Semantic Constructions for the Specification of Objects

Corina Cîrstea*

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Corina.Cirstea@comlab.ox.ac.uk

Abstract. Hidden algebra is a behavioural algebraic specification formalism for objects. It captures their constructional aspect (concerned with the initialisation and evolution of their states), their observational aspect (concerned with the observable behaviour of such states), and the relationship between these two aspects. When attention is restricted to the observational aspect, final/cofree algebras provide suitable denotations for the specification techniques employed by hidden algebra. However, when the constructional aspect is integrated with the observational one, the possibility of underspecification prevents the existence of such algebras. It is shown here that *final/cofree families* of algebras exist in this case, with each algebra in such a family resolving the nondeterminism arising from underspecification in a particular way. The existence of final/cofree families also yields a canonical way of constructing algebras of structured specifications from algebras of their component specifications.

1 Introduction

The use of algebra in the semantics of computation goes back to the 1970s and the use of initial algebras as denotational semantics for data types [GTW78]. The constructional nature of data types makes algebra particularly suitable for their specification – the emphasis is on *generating* the elements of data types by means of *constructor operations*, with minimal structures such as initial or free algebras providing suitable denotations for data type specifications. Recently, the theory of *coalgebras* (the formal duals of algebras) has been used for the specification of state-based systems in general [Rut96], and of objects in particular [Jac96]; here, the emphasis is on *observing* system states by means of *destructor operations*, with maximal structures such as final or cofree coalgebras, incorporating all possible behaviours, being used as denotations.

Objects are characterised by a state together with an interface providing limited access to this (otherwise hidden) state. Specifically, the object interface can be used to initialise the object state, to perform certain changes on the current state, or to observe certain properties of this state. One can identify a constructional aspect of objects, concerned with the initialisation and evolution of their

* Research supported by an ORS Award and an Oxford Bursary.

states, and an observational aspect of objects, concerned with the observable behaviour of such states.

The *hidden algebra* formalism ([Gog91], [GD94], [MG94], [GM97]) combines concepts from algebra and coalgebra in order to capture these two aspects and the relationship between them. One can argue that hidden algebra lies at the intersection of algebra and coalgebra, as its syntax is (a restricted version of) the syntax of many-sorted algebra, while its semantics is observational (coalgebraic). (Consequently, the behaviours specifiable in hidden algebra are, in a sense, both algebraic and coalgebraic.)

The coalgebraic nature of hidden algebra, already observed in [GM97], has been further investigated in [Cîr98], where the relevance of final/cofree constructions to *destructor* hidden specifications and their reuse along specification maps has been emphasised. Final hidden algebras have been shown to provide a characterisation of the abstract behaviours associated to a destructor hidden specification, while cofree hidden algebras have been used as formal denotations for the reuse of such specifications.

When arbitrary hidden specifications are considered, the nondeterminism arising from underspecifying the behaviour of the constructor operations prevents the existence of final/cofree hidden algebras. It has been suggested in [Cîr98] that, in this case, *final/cofree families* of hidden algebras should be used as denotations, since these constructions are able to characterise all possible ways of resolving the nondeterminism involved. Final/cofree families generalise final/cofree objects in a category, while still retaining their universal properties. This paper gives a detailed account of the existence of such families in hidden algebra, illustrating their suitability as semantic constructions for the specification of objects. The existence of final/cofree families of hidden algebras also yields a canonical way of constructing algebras of structured specifications from algebras of their component specifications.

The paper is structured as follows. After recalling some category-theoretic concepts to be used later in the paper, Section 2 introduces the hidden algebra formalism and briefly summarises the results in [Cîr98] regarding the existence of final/cofree constructions in a restricted version of hidden algebra. Section 3 then focuses on final/cofree families of hidden algebras, proving their existence and emphasising their suitability as denotations for hidden specifications and their reuse along specification maps. Section 4 uses a generalisation of the category-theoretic notion of limit [Die79] to define a canonical way of combining algebras of component specifications into algebras of structured specifications. This also yields a compositional semantics for structured hidden specifications. Section 5 discusses the relation between hidden algebra and other existing approaches to system specification, based either exclusively on coalgebra or on a combination of algebra and coalgebra. Finally, Section 6 summarises the results presented and briefly outlines future work.

2 Preliminaries

The first part of this section introduces some categorical concepts that will be used later in the paper, including a generalisation of the notions of *final/cofree object* (Section 2.1) and the concept of *fibration* (Section 2.2), while the second part (Section 2.3) gives an outline of the hidden algebraic approach to object specification and of some existing results regarding the existence of final/cofree constructions in a restricted version of hidden algebra.

2.1 Final and Cofree Families of Objects

A *final object* in a category is an object into which any other object of the category has a unique arrow. Final objects do not exist in any category. Practical examples have, however, suggested a generalisation of the notion of final object which exists in situations where a final object does not. This generalisation involves partitioning the category into subcategories with final objects. The notion of *cofree object* has also been generalised in a similar way. The generalisations are due to Diers [Die79] and will be briefly recalled in the following. In addition, we show how these generalisations can be subsumed under the standard concepts.

The notion of *final family of objects* generalises the notion of final object by requiring the existence of a unique arrow from any object of the category into an object in the final family.

Definition 1. *Given a category* C*, a family* $(F_j)_{j \in J}$ *of* C*-objects is a* **final family of** C**-objects** *if and only if, for any* C*-object* C*, there exist unique* $j \in J$ *and* C*-arrow* $f : C \to F_j$ *in* C*.*

Remark 2. A final family $(F_j)_{j \in J}$ of C-objects determines a partition $(\mathsf{C}_j)_{j \in J}$ of C into subcategories, each of them having a final object (given by an object in the final family). For $j \in J$, C_j is isomorphic to the slice category C/F_j. (The fact that the slices over the final family determine a partition of C is a consequence of the universal property of the final family.)

[Die79] presents a generalisation of the category-theoretic notion of limit, called a *multi-limit*.

Definition 3. *Given a diagram* $\mathsf{d} : \mathsf{D} \to \mathsf{C}$ *in a category* C*, a* **multi-limit for** d *consists of a family* $(L^i, (l^i_D : L^i \to \mathsf{d}(D))_{D \in |\mathsf{D}|})_{i \in I}$ *of cones for* d*, having the property that given any other cone* $(C, (c_D)_{D \in |\mathsf{D}|})$ *for* d*, there exist unique* $i \in I$ *and* C*-arrow* $c : C \to L^i$ *such that* $l^i_D \circ c = c_D$ *for each* D*-object* D*.*

Final families of objects now appear as a particular case of multi-limits, namely as multi-limits of empty diagrams.

It is shown in [Die79] that the standard results regarding the existence of finite limits (see e.g. [Bor94]) generalise to multi-limits. In particular, the existence of finite multi-limits in a category is a consequence of the existence of multi-products and of multi-equalisers. The following result can be proved in a similar way.

Theorem 4. *If a category* C *has a final family of objects and multi-pullbacks, then* C *is finitely multi-complete.*

Remark 5. The concept of multi-limit can be subsumed under the ordinary concept of limit by considering *categories of families*. Given a category C, one can define a category $\mathsf{Fam}(\mathsf{C})$ whose objects are indexed families $(C_i)_{i\in I}$ of C-objects and whose arrows from $(C_i)_{i\in I}$ to $(D_j)_{j\in J}$ are given by a (reindexing) function $h : I \to J$ together with an I-indexed family $(f_i)_{i\in I}$ of C-arrows, with $f_i : C_i \to D_{h(i)}$ for $i \in I$. There exists a canonical embedding of C into $\mathsf{Fam}(\mathsf{C})$ which regards C-objects/arrows as families of C-objects/arrows indexed by a one-element set. Then, multi-limits of C-diagrams correspond to limits in $\mathsf{Fam}(\mathsf{C})$ of the translations of these diagrams along the embedding of C into $\mathsf{Fam}(\mathsf{C})$. In particular, C has a final family of objects if and only if $\mathsf{Fam}(\mathsf{C})$ has a final object.

[Die79] also gives a generalisation of the notion of *couniversal arrow*. A couniversal arrow from a functor $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ to a C-object C is a C-arrow of form $\epsilon_C : \mathsf{U}\bar{C} \to C$ for some D-object $\bar{C}$, having the property that given any D-object D and C-arrow $f : \mathsf{U}D \to C$, there exists a unique factorisation of f through ϵ_C of form $f = \mathsf{U}\bar{f}; \epsilon_C$ with $\bar{f} : D \to \bar{C}$. The notion of *couniversal family of arrows* [Die79] generalises that of couniversal arrow as follows.

