)-coequations. Then, \( U_\phi : \text{Coalg}_{D}(\Delta) \to \text{Coalg}_{D}(\Delta) \) is naturally isomorphic to \( U_\phi \), with the hidden-sorted components of the natural transformation \( \eta_\phi : U_G^{\Delta'} \Rightarrow G_\Delta U \) being given by:

\[
(\eta_\phi, x)_h((x_\delta')_{\delta' \in \Delta'_\phi(h)}) = (x_{\phi(h)}\delta)_{\delta \in \Delta_h}, \quad h \in H, \quad X \in |\text{Set}_D^{S'}|
\]

and with \( U : \text{Set}_D^{S'} \to \text{Set}_D^{S} \) denoting the functor taking \( A \in |\text{Set}_D^{S'}| \) to \( (A_{\phi(s)})_{s \in S} \in |\text{Set}_D^{S}| \). Moreover, the translation along \( \phi \) of \( \Delta \)-coequations agrees with the translation along \( \eta_\phi \) of the induced \( G_\Delta \)-coequations.

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

\[
\begin{array}{c}
G_0 U \xrightarrow{(g_0)_U} G_1 U \xrightarrow{(g_1)_U} G_2 U \xrightarrow{(g_2)_U} \ldots
\end{array}
\]

whereas \( UD' \) is the object of a cone for this \( \omega^{op} \)-chain. Moreover, if \( \eta \) is a natural monomorphism, then so is \( \rho \). (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 \( I' \) denotes a final \( C \)-object, then the \( I' \)-component of the natural transformation \( \rho \) defining the comonad morphism gives the unique homomorphism from the \( (C, G) \)-reduct of the final \( (C', G') \)-coalgebra to the final \( (C, G) \)-coalgebra. Hence, the fact that the \( C \)-arrow \( \rho_I' : UD' I' \to DU I' \) 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\(^\text{16}\).

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 \( (U, \rho) \) with \( U : C' \to C \) and \( \rho : UD' \Rightarrow DU \) (subject to suitable constraints), rather than by a natural transformation \( \rho : D' \Rightarrow D \). See [4] (2.4.15) for details.

\(^{16}\)Informally, states in the final \( (C', G') \)-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 \((U, \eta) : (C, G) \rightarrow (C', G')\) is **horizontal** if and only if \(\eta\) is a natural monomorphism.

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

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

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

**Remark 1.36.** If the destructor cosignature morphism \(\phi : (H, \Delta) \rightarrow (H', \Delta')\) 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 \(\phi\) (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\(^\text{17}\) suffices. On the other hand, the requirement regarding the \(\omega^{op}\)-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

\(^\text{17}\)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\textsuperscript{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\textsuperscript{-}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 (\(\text{Epi}(C), \text{Mono}(C)\)) as a factorisation system, and with \(F : C \to C\) an endofunctor which preserves epimorphisms and colimits of \(\omega\)-chains, and is such that the category \(\text{Alg}(F)\) has coequalisers\textsuperscript{19}. An (abstract) **signature morphism** between abstract signatures (\(C, F\)) and (\(C', F'\)) is a pair

\textsuperscript{18}This result is only needed in the dual case (namely for the proof of Prop. 3.7).

\textsuperscript{19}The existence of coequalisers in \(\text{Alg}(C, F)\) is only needed for the formulation of results dual to Propositions 1.22 and 1.23.
(U, ξ), with U: C′ → C a functor which preserves colimits and has a left adjoint L, and with ξ: FU ⇒ 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 Alg(C, F), while the functor taking (C, F)-algebras to their carrier is denoted UC: Alg(C, F) → C.

**Remark 2.3.** Similarly to Remark 1.14, abstract signatures (C, F) induce monads (T, η, μ) on C such that Alg(C, F) ≃ Alg(T). Moreover, since F preserves epimorphisms, so does T. Also, similarly to Remark 1.32, abstract signature morphisms (U, ξ) : (C, F) → (C′, F′) induce monad morphisms (U, ν) : (T, η, μ) → (T′, η′, μ′). And if, in addition, ξ is a natural epimorphism, then so is ν. (The fact that the C-arrow i!hA;idl;r = (TU)l;r, with ! denoting an initial C′-object, is an epimorphism formalises the fact that the notion of reachability associated to (C, F) does not refine the notion of reachability associated to (C, F).)