Definition 6. *Given a functor* $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ *and a* C*-object* C*, a family of* C*-arrows* $(\epsilon_{C,j} : \mathsf{U}\bar{C}_j \to C)_{j\in J}$ *with* $\bar{C}_j$ *a* D*-object for each* $j \in J$ *is a* **couniversal family of arrows from** U **to** C *if and only if, for any* D*-object* D *and* C*-arrow* $f : \mathsf{U}D \to C$*, there exist unique* $j \in J$ *and* D*-arrow* $\bar{f} : D \to \bar{C}_j$ *such that* $U\bar{f}; \epsilon_{C,j} = f$*. The family* $(\bar{C}_j)_{j\in J}$ *is called a* **cofree family of** D**-objects over** C **w.r.t.** U*. If, for any* C*-object* C*, there exists a couniversal family of arrows from* U *to* C*, then* U *is said to have a* **right multi-adjoint***.*

It should be noted that a right multi-adjoint does not define a functor from C to D, since it maps C-objects to *families* of D-objects.

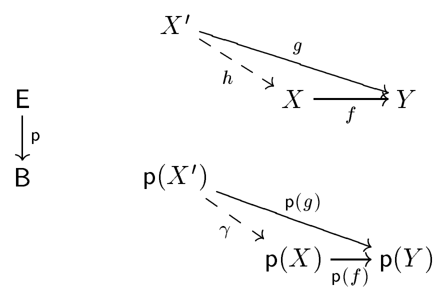
Remark 7. Again, by using categories of families, the concept of couniversal family of arrows can be subsumed under the concept of couniversal arrow. Given categories C and D, a functor $\mathsf{U} : \mathsf{D} \to \mathsf{C}$ induces a functor $\mathsf{Fam}(\mathsf{U}) : \mathsf{Fam}(\mathsf{D}) \to \mathsf{Fam}(\mathsf{C})$, mapping $(D_i)_{i\in I}$ to $(\mathsf{U}(D_i))_{i\in I}$ and $\langle h, (f_i)_{i\in I}\rangle : (D_i)_{i\in I} \to (D'_j)_{j\in J}$ to $\langle h, (\mathsf{U}(f_i))_{i\in I}\rangle$. Then, a couniversal family of arrows from U to C corresponds to a couniversal arrow from $\mathsf{Fam}(\mathsf{U})$ to the one-element family C. Furthermore, the existence of a right multi-adjoint to U yields the existence of a right adjoint R to $\mathsf{Fam}(\mathsf{U})$, and conversely. Given a family $(C_i)_{i\in I}$ of C-objects, for each $i \in I$ let $(\epsilon_{i,j} : \mathsf{U}\bar{C}_{i,j} \to C_i)_{j\in J_i}$ denote a couniversal family of arrows from U to C_i. Also, let $K = \bigcup_{i\in I} J_i$ and define $h : K \to I$ by $h(j) = i$ if $j \in J_i$. Then, $\langle h, (\epsilon_{i,j})_{j\in J_i, i\in I}\rangle : \mathsf{Fam}(\mathsf{U})((\bar{C}_{i,j})_{j\in J_i, i\in I}) \to (C_i)_{i\in I}$ defines a couniversal morphism from $\mathsf{Fam}(\mathsf{U})$ to $(C_i)_{i\in I}$. Conversely, a right adjoint R to $\mathsf{Fam}(\mathsf{U})$ yields a right multi-adjoint to U: a couniversal family of arrows from U to a C-object C is obtained as a couniversal arrow from $\mathsf{Fam}(\mathsf{U})$ to the one-element family C.

2.2 Fibrations

A *fibration* defines an indexing of the objects of a category by objects of another, less structured category, additionally equipped with a way of reindexing objects of the former category along arrows of the latter.

Let $\mathsf{p} : \mathsf{E} \to \mathsf{B}$ be a functor indexing objects of a category E by objects of a category B. B will be called the *base category*, while E will be called the *structure category*. Then, p is said to be a fibration if, for each E-object Y, B-arrows $\alpha : B \to \mathsf{p}(Y)$ can be *lifted* to universal E-arrows $f : X \to Y$, with $\mathsf{p}(f) = \alpha$. This is formalised in the following.

Definition 8. *Let* $\mathsf{p} : \mathsf{E} \to \mathsf{B}$ *be a functor. A* **cartesian map** *for* p *is an* E*-arrow* $f : X \to Y$ *having the property that given any* E*-arrow* $g : X' \to Y$ *such that* $\mathsf{p}(g)$ *factors through* $\mathsf{p}(f)$ *(i.e.* $\mathsf{p}(g) = \mathsf{p}(f) \circ \gamma$ *for some* $\gamma : \mathsf{p}(X') \to \mathsf{p}(X)$*), there exists a unique* E*-arrow* $h : X' \to X$ *with* $\mathsf{p}(h) = \gamma$ *such that* $g = f \circ h$.



f *is alternatively called a* **cartesian lifting** *of* $\mathsf{p}(f)$.

p *is a* **fibration** *if and only if given any* E*-object* Y *and* B*-arrow* $\alpha : B \to \mathsf{p}(Y)$*, there exists a cartesian map* $f : X \to Y$ *with* $\mathsf{p}(f) = \alpha$.

A **cleavage** *for a fibration* p *is a choice of a cartesian map for each* Y *and* α*. A fibration equipped with a cleavage is called a* **cloven fibration**.

Given a B*-object* B*, the subcategory of* E *whose objects are indexed by* B *and whose arrows are indexed by* 1_B *is called the* **fibre over** B *and is denoted* E_B*. The arrows of* E *which are taken by* p *to identities in* B *are called* **vertical**.

Example 9. For any category C, the functor $\mathsf{p} : \mathsf{Fam}(\mathsf{C}) \to \mathsf{Set}$ mapping $(X_i)_{i\in I}$ to I and $\langle h, (f_i)_{i\in I}\rangle : (X_i)_{i\in I} \to (Y_j)_{j\in J}$ to $h : I \to J$ is a fibration. Given $h : I \to J$, any cartesian lifting of h is of the form $\langle h, (1_{h(i)})_{i\in I}\rangle : (Y_{h(i)})_{i\in I} \to (Y_j)_{j\in J}$ for some J-indexed family Y.

A cloven fibration induces, for each arrow $\alpha : B \to B'$ in the base category, a functor $\alpha^* : \mathsf{E}_{B'} \to \mathsf{E}_B$, called *reindexing functor*, which takes an object X of $\mathsf{E}_{B'}$ to the domain $\alpha^*(X)$ of the cartesian map $\bar{\alpha}(X) : \alpha^*(X) \to X$ over α (uniquely determined by the cleavage), and an arrow $m : X \to Y$ of $\mathsf{E}_{B'}$ to the unique (vertical) arrow $\alpha^*(m) : \alpha^*(X) \to \alpha^*(Y)$ satisfying $\bar{\alpha}(Y) \circ \alpha^*(m) = m \circ \bar{\alpha}(X)$.

Definition 10. *Given a fibration* $\mathsf{p} : \mathsf{E} \to \mathsf{B}$ *and a shape category* I*, cartesian liftings in* p *are said to* **preserve** I**-limits** *if, for each* $\alpha : B \to B'$ *in* B*, whenever* *an* I*-shaped diagram* $\mathsf{d} : \mathsf{I} \to \mathsf{E}_{B'}$ *has a limit, any reindexing of this limit along* α *is a limit for any reindexing of* d *along* α.

p **has fibred** I**-limits** *if and only if every fibre of* p *has* I*-limits, and cartesian liftings preserve* I*-limits.*

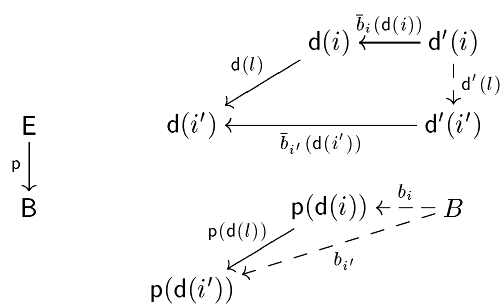
Since limits provide a canonical way of combining objects in a category, completeness is a desirable property of categories in general, and of structure categories of fibrations in particular. A consequence of a result in [Her93] is that completeness of the structure category of a fibration follows from the completeness of the base category and of each of the fibre categories, together with the preservation of limits by cartesian liftings.

Theorem 11 ([Her93]). *Let* I *be a shape category and let* $\mathsf{p} : \mathsf{E} \to \mathsf{B}$ *be a fibration such that* B *has* I*-limits. The following are equivalent:*

(i) p *has fibred* I*-limits*
(ii) E *has, and* p *preserves* I*-limits.*

Here we only recall the way limits in the structure category of a fibration are computed. We let $\mathsf{p} : \mathsf{E} \to \mathsf{B}$ denote a cloven fibration and let $\mathsf{d} : \mathsf{I} \to \mathsf{E}$ denote an I-shaped diagram in E. A limit for d in E is obtained as follows:

1. First, a limit $(B, (b_i)_{i\in|\mathsf{I}|})$ for $\mathsf{p} \circ \mathsf{d} : \mathsf{I} \to \mathsf{B}$ is computed.
2. Next, the diagram d is reindexed to a diagram $\mathsf{d}' : \mathsf{I} \to \mathsf{E}_B$, defined as follows.
   (a) For i in $|\mathsf{I}|$, $\mathsf{d}'(i)$ is the domain of the cartesian map $\bar{b}_i(\mathsf{d}(i)) : \mathsf{d}'(i) \to \mathsf{d}(i)$ over $b_i : B \to \mathsf{p}(\mathsf{d}(i))$.
   (b) For $l : i \to i'$ in $\|\mathsf{I}\|$, $\mathsf{d}'(l)$ is the unique E-arrow satisfying: $\mathsf{d}(l) \circ \bar{b}_i(\mathsf{d}(i)) = \bar{b}_{i'}(\mathsf{d}(i')) \circ \mathsf{d}'(l)$ (given by the universal property of $\bar{b}_{i'}(\mathsf{d}(i'))$).