The existence of an initial object in C and of a left adjoint to UC result in the existence of an initial object in Alg(C, 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 ⟨A, α⟩ is reachable if and only if the C-arrow underlying the unique F-algebra homomorphism from the initial (C, F)-algebra to ⟨A, α⟩ is an epimorphism.

Also, for an abstract signature (C, F), Alg(C, F) has factorisations of form f = m ◦ e, with e defining a homomorphic image and with m defining a subalgebra. Moreover, (UC′(Mono(C)), UC′(Epi(C))) defines a factorisation system for Alg(C, 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 K: C → C a functor which preserves epimorphisms, and with c: KUC ⇒ UC 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 ⟨A, α⟩ satisfies a (C, F)-equation (K, l, r) (written ⟨A, α⟩ |= (K, l, r)) if and only if lα = rα. Also, ⟨A, α⟩ satisfies a (C, F)-equation (K, l, r) up to reachability (written ⟨A, α⟩ |=∗ (K, l, r)) if and only if lα ◦ KUC! = rα ◦ KUC!, with !: ⟨I, ξ⟩ → ⟨A, α⟩ denoting the unique (C, F)-algebra homomorphism from the initial (C, F)-algebra to ⟨A, α⟩.

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 (C′, 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 \((S, \Sigma)\) denotes a many-sorted signature and \(F_\Sigma : \text{Set}^S \to \text{Set}^S\) denotes the endofunctor given by:

\[
(F_\Sigma X)_s = \prod_{s \in S} (X_{s_1} \times \ldots \times X_{s_n}), \; X \in |\text{Set}^S|, \; s \in S
\]

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

\[
(KX)_s = \begin{cases} 
X_{s_1} \times \ldots \times X_{s_m} & \text{if } s' = s \\
\emptyset & \text{otherwise}
\end{cases}
\]

and with \(\ell : KU \Rightarrow U\) (where \(U : \text{Alg}(\text{Set}^S, F_\Sigma) \to \text{Set}^S\) denotes the functor taking \((\text{Set}^S, F_\Sigma)\)-algebras to their carrier) being given by:

\[
(\ell(A, \alpha))_{s'} = \begin{cases} 
tA & \text{if } s' = s \\
!: \emptyset \Rightarrow A' & \text{otherwise}
\end{cases}
\]

for each \((\text{Set}^S, F_\Sigma)\)-algebra \((A, \alpha)\) with \(A\) its associated \(\Sigma\)-algebra. Finally, unconditional \(\Sigma\)-equations \(e\) of form \((\forall V) \; l = r\) induce \((\text{Set}^S, F_\Sigma)\)-equations \((K, \ell, \rho)\), with \(K : \text{Set}^S \to \text{Set}^S\) being as before, and with \(\ell, \rho : KU \Rightarrow U\) being the \((\text{Set}^S, F_\Sigma)\)-constructors associated to \(l\) and \(r\). Moreover, \(A \models_\Sigma e\) is equivalent to \((A, \alpha) \models (K, \ell, \rho)\).

**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 \(S_{\text{ST}}\) over visible sorts \(1^2\) and \(\text{Nat}\), consisting of a hidden sort \(\text{Stack}\) together with operation symbols \(\text{empty} : \to \text{Stack}\), \(\text{push} : \text{Stack} \times \text{Nat} \to \text{Stack}\) and \(\text{pop} : \text{Stack} \to \text{Stack}\). The abstract signature associated to this constructor signature is \((\text{Set}^S_{\text{ST}}, F_{\text{ST}})\), with \(S_{\text{ST}}\) and \(D\) being as in Example 1.6, and with the hidden-sorted component of \(F_{\text{ST}}\) being given by:

\[
(F_{\text{ST}}X)_{\text{Stack}} = X_1 + (X_{\text{Stack}} \times X_{\text{Nat}}) + X_{\text{Stack}}, \; X \in \text{Set}^D_{\text{ST}}.
\]

\(^{22}\) For consistency with Example 1.6.
The hidden-sorted carrier of the initial algebra of the stack signature is isomorphic to the set of $\Sigma_{\text{ST}}$-terms over $D$, of sort $\text{Stack}$.

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

\[
pop(\text{empty}) = \text{empty} \\
pop(\text{push}(s,n)) = s.
\]

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

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 $\sigma : TUC \Rightarrow GTUC$, with $U_C : \text{Coalg}(C,G) \rightarrow C$ denoting the functor taking $(C,G)$-coalgebras to their carrier, and with $(T,\eta,\mu)$ denoting the monad induced by the abstract signature $(C,F)$, to deﬁne liftings of the monad $T$ to the category $\text{Coalg}(C,G)$.

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

\[
\begin{array}{ccc}
T^2UC & \xrightarrow{\sigma T^C} & GT^2UC \\
\downarrow{\mu_{UC}} & & \downarrow{G\mu_{UC}} \\
TUC & \xrightarrow{\sigma} & GTUC \\
\uparrow{\eta_{UC}} & & \uparrow{G\eta_{UC}} \\
U_C & \xrightarrow{\lambda} & GU_C
\end{array}
\]

where the natural transformation $\lambda : U_C \Rightarrow GU_C$ is given by: $\lambda_{(C,G)} = \gamma$ for $(C,\gamma) \in \text{[Coalg(G)]}$, while the functor $T_\sigma : \text{Coalg}(G) \rightarrow \text{Coalg}(G)$ is given by:

- $T_\sigma(C,\gamma) = (TC,\sigma_\gamma)$ for $(C,\gamma) \in \text{[Coalg(G)]}$
- $U_CT_\sigma f = TU Cf$ for $f \in \text{[Coalg(G)]}$

(and consequently $U_CT_\sigma = TU_C$).

An (abstract) lifted signature morphism from $(C,G,F,\sigma)$ to $(C',G',F',\sigma')$ is a tuple $(U,\tau,\xi)$ with $(U,\tau) : (C,G) \rightarrow (C',G')$ an abstract cosignature morphism.

\[\text{Naturality of } \sigma \text{ ensures that } TU_C f \text{ defines a } G\text{-coalgebra homomorphism.}\]
and \((U, \xi) : (C, F) \rightarrow (C', F')\) a horizontal abstract signature morphism, such that the following diagram commutes:

\[
\begin{array}{ccc}
TU_U \times & \overset{\sigma_{U'}}{\longrightarrow} & GTU_U \\
\downarrow & & \downarrow \\
TU_U & \overset{\sigma_{U}}{\longrightarrow} & GTU_U \\
\downarrow & & \downarrow \\
UT'U & \overset{U\sigma'}{\longrightarrow} & UG'T'U \\
\end{array}
\]

where \((U, \nu) : (T, \eta, \mu) \rightarrow (T', \eta', \mu')\) is the monad morphism induced by the signature morphism \((U, \xi)\).

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

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

The constraints defining a lifted signature morphism \((U, \tau, \xi) : (C, G, F, \sigma) \rightarrow (C', G', F', \sigma')\) ensure that, for any \((C', G')\)-coalgebra \((C'', \gamma')\), the \((C', G')\)-coalgebra structure induced by \(\sigma'\) on \(T'C''\) agrees with the \((C, G)\)-coalgebra structure induced by \(\sigma\) on \(TUC\). This results in lifted signature morphisms \((U, \tau, \xi) : (C, G, F, \sigma) \rightarrow (C', G', F', \sigma')\) inducing reduct functors \(U_{(\tau, \xi)} : \text{Alg}(T_\sigma') \rightarrow \text{Alg}(T_\sigma)\), with \(U_{(\tau, \xi)}\) taking a \(T_\sigma'\)-algebra \((C'', \gamma', \alpha')\) to the \(T_\sigma\)-algebra \((\{UC'', \tau C'' \circ U\gamma', \alpha'' \circ \nu C''\})\).