3. Finally, a limit for d' in E_B is computed.

2.3 Hidden Algebra

This section recalls the underlying definitions of the *hidden algebra* formalism, together with some earlier results on the existence of semantic constructions based on finality in a restricted version of hidden algebra used to specify coalgebraic behaviours.

Hidden algebra was first introduced in [Gog91] and then further developed in [GD94], [MG94], [GM97] as a behavioural algebraic specification formalism for objects. Its syntax reflects the fundamental distinction between (immutable) data values and (mutable) object states through the use of visible sorts and operation symbols for the data, and of hidden sorts and operation symbols for the states of objects. Furthermore, object specifications and their implementations use a fixed specification, respectively implementation for the data, given by a many-sorted signature (V, Ψ) (the **data signature**) and respectively a many-sorted (V, Ψ)-algebra D (the **data algebra**), with the additional constraint that each element of D is named by some constant symbol in Ψ. For convenience, we assume $D_v \subseteq \Psi_{\langle\rangle,v}$ for each $v \in V$.

The operations available for creating and accessing the states of objects are specified using *hidden signatures*.

Definition 12. *A* **(hidden) signature (over** (V, Ψ)**)** *is a pair* (H, Σ) *with* H *a set of* **hidden sorts** *and* Σ *a* $V \cup H$*-sorted signature satisfying:*

1. $\Sigma_{w,v} = \Psi_{w,v}$ *for* $w \in V^*$ *and* $v \in V$
2. *for* $\sigma \in \Sigma_{w,s}$*, at most one sort appearing in* w *(by convention, the first one) is hidden.*

In the following, hidden signatures (H, Σ) will be abbreviated Σ whenever the set of hidden sorts is clear from the context.

Apart from the operation symbols of Ψ, hidden signatures contain operation symbols whose result type is a hidden type (used to *construct* new states), and operation symbols whose argument types include a hidden type (used to *observe* the current states of objects). Some of the $\Sigma \setminus \Psi$-operation symbols, namely those having both a hidden argument and a hidden result, can be viewed both as a means of constructing a new state and as a means of observing an existing state. However, since we are mainly interested in the observational aspect of objects, we will refer to $\Sigma \setminus \Psi$-operation symbols having exactly one hidden argument as *destructor symbols*, and to $\Sigma \setminus \Psi$-operation symbols having only visible arguments as *constructor symbols*. Then, condition 2 of Definition 12 expresses the fact that destructors act on the states of single objects.

An *algebra* of a hidden signature agrees with the data algebra on the interpretation of the visible sorts and operation symbols and, in addition, provides interpretations for the hidden sorts and operation symbols.

Definition 13. *Let* Σ *denote a hidden signature. A* **(hidden)** Σ**-algebra (over** D**)** *is a many-sorted* $(V \cup H, \Sigma)$*-algebra* A *such that* $A\restriction_{\Psi} = D$.

A **(hidden)** Σ**-homomorphism** *between* Σ*-algebras* A *and* B *is a many-sorted* Σ*-homomorphism* $f : A \to B$ *such that* $f_v = 1_{D_v}$ *for* $v \in V$.

Σ-algebras over D and Σ-homomorphisms form a category, which will be denoted $\mathsf{Alg}_D(\Sigma)$.

Remark 14. The fact that hidden algebras use the same data algebra for their visible part and that the visible components of hidden homomorphisms are identities will prove crucial for the forthcoming results. In particular, these restrictions will allow hidden algebraic structures to be regarded as coalgebraic structures, with hidden homomorphisms corresponding to coalgebra homomorphisms.

Hidden algebra takes a behavioural approach to specifying objects – their states are only specified *up to observability*. State observations are formalised by *contexts*, while indistinguishability of states by observations is captured by *behavioural equivalence*.

Definition 15. *Let Σ denote a hidden signature. A* **Σ-context** *for sort $s \in V \cup H$ is an element of $T_\Sigma(\{z\})_v$ with z an s-sorted variable, $T_\Sigma(\{z\})$ denoting the $V \cup H$-sorted set of Σ-terms over $\{z\}$ and $v \in V$. Given $t \in T_\Sigma(\mathcal{V})_s$, we write $c[t]$ for the Σ-term obtained by substituting t for z in c.*

Given a Σ-algebra A, **behavioural equivalence on** *A (denoted $\sim_A$) is given by: $a \sim_{A,s} a'$ if and only if $c_A(a) = c_A(a')$ for all contexts c for sort s, with $s \in V \cup H$ and $a, a' \in A_s$.*

Remark 16. The fact that terms containing visible-sorted variables (other than z) need not be considered when defining contexts is a consequence of each data value in D being named by a constant symbol in Ψ.

Many-sorted equations are used in hidden algebra to constrain the behaviour of system states. The associated notion of satisfaction captures the indistinguishability of the lhs and rhs of equations by observations.

Definition 17. *Let Σ denote a hidden signature. A* **Σ-equation** *is a many-sorted (conditional) $(V \cup H, \Sigma)$-equation of form: $(\forall X)\ l = r$ if $l_1 = r_1, \ldots, l_n = r_n$.*

A Σ-algebra A **behaviourally satisfies** *a Σ-equation e of the above form (written $A \models\!\!\equiv_\Sigma e$) if and only if, for any assignment $\theta : X \to A$ of values in A to the variables in X, $\bar{\theta}(l) \sim_A \bar{\theta}(r)$ whenever $\bar{\theta}(l_i) \sim_A \bar{\theta}(r_i)$ for $i = 1, \ldots, n$ (with $\bar{\theta} : T_\Sigma(X) \to A$ denoting the unique extension of the S-sorted function θ to a many-sorted Σ-homomorphism on the algebra of Σ-terms with variables in X).*

Definition 18. *A* **(hidden) specification** *is a pair (Σ, E) with Σ a hidden signature and E a set of Σ-equations.*

A Σ-algebra A **behaviourally satisfies** *a hidden specification (Σ, E) (written $A \models\!\!\equiv_\Sigma E$) if and only if $A \models\!\!\equiv_\Sigma e$ for each $e \in E$.*

Given a set E of Σ-equations and a Σ-equation e, E is said to **semantically entail** *e (written $E \models\!\!\equiv_\Sigma e$) if and only if $A \models\!\!\equiv_\Sigma E$ implies $A \models\!\!\equiv_\Sigma e$ for any Σ-algebra A.*

The following properties of behavioural satisfaction will be used later on.

Proposition 19. *Let A and B denote Σ-algebras and $f : A \to B$ denote a Σ-homomorphism. Then:*

1. *$B \models\!\!\equiv_\Sigma e$ implies $A \models\!\!\equiv_\Sigma e$ for each Σ-equation e.*
2. *$A \models\!\!\equiv_\Sigma e$ implies $B \models\!\!\equiv_\Sigma e$ for each Σ-equation e in visible-sorted variables.*

Proof (sketch).

1. If X denotes the S-sorted set of variables quantifying e, then any assignment $\theta : X \to A$ translates along f to an assignment $f \circ \theta : X \to B$.
2. If X contains visible-sorted variables only, then any assignment $\theta : X \to B$ is of form $f \circ \theta'$ with $\theta' : X \to A$. (The fact that f is the identity on visible sorts is used here.)

We let $\mathsf{Alg}_D(\Sigma, E)$ denote the full subcategory of $\mathsf{Alg}_D(\Sigma)$ whose objects are Σ-algebras that behaviourally satisfy E.

Proposition 20. *The category $\mathsf{Alg}_D(\Sigma, E)$ has pullbacks.*

Proof (sketch). Pullbacks in $\mathsf{Alg}_D(\Sigma, E)$ are constructed as pullbacks in the category of many-sorted Σ-algebras and Σ-homomorphisms.

We restrict our attention to specifications whose equations have visible-sorted conditions, if any. Given an equation e of form $(\forall X)\ l = r$ if $l_1 = r_1, \ldots, l_n = r_n$ such that $l_1, r_1, \ldots, l_n, r_n$ are all visible-sorted, the **visible consequences** of e are of form: $(\forall X)\ c[l] = c[r]$ if $l_1 = r_1, \ldots, l_n = r_n$ ($c[e]$ for short), with $c \in T_\Sigma(\{z\})$ appropriate for l, r. Then, $A \models\!\!\equiv_\Sigma e$ if and only if $A \models_\Sigma c[e]$ for each $c \in T_\Sigma(\{z\})$ appropriate for e (where $\models$ denotes the standard satisfaction relation of many-sorted equational logic).

Translations from one signature to another are specified using *hidden signature maps*.

Definition 21. *Let Σ and Σ' denote hidden signatures. A* **(hidden) signature map** *$\phi : \Sigma \to \Sigma'$ is a many-sorted signature morphism $\phi : (V \cup H, \Sigma) \to (V \cup H', \Sigma')$ such that $\phi\restriction_\Psi = 1_\Psi$ and $\phi(H) \subseteq H'$.*

Hidden signature maps $\phi : \Sigma \to \Sigma'$ induce reduct functors $\mathsf{U}_\phi : \mathsf{Alg}_D(\Sigma') \to \mathsf{Alg}_D(\Sigma)$. For a Σ'-algebra A' (Σ'-homomorphism f'), we write $A'\restriction_\Sigma$ (respectively $f'\restriction_\Sigma$) for $\mathsf{U}_\phi(A')$ (respectively $\mathsf{U}_\phi(f')$) whenever ϕ is clear from the context.

Definition 22. *Let Σ and Σ' denote hidden signatures and let $\phi : \Sigma \to \Sigma'$ denote a hidden signature map. A Σ'-algebra A' is said to be a* **coextension** *of a Σ-algebra A along ϕ if and only if there exists a Σ-homomorphism $f : \mathsf{U}_\phi(A') \to A$.*

Hidden algebra provides support for the reuse of specifications through the notion of *hidden specification map*.

Definition 23. *A hidden signature map $\phi : \Sigma \to \Sigma'$ defines a* **(hidden) specification map** *$\phi : (\Sigma, E) \to (\Sigma', E')$ if and only if $E' \models\!\!\equiv_{\Sigma'} \phi(c[e])$ for each $e \in E$ and each Σ-context c appropriate for e.*

If $\phi : (\Sigma, E) \to (\Sigma', E')$ is a specification map, the reduct functor $\mathsf{U}_\phi : \mathsf{Alg}_D(\Sigma') \to \mathsf{Alg}_D(\Sigma)$ induced by the signature map $\phi : \Sigma \to \Sigma'$ takes hidden (Σ', E')-algebras to hidden (Σ, E)-algebras.

Definition 23 exploits the fact that the equations in E have visible-sorted conditions, if any. A more general definition of specification maps which does not use such an assumption can be given by requiring that $A' \models\!\equiv_{\Sigma'} E'$ implies $\mathsf{U}_\phi(A') \models\!\equiv_{\Sigma} E$ for any Σ'-algebra A'.

We let Spec denote the category of hidden specifications and specification maps. The following result allows a finite number of specifications related by specification maps to be combined in a canonical way.

Proposition 24. Spec *is finitely cocomplete.*

Proof (sketch). The existence of finite colimits is a consequence of the existence of an initial object and of pushouts. An initial object in Spec is given by the specification with no hidden sorts and no equations, while pushouts in Spec are computed as pushouts in the category of many-sorted specifications of the specifications obtained by replacing each hidden equation by its visible consequences.

Remark 25. It should be noted that colimits of specifications with finite presentations do not, in general, have finite presentations. However, if the signature maps underlying the specification maps are such that any operation symbol in the target signature which has a hidden sort from the source signature as argument sort is itself from the source signature, then finite presentations exist; in this case, the hidden equation itself can be considered instead of its visible consequences.

It has been shown in [GM97] that final algebras exist for hidden signatures containing no constructor symbols. This observation has constituted the starting point of [Cîr98], where the relationship between hidden algebra and coalgebra has been further investigated. The rest of this section recalls the results in [Cîr98] regarding the existence of semantic constructions based on finality in hidden algebra.

The results in [Cîr98] concern hidden specifications whose underlying signatures consist only of destructor symbols, and whose equations relate different observations of the *same* hidden state. These conditions are formalised in the following definition.

Definition 26. *A hidden signature Σ is a* **destructor signature** *if and only if $(\Sigma \setminus \Psi)_{w,h} = \emptyset$ for any $w \in V^*$ and any $h \in H$.*

A hidden specification (Σ, E) is a **destructor specification** *if and only if Σ is a destructor signature, and each equation in E is quantified over one hidden-sorted variable (and possibly some visible-sorted variables).*

A first result in [Cîr98] shows the existence of a one-to-one correspondence between hidden algebras of destructor signatures and coalgebras of endofunctors induced by such signatures. This correspondence automatically yields a final algebra for each destructor signature, as well as a coalgebraic formulation of behavioural equivalence on a hidden algebra as greatest bisimulation on the associated coalgebra.

Theorem 27. *For any destructor signature Δ, there exists a final Δ-algebra, having hidden carriers:*

$$F_{\Delta,h} = \prod_{v \in V} [L_\Delta(\{z\})_v \to D_v], \quad h \in H$$

(with $L_\Delta(\{z\}) \subseteq T_\Delta(\{z\})$ consisting of those Δ-contexts in which the variable $z : h$ occurs exactly once) and hidden operations:

- $\delta_{F_\Delta}((s_v)_{v \in V}, \bar{d}) = s_{v'}(\delta(z, \bar{d}))$, *for* $\delta \in \Delta_{hw,v'}$ *and* $\bar{d} \in D_w$
- $\delta_{F_\Delta}((s_v)_{v \in V}, \bar{d}) = (s'_v)_{v \in V}$ *with* $s'_v(c) = s_v(c[\delta(z, \bar{d})])$, $c \in L_\Delta(\{z'\})_v$, *for* $\delta \in \Delta_{hw,h'}$ *and* $\bar{d} \in D_w$

Furthermore, behavioural equivalence on a Δ-algebra A is given by the kernel of the unique Δ-homomorphism of A into F_Δ.

The fact that destructor specifications induce predicates on the carriers of algebras of the underlying signatures can be used to lift the existence of final algebras from signatures to specifications. The elements of the final algebra of a destructor specification provide abstract descriptions of all the behaviours over the specified destructors which satisfy the constraints imposed by the equations.

Theorem 28. *Let (Δ, E) denote a destructor specification and let F denote a final Δ-algebra. There exists a final (Δ, E)-algebra, having hidden carriers:*

$$F_{E,h} = \{\ f \in F_h \mid l_F(t_F(f), \bar{d}) = r_F(t_F(f), \bar{d}) \text{ for any } t \in L_\Delta(\{z\})_{h'}, \\ ((\forall H')(\forall V_1) \ldots (\forall V_n)\ l = r) \in E \text{ and } \bar{d} \in D_{v_1} \times \ldots \times D_{v_n}\ \}, \quad h \in H$$

Corollary 29. *Let (Δ, E) denote a destructor specification. Then, $\mathsf{Alg}_D(\Delta, E)$ has finite limits.*

Proof. The existence of finite limits in $\mathsf{Alg}_D(\Delta, E)$ is a consequence of the existence of a final object (Theorem 28) and of pullbacks (Proposition 20).

The main result in [Cîr98] shows the existence of cofree constructions w.r.t. reduct functors induced by destructor specification maps. Such constructions are then shown to provide suitable denotations for the reuse of specifications along destructor specification maps, as well as a canonical way of reusing implementations along the underlying reuse of specifications.

Theorem 30. *Let (Δ, E) and (Δ', E') denote destructor specifications and let $\phi : (\Delta, E) \to (\Delta', E')$ denote a specification map. Then, the reduct functor $\mathsf{U}_\phi : \mathsf{Alg}_D(\Delta', E') \to \mathsf{Alg}_D(\Delta, E)$ has a right adjoint C_ϕ.*

The counit of the adjunction yields, for each (Δ, E)-algebra A, a couniversal arrow $\epsilon_A : \mathsf{U}_\phi(\mathsf{C}_\phi(A)) \to A$ from U_ϕ to A. That is, $\mathsf{C}_\phi(A)$ coextends A along ϕ. Furthermore, the universal property of ϵ_A makes $\mathsf{C}_\phi(A)$ final among all (Δ', E')-coextensions of A along ϕ. $\mathsf{C}_\phi(A)$ will be called a **cofree coextension of** A **along** ϕ.

3 Semantics with Final/Cofree Families

Due to the possibility of underspecifying constructor operations, existence of final/cofree hidden algebras does not generalise to arbitrary hidden specifications and specification maps. However, as already suggested in [Cîr98], final/cofree families of hidden algebras can be used to characterise all possible ways of resolving the nondeterminism arising from underspecification. Here we prove the existence of such constructions in hidden algebra and emphasise their suitability as denotations for hidden specifications/specification maps.

Theorem 31. *Let (Σ, E) denote a hidden specification. If each equation in E contains at most one hidden-sorted variable, then there exists a final family of hidden (Σ, E)-algebras.*

Proof. We define a relation $\sim$ on hidden (Σ, E)-algebras and use it to partition the category $\mathsf{Alg}_D(\Sigma, E)$ into subcategories. Next we show that each of these subcategories has a final object. It then follows that $\mathsf{Alg}_D(\Sigma, E)$ has a final family of objects.