**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 \(\lambda : TD \Rightarrow DT\) 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 \(\lambda : TD \Rightarrow DT\) are in one-to-one correspondence with liftings \(T\) of \(T\) to \(D\)-coalgebras, and moreover, the category
Alg(T) is isomorphic to the category of (T, D)-bialgebras satisfying λ. Thus, the models of a lifted signature (C, G, F, σ) 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 σ : TUC ⇒ G(UC + FUC) required by the definition of lifted signatures can be given in terms of a natural transformation ρ : FUC ⇒ G(UC + FUC) (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 (C, G, F, σ) and (C', G', F', σ') are the lifted signatures induced by the natural transformations ρ : FUC ⇒ G(UC + FUC) and respectively ρ' : F'U'C ⇒ G'(UC' + F'U'C), and if (U, τ) : (C, G) → (C', G') is a cosignature morphism and (U, ξ) : (C, F) → (C', F') is a horizontal signature morphism, such that the following diagram commutes:

\[
\begin{array}{ccc}
FUC & \xrightarrow{ρ} & G(UC + FUC) \\
\xi_U & & \xi_G \\
UC & \xrightarrow{Uτ} & UC + FUC \\
\end{array}
\]

then (U, τ, ξ) defines a lifted signature morphism from (C, G, F, σ) to (C', G', F', σ'). (The above constraint on ρ, ρ', ξ and τ ensures that σ, σ', ν and τ satisfy the requirements of 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 STACK = (Set\(^{Set}_D\), G\(_{ST}\), F\(_{ST}\), σ), with \(S\)\(_{ST}\), \(D\)\(_{ST}\) and \(F\)\(_{ST}\) being as in Examples 1.6 and respectively 2.7, and with the natural transformation σ : T\(_{ST}\)U ⇒ G\(_{ST}\)T\(_{ST}\)U (where T\(_{ST}\) denotes the monad induced by \(F\)\(_{ST}\), and where U : Coalg(Set\(^{Set}_D\), G\(_{ST}\)) → Set\(^{Set}_D\) denotes the functor taking (Set\(^{Set}_D\), G\(_{ST}\))-coalgebras to their carrier) being induced by the natural transformation ρ : F\(_{ST}\)U ⇒ G\(_{ST}\)(U + F\(_{ST}\)U) whose hidden-sorted component is given by:

\[
\begin{align*}
(r(C, γ))_{\text{Stack}}(t_1(*)) &= \langle l_1(*D), l_1(*D) \rangle \\
(r(C, γ))_{\text{Stack}}(t_2(c, d)) &= \langle l_2(d), l_2(c) \rangle \\
π_1((r(C, γ))_{\text{Stack}}(t_3(c))) &= \begin{cases} 
 t_1(*D) & \text{if } [t_1, \text{topc}][\text{restc}(c)] \in t_1(C_1) \\
 t_2(d) & \text{if } [t_1, \text{topc}][\text{restc}(c)] = t_2(d) \in t_2(C_{\text{Nat}})
\end{cases} \\
π_2((r(C, γ))_{\text{Stack}}(t_3(c))) &= \begin{cases} 
 t_1(*D) & \text{if } [t_1, \text{restc}][\text{restc}(c)] \in t_1(C_1) \\
 t_2(c') & \text{if } [t_1, \text{restc}][\text{restc}(c)] = t_2(c') \in t_2(C_{\text{Stack}})
\end{cases}
\]