Given (Σ, E)-algebras A and B, we let $A \sim B$ if and only if there exist a (Σ, E)-algebra C and Σ-homomorphisms $f : C \to A$ and $g : C \to B$. Since $\mathsf{Alg}_D(\Sigma, E)$ has pullbacks (see Proposition 20), it follows that $A \sim B$ holds if and only if A and B are *connected* in $\mathsf{Alg}_D(\Sigma, E)$, i.e. there exists a *zigzag morphism* from A to B in $\mathsf{Alg}_D(\Sigma, E)$ (see [Bor94], page 58). Hence, $\sim$ determines a partition $\mathcal{C}$ of $\mathsf{Alg}_D(\Sigma, E)$ into subcategories.

We now show that each category in $\mathcal{C}$ has a final object. For this, we fix such a category C. Also, we let Δ denote the destructor subsignature of Σ (consisting of all the sorts and all the destructor symbols of Σ), and let F_Δ denote a final Δ-algebra. We define a many-sorted subset F_C of F_Δ as follows:

- $F_{\mathsf{C},h} = \{f \in F_{\Delta,h} \mid f = f_A(a) \text{ for some } A \in |\mathsf{C}| \text{ and } a \in A_h\}, \quad h \in H$
- $F_{\mathsf{C},v} = D_v, \quad v \in V$

where, for a Σ-algebra A, $f_A : A{\restriction}_\Delta \to F_\Delta$ denotes the unique Δ-homomorphism of its Δ-reduct into F_Δ. Then, F_C defines a Δ-subalgebra of F_Δ: given $f \in F_{\mathsf{C},h}$ with $f = f_A(a)$ for some $A \in |\mathsf{C}|$ and $a \in A_h$, and given $\delta \in \Delta_{hw,h'}$ with $h, h' \in H$ and $w \in V^*$, we have: $\delta_{F_\Delta}(f, \bar{d}) = f_A(\delta_A(a, \bar{d}))$, and hence $\delta_{F_\Delta}(f, \bar{d}) \in F_{\mathsf{C},h'}$ for each $\bar{d} \in D_w$. Moreover, F_C can be given the structure of a Σ-algebra by arbitrarily choosing $A \in |\mathsf{C}|$ and then letting $\gamma_{F_\mathsf{C}}(\bar{d}) = f_A(\gamma_A(\bar{d}))$ for each $\gamma \in \Sigma_{w,h}$ with $w \in V^*$ and $h \in H$, and each $\bar{d} \in D_w$. The definition of $\sim$ together with uniqueness of a Δ-homomorphism into a final Δ-algebra ensure that the definition of γ_{F_C} does not depend on the choice of A. Then, $F_\mathsf{C} \models_\Sigma E$ follows from each $e \in E$ containing at most one hidden-sorted variable: in this case, any assignment of values in F_C to the variables in e is obtained by post-composing a similar assignment into some $A \in |\mathsf{C}|$ with f_A; behavioural satisfaction of e in (a state f of) F_C then follows from its behavioural satisfaction in (a state a of) A, with $A \in |\mathsf{C}|$.

Hence, $F_\mathsf{C} \in |\mathsf{C}|$; furthermore, F_C is final in C: given $A \in |\mathsf{C}|$, $A{\restriction}_\Delta$ has a unique Δ-homomorphism f_A into F_Δ which, by the definition of F_C, defines a

Σ-homomorphism $f_A : A \to F_{\mathsf{C}}$. Uniqueness of such a Σ-homomorphism follows from uniqueness of a Δ-homomorphism into F_Δ.

It then follows that $(F_{\mathsf{C}})_{\mathsf{C}\in\mathcal{C}}$ is a final family of hidden (Σ, E)-algebras: given any (Σ, E)-algebra A, say $A \in |\mathsf{C}|$ for some $\mathsf{C} \in \mathcal{C}$, there exists a unique Σ-homomorphism $f_A : A \to F_{\mathsf{C}}$; also, for $\mathsf{C}' \neq \mathsf{C}$, there exists no Σ-homomorphism of A into $F_{\mathsf{C}'}$, as C and C' are disjoint. This concludes the proof.

Remark 32. The existence of a final family of (Σ, E)-algebras results in the existence of a final object in the category $\mathsf{Fam}(\mathsf{Alg}_D(\Sigma, E))$ (see Remark 5), given by $(F_{\mathsf{C}})_{\mathsf{C}\in\mathcal{C}}$.

The next result states an important property of the final family.

Theorem 33. *Let (Σ, E) denote a hidden specification, $(F_i)_{i\in I}$ denote a final family of hidden (Σ, E)-algebras and e denote an arbitrary Σ-equation. Then, e is behaviourally satisfied by any (Σ, E)-algebra if and only if e is behaviourally satisfied by each F_i, with $i \in I$.*

Proof. The **only if** direction follows by each F_i being a (Σ, E)-algebra. For the **if** direction, given an arbitrary (Σ, E)-algebra A, existence of a Σ-homomorphism from A to one of the F_is together with Proposition 19 and $F_i \models\!\equiv_\Sigma e$ yield $A \models\!\equiv_\Sigma e$.

The above result justifies the use of final families as denotations for hidden specifications satisfying the hypothesis of Theorem 31.

The proof of Theorem 31 also provides some information about how the algebras in the final family look like: for a hidden specification (Σ, E), the Δ-reduct of each algebra in the final family is a Δ-subalgebra of a final Δ-algebra (with Δ denoting the destructor subsignature of Σ). However, in most cases, the final family has a more concrete representation than the one above. Such cases correspond to *split specifications*.

Definition 34. *Given a hidden signature Σ with destructor subsignature Δ, a hidden specification (Σ, E) is called* **split** *if and only if $E = E_\Delta \cup E_\Sigma$, with E_Δ consisting of Δ-equations in one hidden-sorted variable and E_Σ consisting of Σ-equations in no hidden-sorted variables.*

The intuition behind the above definition is that E_Δ constrains the state space of Σ-algebras (by means of equations that use Δ-symbols only), whereas E_Σ constrains the interpretation of the constructor symbols in the state space defined by E_Δ, without imposing further constraints to this state space.

Proposition 35. *Let (Σ, E) denote a split hidden specification ($E = E_\Delta \cup E_\Sigma$), let F_{Δ,E_Δ} denote a final (Δ, E_Δ)-algebra, and let $\mathcal{F} = \{F \in \mathsf{Alg}_D(\Sigma) \mid F{\restriction}_\Delta = F_{\Delta,E_\Delta},\ F \models\!\equiv_\Sigma E_\Sigma\}$. Then, $\mathcal{F}$ defines a final family of hidden (Σ, E)-algebras.*

Proof. We must show that an arbitrary (Σ, E)-algebra A has exactly one Σ-homomorphism into an $F \in \mathcal{F}$. Any such homomorphism must extend the unique Δ-homomorphism $f_A : A{\restriction}_\Delta \to F_{\Delta,E_\Delta}$ resulting from $A{\restriction}_\Delta \models\!\equiv_\Delta E_\Delta$ on one hand, and must preserve the $\Sigma \setminus \Delta$-structure on the other. Hence, the *only* $F \in \mathcal{F}$

that A can have a Σ-homomorphism into has its $\Sigma \setminus \Delta$-structure induced by the $\Sigma \setminus \Delta$-structure of A: given $\gamma \in (\Sigma \setminus \Delta)_{w,h}$ with $w \in V^*$ and $h \in H$, $\gamma_F(\bar{d}) = f_A(\gamma_A(\bar{d}))$ for each $\bar{d} \in D_w$. Since all the equations in E_Σ are quantified over data only and since $A \models\!\equiv_\Sigma E_\Sigma$, it follows by Proposition 19 that $F \models\!\equiv_\Sigma E_\Sigma$. This concludes the proof.

Therefore, the carriers of all the algebras in the final family of a *split* hidden specification coincide with the carrier of the final algebra of its destructor subspecification.

Finally, it is worth noting that for a hidden specification (Σ, E), the final family of (Σ, E)-algebras may be empty – this happens precisely when the specification (Σ, E) is *inconsistent*, i.e. when there are no (Σ, E)-algebras.

Example 36. We use a specification of one-place buffers to exemplify the construction of the final family of algebras of a specification satisfying the hypothesis of Theorem 31.

The data universe underlying this specification includes the visible sorts `Bool` for the booleans (interpreted by D as {`true`,`false`}) and `Val` for the values to be stored by buffers. Then, one-place buffers are specified using a hidden sort `Buffer`, operation symbols:

```
empty : → Buffer
empty? : Buffer → Bool
val : Buffer → Val
put : Buffer Val → Buffer
get : Buffer → Buffer
```

and equations:

```
empty?(empty) = true
empty?(put(B,V)) = false
val(put(B,V)) = V if empty?(B) = true
put(B,V) = B if empty?(B) = false
get(B) = empty
```

We note that this specification is not split, as the last equation contains both a hidden-sorted variable and a constructor symbol.

The final algebra of the destructor subsignature of the buffer signature (consisting of all the operation symbols except from `empty`) has its elements defined by mappings of form $f : (\{\texttt{put(v)} \mid \texttt{v} \in D_{\texttt{Val}}\} \cup \{\texttt{get}\})^* \to D_{\texttt{Bool}} \times D_{\texttt{Val}}$ (with A^* denoting the set of finite sequences of elements of A). Then, the image of any algebra satisfying the buffer specification under the unique homomorphism into the final algebra of the destructor subsignature of the buffer signature will consist of mappings f of the above form, additionally satisfying the following:

1. the value of f on any sequence ending with `get` is $\langle \texttt{true}, \texttt{v}_0 \rangle$, with $\texttt{v}_0 \in D_{\texttt{Val}}$ being given by the interpretation of `val(empty)` in the algebra (as `get(B) = empty` and `empty?(empty) = true` hold)
2. the value of f on a sequence containing successive `put`s coincides with the value of f on the sequence obtained by eliminating all `put`s preceded by

another `put` (as `put(B,V) = B if empty?(B) = false` and `empty?(put(B,V)) = false` hold)

3. the value of f on any sequence ending with `get;put(v)` is $\langle$`false,v`$\rangle$ (as `empty?(put(B,V)) = false`, `val(put(B,V)) = V if empty?(B) = true`, `get(B) = empty` and `empty?(empty) = true` hold)
4. the value of f on the sequence `put(v)` is either the value of f on the empty sequence, if this value is of form $\langle$`false,v'`$\rangle$ (as `put(B,V) = B if empty?(B) = false` holds), or $\langle$`false,v`$\rangle$, if the value of f on the empty sequence is of form $\langle$`true,v'`$\rangle$ (as `val(put(B,V)) = V if empty?(B) = true` and `empty?(put(B,V)) = false` hold).

Hence, f is completely determined by its value on the empty sequence. Moreover, all the values in $D_{\mathtt{Bool}} \times D_{\mathtt{Val}}$ are reached by some homomorphism, independently of the value $\mathtt{v}_0$. Hence, all the algebras in the final family have their carrier given by $D_{\mathtt{Bool}} \times D_{\mathtt{Val}}$. The only thing that distinguishes these algebras is the value $\mathtt{v}_0$ defining `val(empty)`.

A different final family of algebras would be obtained if the equation:

```
B=empty if empty?(B) = true
```

(identifying all the empty buffers up to behavioural equivalence) was added to the specification. In this case, the carriers of the algebras in the final family would not coincide anymore – each such carrier would be of form: $\{\langle \mathtt{false,v}\rangle \mid \mathtt{v} \in D_{\mathtt{Val}}\} \cup \{\langle \mathtt{true,v_0}\rangle\}$, for some (fixed) $\mathtt{v}_0 \in D_{\mathtt{Val}}$, while the corresponding algebra would interpret `empty` as $\langle \mathtt{true,v_0}\rangle$.

We have seen that cofree algebras provide suitable denotations for the reuse of specifications along destructor specification maps. When specifications comprising both algebraic and coalgebraic structure are considered, the semantics involves *cofree families of algebras*.

Theorem 37. *Let $\phi : (\Sigma, E) \to (\Sigma', E')$ denote a hidden specification map. If each equation in E' contains at most one hidden-sorted variable, then the reduct functor $\mathsf{U}_\phi : \mathsf{Alg}_D(\Sigma', E') \to \mathsf{Alg}_D(\Sigma, E)$ has a right multi-adjoint.*

Proof. We let Δ and Δ' denote the destructor subsignatures of Σ and Σ' respectively, and let $\phi_\Delta : \Delta \to \Delta'$ denote the restriction of the signature map $\phi : \Sigma \to \Sigma'$ to destructor subsignatures. We fix a (Σ, E)-algebra A and construct a cofree family of (Σ', E')-algebras over A. We let $\bar{A}$ denote the cofree coextension of $A\restriction_\Delta$ along ϕ_Δ, with $\epsilon_A : \bar{A}\restriction_\Delta \to A\restriction_\Delta$ as the associated couniversal arrow.

The proof now follows the same line as the proof of Theorem 31. We consider a category $\mathsf{Alg}_D(\Sigma', E', A)$ whose objects correspond to (Σ', E')-coextensions of A, and use a relation $\sim$ on its objects to partition it into subcategories with final objects. These final objects then yield a final family for $\mathsf{Alg}_D(\Sigma', E', A)$, which at the same time defines a cofree family of (Σ', E')-algebras over A.

$\mathsf{Alg}_D(\Sigma', E', A)$ is the category whose objects are pairs $\langle A', f\rangle$ with A' a (Σ', E')-algebra and $f : A'\restriction_\Sigma \to A$ a Σ-homomorphism, and whose arrows from

$\langle A'_1, f_1\rangle$ to $\langle A'_2, f_2\rangle$ are Σ'-homomorphisms $g : A'_1 \to A'_2$ such that $\bar{f}_1 = \bar{f}_2 \circ g\restriction_{\Delta'}$ (where $\bar{f}_1 : A'_1\restriction_{\Delta'} \to \bar{A}$ and $\bar{f}_2 : A'_2\restriction_{\Delta'} \to \bar{A}$ denote the unique Δ'-homomorphisms satisfying $\epsilon_A \circ \bar{f}_1 \restriction_{\Delta} = f_1 \restriction_{\Delta}$, respectively $\epsilon_A \circ \bar{f}_2 \restriction_{\Delta} = f_2 \restriction_{\Delta}$). Given $\langle A'_1, f_1\rangle$ and $\langle A'_2, f_2\rangle$ in $\mathsf{Alg}_D(\Sigma', E', A)$, $\langle A'_1, f_1\rangle \sim \langle A'_2, f_2\rangle$ if and only if there exist $\langle A', f\rangle$ together with $g_1 : \langle A', f\rangle \to \langle A'_1, f_1\rangle$, $g_2 : \langle A', f\rangle \to \langle A'_2, f_2\rangle$ in $\mathsf{Alg}_D(\Sigma', E', A)$. One can easily show that $\mathsf{Alg}_D(\Sigma', E', A)$ has pullbacks, and therefore $\langle A'_1, f_1\rangle \sim \langle A'_2, f_2\rangle$ holds if and only if $\langle A'_1, f_1\rangle$ and $\langle A'_2, f_2\rangle$ are connected in $\mathsf{Alg}_D(\Sigma', E', A)$. Hence, $\sim$ determines a partition $\mathcal{C}$ of $\mathsf{Alg}_D(\Sigma', E', A)$ into subcategories. Furthermore, each such subcategory C has a final object $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle$.

The hidden carriers of $\bar{A}_{\mathsf{C}}$ are given by:

$$\bar{A}_{\mathsf{C},h} = \{a \in \bar{A}_h \mid a = \bar{f}(a') \text{ for some } \langle A', f\rangle \in |\mathsf{C}| \text{ and } a' \in A'_h\}, \quad h \in H$$

where $\bar{f} : A'\restriction_{\Delta'} \to \bar{A}$ denotes the unique Δ'-homomorphism satisfying $\epsilon_A \circ \bar{f}\restriction_{\Delta} = f\restriction_{\Delta}$. Then, $(\bar{A}_{\mathsf{C},h})_{h\in H}$ defines a Δ'-subcoalgebra of $\bar{A}$. The Δ'-structure of $\bar{A}_{\mathsf{C}}$ is therefore induced by the Δ'-structure of $\bar{A}$. Also, the $\Sigma' \setminus \Delta'$-structure of $\bar{A}_{\mathsf{C}}$ is induced by the $\Sigma' \setminus \Delta'$-structure of (any of) the (Σ', E')-algebras in C: $\gamma'_{\bar{A}_{\mathsf{C}}}(\bar{d}) = \bar{f}(\gamma'_{A'}(\bar{d}))$ for some $\langle A', f\rangle \in |\mathsf{C}|$, for each $\gamma' \in \Sigma'_{w,h'}$ with $w \in V^*$ and $h' \in H'$. (The definition of $\mathsf{Alg}_D(\Sigma', E', A)$ ensures that the definition of $\gamma'_{\bar{A}_{\mathsf{C}}}$ does not depend on the choice of $\langle A', f\rangle$.) Also, $\bar{A}_{\mathsf{C}}$ behaviourally satisfies E', since each algebra in C does and since each equation in E' contains at most one hidden-sorted variable. Finally, the Δ-homomorphism $\epsilon_A : \bar{A}\restriction_{\Delta} \to A$ defines a Σ-homomorphism $\epsilon_{A,\mathsf{C}} : \bar{A}_{\mathsf{C}}\restriction_{\Sigma} \to A$. (The way $\Sigma' \setminus \Delta'$-operation symbols are interpreted in $\bar{A}_{\mathsf{C}}$ is used to prove this.) Hence, $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle \in |\mathsf{C}|$.