\(^{24}\)This observation is based on the approach in [18].
for \((C, \gamma) \in \text{Coalg}(\text{Set}_{\text{SST}}^{\setminus}, \text{G}_{\text{SST}})\) with \(C\) its associated \(\Delta_{\text{SST}}\)-coalgebra. The natural transformation \(\rho\) can alternatively be specified using the following constraints\(^{25}\):

\[
\begin{align*}
\text{[Z,N]\text{top}\text{.empty}} &= \text{Z.}\ast \\
\text{[Z,S]\text{rest}\text{.empty}} &= \text{Z.}\ast \\
\text{[Z,N]\text{top}\text{.push(s,n)}} &= \text{N.n} \\
\text{[Z,S]\text{rest}\text{.push(s,n)}} &= \text{S.s} \\
\text{[Z,N]\text{top}\text{.pop(s)}} &= \text{Z.}\ast \text{ if [Z,[Z,N]\text{top}\text{]rest}\text{.s}} = \text{Z.z} \\
\text{[Z,N]\text{top}\text{.pop(s)}} &= \text{N.n} \text{ if [Z,[Z,N]\text{top}\text{]rest}\text{.s}} = \text{N.n} \\
\text{[Z,S]\text{rest}\text{.pop(s)}} &= \text{Z.}\ast \text{ if [Z,[Z,S]\text{rest}\text{]rest}\text{.s}} = \text{Z.z} \\
\text{[Z,S]\text{rest}\text{.pop(s)}} &= \text{S.s'} \text{ if [Z,[Z,S]\text{rest}\text{]rest}\text{.s}} = \text{S.s'}
\end{align*}
\]

defining the results of \(\Delta_{\text{SST}}\)-observers on \(\Sigma_{\text{SST}}\)-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 \(\text{top}\) on the result yielded by the constructor \(\text{pop}\), 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 \([Z,N]\text{top}\) yields a result of type \(1\), whereas in the second case the resulting stack contains at least one element, and hence observing it using \([Z,N]\text{top}\) yields the same result (of type \(\text{Nat}\)) as when observing the original stack using \([Z,[Z,N]\text{top}\text{]rest}\). This is captured by the use of the variable \(n\) both in the condition of the second constraint defining the value of \(\text{top}\) on the result yielded by \(\text{pop}\) (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 \((C, G, F, \sigma)\), the existence of finite limits in \(\text{Coalg}(C, G)\) (see Cor. 1.10) results in the existence of finite limits in \(\text{Alg}(T_\sigma)\).

**Proposition 3.5.** Let \((C, G, F, \sigma)\) denote a lifted signature. Then, \(\text{Alg}(T_\sigma)\) has finite limits and \(U_{\text{Coalg}(C,G)}\) preserves them.

**Proof.** The conclusion follows from Corollary 1.10 together with the observation that \(U_{\text{Coalg}(C,G)}\) creates limits. \(\square\)

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

Also, kernel pairs exist in \(\text{Alg}(T_\sigma)\) and are created by \(U_{\text{Coalg}(C,G)}\). This yields a \(T_\sigma\)-algebra structure on \((C, G)\)-bisimilarity on the underlying coalgebra of a \(T_\sigma\)-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 $T_{\sigma}$-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 $T_{\sigma}$-algebra is preserved by the $T$-algebra structure.

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

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

The initial object is given by the $T_{\sigma}$-algebra $\langle T_{\sigma}(0,1_{G0}),\mu_0 \rangle = \langle \langle T0,\sigma_{\mu0} \rangle,\mu_0 \rangle$, where $0$ denotes an initial $C$-object (and consequently $1_{G0} : 0 \rightarrow G0$ defines an initial $(C,G)$-coalgebra). This $T_{\sigma}$-algebra provides an observational structure on ground computations.

Now recall from Corollary 1.14 that $\langle U_C^{-1}(Epi(C)),U_C^{-1}(Mono(C)) \rangle$ defines a factorisation system for $Coalg(C,G)$. 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 $T_{\sigma}$ also preserves epimorphisms. These observations result in the existence in $Alg(T_{\sigma})$ of factorisations of form $f = m \circ e$, with $e$ defining a $T_{\sigma}$-homomorphic image and with $m$ defining a $T_{\sigma}$-subalgebra.