We now show that $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle_{\mathsf{C}\in\mathcal{C}}$ defines a final $\mathsf{Alg}_D(\Sigma', E', A)$-family. Given any $\langle A', f\rangle \in |\mathsf{Alg}_D(\Sigma', E', A)|$, say $\langle A', f\rangle \in |\mathsf{C}|$ with $\mathsf{C} \in \mathcal{C}$, $\bar{f} : A'\restriction_{\Delta'} \to \bar{A}$ defines a Σ'-homomorphism $g : A' \to \bar{A}_{\mathsf{C}}$, and this is the *only* Σ'-homomorphism from A' to $\bar{A}_{\mathsf{C}}$. From $\bar{f} = g\restriction_{\Delta'} = 1_{\bar{A}} \circ g\restriction_{\Delta'} = \bar{\epsilon}_A \circ g\restriction_{\Delta'}$, it follows that g defines an arrow from $\langle A', f\rangle$ to $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle$ in $\mathsf{Alg}_D(\Sigma', E', A)$. Also, for $\mathsf{C}' \neq \mathsf{C}$, $\langle A', f\rangle$ has no arrow into $\langle \bar{A}_{\mathsf{C}'}, \epsilon_{A,\mathsf{C}'}\rangle$, as C and C' are disjoint. Hence, $g : \langle A', f\rangle \to \langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle$ is the only $\mathsf{Alg}_D(\Sigma', E', A)$-arrow from $\langle A', f\rangle$ into an object of $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle_{\mathsf{C}\in\mathcal{C}}$.

The universal property of $\langle \bar{A}_{\mathsf{C}}, \epsilon_{A,\mathsf{C}}\rangle_{\mathsf{C}\in\mathcal{C}}$ as a final family together with $\epsilon_{A,\mathsf{C}} \circ g\restriction_{\Sigma} = f$ (following from $\epsilon_A \circ \bar{f}\restriction_{\Delta} = f\restriction_{\Delta}$) for any $\langle A', f\rangle \in |\mathsf{C}|$ then result in $(\epsilon_{A,\mathsf{C}} : \bar{A}_{\mathsf{C}}\restriction_{\Sigma} \to A)_{\mathsf{C}\in\mathcal{C}}$ being a couniversal family of arrows from U_ϕ to A, and therefore in $(\bar{A}_{\mathsf{C}})_{\mathsf{C}\in\mathcal{C}}$ being a cofree family of (Σ', E')-algebras over A.

Right multi-adjoints to the reduct functors induced by hidden specification maps satisfying the hypothesis of Theorem 37 provide suitable denotations for specification steps given by such specification maps: given an algebra A of the source specification, the right multi-adjoint yields a family of algebras of the target specification each of whose elements is maximal in the class of algebras that coextend A.

For a hidden specification map $\phi : (\Sigma, E) \to (\Sigma', E')$, the right multi-adjoint to U_ϕ yields a right adjoint to the functor $\mathsf{Fam}(\mathsf{U}_\phi) : \mathsf{Fam}(\mathsf{Alg}_D(\Sigma', E')) \to \mathsf{Fam}(\mathsf{Alg}_D(\Sigma, E))$ (see Remark 7). This right adjoint can alternatively be used as denotation for the hidden specification map ϕ.

A result similar to Proposition 35 can be stated for hidden specification maps whose codomain is a split hidden specification.

Proposition 38. *Let $\phi : (\Sigma, E) \to (\Sigma', E')$ denote a hidden specification map such that (Σ', E') is a split hidden specification (i.e. $E' = E'_{\Delta'} \cup E'_{\Sigma'}$ with $E'_{\Delta'}$ consisting of Δ'-equations in one hidden-sorted variable and $E'_{\Sigma'}$ consisting of Σ'-equations in no hidden-sorted variables). Also, let $\phi_\Delta : (\Delta, \emptyset) \to (\Delta', E'_{\Delta'})$ denote the hidden specification map induced by the signature map $\phi\restriction_\Delta : \Delta \to \Delta'$. Then, for any (Σ, E)-algebra A, with $\epsilon_A : \mathsf{U}_{\phi_\Delta}(\bar{A}) \to A\restriction_\Delta$ as a couniversal arrow from U_{ϕ_Δ} to $A\restriction_\Delta$ (given by Theorem 30), the family $\mathcal{A} = \{A' \in \mathsf{Alg}_D(\Sigma', E') \mid A'\restriction_{\Delta'} = \bar{A},\ \epsilon_A$ defines a Σ-homomorphism from $A'\restriction_\Sigma$ to $A,\ A' \models_{\Sigma'} E'_{\Sigma'}\}$ defines a cofree family of (Σ', E')-algebras over A w.r.t. U_ϕ.*

Proof. Similar to the proof of Proposition 35.

That is, if (Σ', E') is split, then the carriers of all the algebras in the cofree family of (Σ', E')-algebras over A w.r.t. U_ϕ coincide with the carrier of the cofree $(\Delta', E'_{\Delta'})$-algebra over $A\restriction_\Delta$ w.r.t. U_{ϕ_Δ}, where $(\Delta', E'_{\Delta'})$ denotes the destructor subspecification of (Σ', E').

We conclude this section by noting that initial and respectively free families of hidden algebras also exist (no restriction on the specifications involved is needed in this case). Although initial families do not satisfy properties similar to the ones stated in Theorem 33, they are relevant for characterising behaviours which are reachable through ground Σ-terms. A consequence of the existence of both initial and final families of hidden specifications is the existence of a partition of the category of hidden algebras of such a specification into subcategories, with each subcategory corresponding to a particular behaviour for the constructor operations, and having an initial as well as a final representative.

4 Semantics with Multi-limits

In algebraic approaches to the specification of data types, colimit constructions provide canonical ways of combining specifications, while free extensions of algebras together with colimit (pushout) constructions yield a compositional semantics for such combined specifications [EM85], [EBO93]. In hidden algebra, colimits are used in a similar way at the specification level. However, at the model level the interest is in *coextending* (restricting) collections of behaviours, rather than in *extending* collections of values, and consequently dual constructions should be considered. Since categories of hidden algebras do not, in general, have finite limits, multi-limits are the obvious candidate for such constructions – like standard limits, they define final solutions to categorically-formulated constraints. Here we prove the existence of multi-limits in a general category of hidden algebras. This then yields a canonical construction for algebras of structured specifications from algebras of their component specifications, as well as a compositional semantics for structured hidden specifications.

Theorem 39. *Let (Σ, E) denote a hidden specification such that each equation in E contains at most one hidden-sorted variable. Then, the category $\mathsf{Alg}_D(\Sigma, E)$ has finite multi-limits. Furthermore, if (Σ, E) is a destructor specification, then finite multi-limits coincide with finite limits.*

Proof. By Theorem 31, $\mathsf{Alg}_D(\Sigma, E)$ has a final family. Also, by Proposition 20, $\mathsf{Alg}_D(\Sigma, E)$ has pullbacks, and hence multi-pullbacks. It then follows by Theorem 4 that $\mathsf{Alg}_D(\Sigma, E)$ has finite multi-limits. Furthermore, if (Σ, E) is a destructor specification, the final (Σ, E)-family is given by a final (Σ, E)-algebra (see Theorem 27). The existence of finite limits in $\mathsf{Alg}_D(\Sigma, E)$ then follows from the existence of a final object and of pullbacks.

Theorem 39 will now be used to prove a similar result for a general category Alg_D, whose objects are hidden algebras and whose arrows correspond to coextension relations between their source and target. One can also consider a subcategory CoAlg_D of Alg_D, whose objects are hidden algebras of destructor specifications. CoAlg_D will be shown to have finite limits, while Alg_D will be shown to have finite multi-limits.

Theorem 40. *Let Alg_D denote the category having:*

- *objects: pairs $\langle P, A\rangle$, with P a hidden specification whose equations contain at most one hidden-sorted variable, and A a P-algebra;*
- *arrows from $\langle P', A'\rangle$ to $\langle P, A\rangle$: pairs $\langle \phi, f\rangle$, with $\phi : P \to P'$ a hidden specification map, and $f : A'\restriction_P \to A$ a Σ_P-homomorphism.*

Also, let CoAlg_D denote the full subcategory of Alg_D whose objects are such that their first component is a destructor specification. Then, the following hold:

1. *CoAlg_D has finite limits.*
2. *Alg_D has finite multi-limits.*

Proof. We start by noting that an arrow $\langle \phi, f\rangle : \langle P', A'\rangle \to \langle P, A\rangle$ in Alg_D corresponds to A' coextending A along ϕ via f.