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

**Proof.** Similar to the proof of Proposition 1.13.

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

**Proof.** Similar to the proof of Corollary 1.14.

By taking the $T_{\sigma}$-algebra homomorphism $f$ in the statement of Proposition 3.7 to be the unique homomorphism from the initial $T_{\sigma}$-algebra to an arbitrary one, one obtains, for each $T_{\sigma}$-algebra $\langle \langle C,\gamma \rangle,\alpha \rangle$, a $T_{\sigma}$-subalgebra of $\langle \langle C,\gamma \rangle,\alpha \rangle$ which is reachable. This $T_{\sigma}$-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 \((C, G, F, \sigma)\) denote a lifted signature. A \(T_\sigma\)-algebra \((\langle C, \gamma \rangle, \alpha)\) satisfies a \((C, G)\)-coequation \((K, l, r)\) up to reachability (written \(\langle C, \gamma \rangle, \alpha \models^l (K, l, r)\)) if and only if \(l_0 \circ U_C U_{Coalg(C,G)} l = r_0 \circ U_C U_{Coalg(C,G)}!\), with \(!\) denoting the unique \(T_\sigma\)-algebra homomorphism from the initial \(T_\sigma\)-algebra to \((\langle C, \gamma \rangle, \alpha)\).

Also, \(\langle C, \gamma \rangle, \alpha\) satisfies a \((C, F)\)-equation \((K', l', r')\) up to bisimulation (written \(\langle C, \gamma \rangle, \alpha \models^b (K', l', r')\)) if and only if \(U_C U_{Coalg(C,G)} l' \circ l'_0 = U_C U_{Coalg(C,G)} r' \circ r'_0\), with \(!\) denoting the unique \(T_\sigma\)-algebra homomorphism from \((\langle C, \gamma \rangle, \alpha)\) to the final \(T_\sigma\)-algebra, and with \((C, \alpha')\) denoting the \((C, F)\)-algebra induced by the \(T\)-algebra \((C, \alpha)\).

Standard satisfaction of coequations (respectively equations) by the underlying coalgebra (algebra) of a \(T_\sigma\)-algebra implies their satisfaction up to reachability (up to bisimulation) by the \(T_\sigma\)-algebra in question. That is, \(\langle C, \gamma \rangle \models (K, l, r)\) implies \(\langle C, \gamma \rangle, \alpha \models^l (K, l, r)\), while \(\langle C, \alpha' \rangle \models (K', l', r')\) implies \(\langle C, \gamma \rangle, \alpha \models^b (K', l', r')\).

Moreover, if the underlying algebra (coalgebra) is reachable (observable), then the converse also holds. For, in this case, \(U_C U_{Coalg(C,G)}!\) (respectively \(U_C U_{Coalg(C,G)}!\)) is an epimorphism (monomorphism).

Since \(U_C\) preserves kernel pairs, it follows that \(\langle C, \gamma \rangle, \alpha \models^b (K', l', r')\) holds if and only if \((l'_0, r'_0)\) factors through \((r_1, r_2)\):

\[
\begin{array}{ccc}
KC & \xrightarrow{(l'_0, r'_0)} & C \times C \\
\downarrow c & & \downarrow (r_1, r_2) \\
R & \xrightarrow{R} & \end{array}
\]

with \((R, r_1, r_2)\) 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 \(A\) of \(Alg(T_\sigma)\) can be reduced to exhibiting a generic \((C, G)\)-bisimulation\(^{26}\) \((R, \pi_1, \pi_2)\) on \(U_{Coalg(C,G)}(A)\), such that \((l'_0, r'_0)\) factors through \(\pi_{1, \gamma}, \pi_{2, \gamma}\) for any \(\langle C, \gamma \rangle, \alpha \in |A|\).

\[
\begin{array}{ccc}
KC & \xrightarrow{(l'_0, r'_0)} & C \times C \\
\downarrow c & & \downarrow \pi_{1, \gamma}, \pi_{2, \gamma} \\
R \gamma & \xrightarrow{R} & \end{array}
\]

For this, in turn, yields a \(C\)-arrow \(d : KC \to R\gamma\) such that \(\langle \pi_{1, \gamma}, \pi_{2, \gamma} \rangle \circ d = (l'_0, r'_0)\). Also, the maximality of \((C, G)\)-bisimilarity on \(\langle C, \gamma \rangle\) yields a \(C\)-arrow \(e : R\gamma \to R\).

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 \( C = (\langle C, \gamma \rangle, \alpha) \), 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 \( C_{\text{Stack}} \) satisfying:

\[
\begin{align*}
  & c R_{C, \text{Stack}} c, \text{ for each } c \in C_{\text{Stack}}; \\
  & \text{pop}(\text{empty}) R_{C, \text{Stack}} \text{ empty}; \\
  & \text{pop}(\text{push}(c, d)) R_{C, \text{Stack}} c, \text{ for each } c \in C_{\text{Stack}} \text{ and each } d \in C_{\text{Nat}}.
\end{align*}
\]

The relations \( R_C \) define a generic bisimulation on the coalgebras underlying STACK-algebras – this follows from the definition of \( \rho \) 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 \( C \models^b \text{push}(\text{empty}) = \text{empty} \) and \( C \models^b \text{push}(\text{empty}, n) = s \) 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 \( \text{empty} \), \( \text{push} \) and \( \text{pop} \). For, the fact that these coequations hold in a state \( s \) does not guarantee that they hold in \( \text{pop}(s) \), and therefore straightforward induction cannot be used to show that the coequations hold in all reachable states. However, the observation that the equations: \( \text{pop}(\text{empty}) = \text{empty} \) and \( \text{pop}(\text{push}(s, n)) = s \) 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 \( \text{empty} \), \( \text{push} \) and \( \text{pop} \) to proving that the stack invariant holds up to reachability under \( \text{empty} \) and \( \text{push} \) only. For, from the satisfaction up to bisimulation of the above equations, one can infer that any stack state reachable under \( \text{empty} \), \( \text{push} \) and \( \text{pop} \) is bisimilar to a stack state reachable under \( \text{empty} \) and \( \text{push} \) only. Then, the satisfaction of the stack invariant follows from this invariant holding in \( \text{empty} \) and being preserved by \( \text{push} \), 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 \( T_\sigma \)-algebras.

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

1. A \((C, G)\)-coequation is satisfied (up to reachability) by the initial \( T_\sigma \)-algebra precisely when it is satisfied up to reachability by any \( T_\sigma \)-algebra;
2. A \((C, F)\)-equation is satisfied (up to bisimulation) by the final \( T_\sigma \)-algebra precisely when it is satisfied up to bisimulation by any \( T_\sigma \)-algebra.
Proof. The conclusion follows immediately using the naturality of \((C, G)\)-observers and respectively of \((C, F)\)-constructors.

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 \(T_\sigma\)-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 \((C, G, F, \sigma)\) denote a lifted signature, and let ! = \(m \circ e\), with \(e: \langle(T0, \sigma_\text{in}), \mu_0 \rangle \to \langle(R, \gamma), \alpha \rangle\) and \(m: \langle(R, \gamma), \alpha \rangle \to \langle(F, \zeta), !_\sigma \rangle\), denote the factorisation of the unique \(T_\sigma\)-algebra homomorphism from the initial \(T_\sigma\)-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 \(\langle(F, \zeta), !_\sigma \rangle \models^t (K, l, r)\) is equivalent to \(\langle(R, \gamma), \alpha \rangle \models^t (K, l, r)\), as well as to \(\langle(R, \gamma) \models (K, l, r)\);
2. If \((K', l', r')\) denotes a \((C, F)\)-equation, then \(\langle(T0, \sigma_\text{in}), \mu_0 \rangle \models^b (K', l', r')\) is equivalent to \(\langle(R, \gamma), \alpha \rangle \models^b (K', l', r')\), as well as to \(\langle(R, \gamma) \models (K', l', r')\).

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

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

Proof. Let !: \(T0 \to UC\) and \(!': T0' \to C\) denote the \(C\)- and respectively \(C'\)-arrows underlying the unique \(T_\sigma\)- and \(T'_\sigma\)-algebra homomorphisms from the initial

\(^{27}\)Proposition 1.12 is also used here.
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


Accepted April 25, 2001.