We prove 1 by viewing the category CoAlg_D as the structure category of a fibration satisfying the hypotheses of Theorem 11. We let $\mathsf{CoSp} : \mathsf{CoAlg}_D \to \mathsf{Spec}^{\mathsf{op}}$ be given by:

- $\mathsf{CoSp}(\langle P, A\rangle) = P$ for each $\langle P, A\rangle \in |\mathsf{CoAlg}_D|$
- $\mathsf{CoSp}(\langle \phi, f\rangle) = \bar{\phi}$ for each $\langle \phi, f\rangle \in \| \mathsf{CoAlg}_D \|$, with $\bar{\phi}$ denoting the $\mathsf{Spec}^{\mathsf{op}}$ arrow induced by the Spec-arrow ϕ.

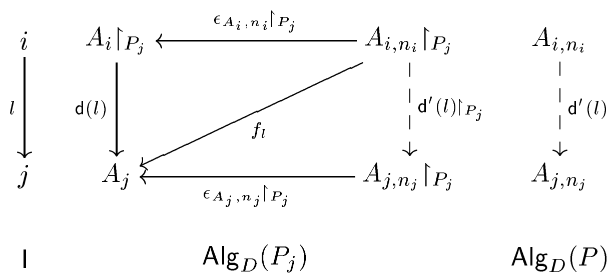
The existence of cofree coextensions along specification maps ϕ between destructor specifications (see Theorem 30) makes CoSp a fibration whose cartesian liftings along arrows $\bar{\phi} \in \| \mathsf{Spec}^{\mathsf{op}} \|$ are the couniversal arrows induced by the adjunction $\mathsf{U}_\phi \dashv \mathsf{C}_\phi$, and whose reindexing functors along arrows $\bar{\phi} \in \| \mathsf{Spec}^{\mathsf{op}} \|$ are the right adjoints C_ϕ to the reduct functors U_ϕ.

We now verify that all the hypotheses of Theorem 11 hold for CoSp. The fact that $\mathsf{Spec}^{\mathsf{op}}$ has finite limits is guaranteed by Proposition 24. Also, the fact that every fibre of CoSp has finite limits follows from Theorem 29 together with the fact that the fibres over specifications other than destructor ones are all empty. Finally, preservation of finite limits by the reindexing functors follows from the limit-preservation property of right adjoints.

It then follows by Theorem 11 ((i) $\Rightarrow$ (ii)) that CoAlg_D has finite limits.

The existence of finite limits in CoAlg_D does not generalise to Alg_D. For, given an arbitrary specification (Σ, E), $\mathsf{Alg}_D(\Sigma, E)$ does not, in general, have all finite limits. Also, the functor $\mathsf{Sp} : \mathsf{Alg}_D \to \mathsf{Spec}^{\mathsf{op}}$ which extends CoSp in a natural way is not a fibration, as cofree constructions along arbitrary specification maps do not exist in general. Still, one can use a strategy similar to the one in the proof of Theorem 11 to construct multi-limits in Alg_D. Specifically, given a finite diagram $\mathsf{d} : \mathsf{I} \to \mathsf{Alg}_D$, its multi-limit is obtained by:

1. constructing the limit of $\mathsf{Sp} \circ \mathsf{d}$ in $\mathsf{Spec}^{\mathsf{op}}$; this limit corresponds to a colimit $(P, (\phi_i : P_i \to P)_{i \in |\mathsf{I}|})$ in Spec, where, for $i \in |\mathsf{I}|$, $P_i = \mathsf{Sp}(\mathsf{d}(i))$;
2. for each $i \in |\mathsf{I}|$, cofreely coextending $A_i = \mathsf{d}(i)$ along ϕ_i to a family $(A_{i,n})_{n \in N_i}$ of P-algebras, with $(\epsilon_{A_i,n} : A_{i,n}\restriction_{P_i} \to A_i)_{n \in N_i}$ as the associated couniversal family;
3. computing the multi-limit of each diagram $\mathsf{d}' : \mathsf{I} \to \mathsf{Alg}_D(P)$ additionally satisfying:
   (a) for i in $|\mathsf{I}|$, $\mathsf{d}'(i) = A_{i,n_i}$ for some $n_i \in N_i$;
   (b) for $l : i \to j$ in $\|\mathsf{I}\|$, n_j is determined (uniquely) by the P_j-homomorphism $f_l = \mathsf{d}(l) \circ \epsilon_{A_i,n_i}\restriction_{P_j}$, while $\mathsf{d}'(l) = \bar{f}_l : A_{i,n_i} \to A_{j,n_j}$:



   each such multi-limit yields a family $(\langle P, L^k \rangle, (\langle \phi_i, l_i^k \rangle)_{i \in |\mathsf{I}|})_{k \in K_{\mathsf{d}'}}$ of cones for d;
4. taking the union $(\langle P, L^k \rangle, (\langle \phi_i, l_i^k \rangle)_{i \in |\mathsf{I}|})_{k \in \bigcup K_{\mathsf{d}'}}$ of all these families.

Then, the family $(\langle P, L^k \rangle, (\langle \phi_i, l_i^k \rangle)_{i \in |\mathsf{I}|})_{k \in \bigcup K_{\mathsf{d}'}}$ defines a multi-limit for d. Its universal property follows from the universal properties of limits in $\mathsf{Spec}^{\mathsf{op}}$, cofree families and respectively multi-limits in $\mathsf{Alg}_D(P)$. This concludes the proof.

Remark 41. Multi-limits of diagrams in Alg_D can alternatively be obtained as limits in $\mathsf{Fam}(\mathsf{Alg}_D)$ of the translations of these diagrams along the embedding of Alg_D into $\mathsf{Fam}(\mathsf{Alg}_D)$ (see Remark 5).

Limits in CoAlg_D and respectively multi-limits in Alg_D provide canonical ways of combining algebras of component specifications into algebras of structured specifications: given algebras of the component specifications, with two such algebras being related by a coextension relation whenever their underlying specifications are related by a specification map, the (multi-)limit construction yields a family of algebras of the combined specification each of whose elements is maximal in the class of algebras that consistently coextend the algebras of the component specifications.

We now show that any algebra of a structured specification can be obtained as the limit object of a diagram in Alg_D whose objects are algebras of the component specifications. This then yields a compositional semantics for structured hidden specifications.

Theorem 42. *Let* $\mathsf{p} : \mathsf{I}^{\mathsf{op}} \to \mathsf{Spec}$ *denote a finite diagram having* $(P, (\phi_i)_{i \in |\mathsf{I}|})$ *as colimit, and let* $A \in \mathsf{Alg}_D(P)$*. Then,* $(\langle P, A\rangle, (\langle \phi_i, 1_{A\restriction_{\mathsf{p}(i)}}\rangle)_{i\in|\mathsf{I}|})$ *is a limit for the diagram* $\mathsf{d} : \mathsf{I} \to \mathsf{Alg}_D$ *defined by:*

- $\mathsf{d}(i) = \langle \mathsf{p}(i), A\restriction_{\mathsf{p}(i)}\rangle$ *for* $i \in |\mathsf{I}|$
- $\mathsf{d}(l) = \langle \mathsf{p}(\bar{l}), 1_{A\restriction_{\mathsf{p}(j)}}\rangle$ *for* $(l : i \to j) \in \|\,\mathsf{I}\,\|$ *(with* $\bar{l} : j \to i$ *denoting the* I^{op} *arrow induced by the* I*-arrow* l*)*

Proof (sketch). To show that $(\langle P, A\rangle, (\langle \phi_i, 1_{A\restriction_{\mathsf{p}(i)}}\rangle)_{i\in|\mathsf{I}|})$ satisfies the universal property of a limit, let $(\langle P', A'\rangle, (\langle \phi'_i, f_i\rangle)_{i\in|\mathsf{I}|})$ denote an arbitrary cone for d. It follows immediately that $(P', (\phi'_i)_{i\in|\mathsf{I}|})$ defines a cocone for p. The universal property of P then yields a unique specification map $\phi' : P \to P'$ satisfying $\phi'_i = \phi' \circ \phi_i$ for any $i \in |\mathsf{I}|$. Also, the homomorphisms $f_i : A'\restriction_{\mathsf{p}(i)} \to A\restriction_{\mathsf{p}(i)}$ (with $i \in |\mathsf{I}|$) uniquely determine a Σ_P-homomorphism $f : A'\restriction_P \to A$ such that $f\restriction_{\mathsf{p}(i)} = f_i$ for each $i \in |\mathsf{I}|$. Then, $\langle \phi', f\rangle : \langle P', A'\rangle \to \langle P, A\rangle$ satisfies $\langle \phi_i, 1_{A\restriction_{\mathsf{p}(i)}}\rangle \circ \langle \phi', f\rangle = \langle \phi'_i, f_i\rangle$ for each $i \in |\mathsf{I}|$ and furthermore, this is the only arrow in $\mathsf{Alg}_D(\langle P', A'\rangle, \langle P, A\rangle)$ with this property. Hence, $(\langle P, A\rangle, (\langle \phi_i, 1_{A\restriction_{\mathsf{p}(i)}}\rangle)_{i\in|\mathsf{I}|})$ is a limit for d in Alg_D.

That is, if P is a structured specification, any P-algebra is obtained as the canonical coextension of its reducts to the components of P. It can also be proved that any Σ_P-homomorphism is obtained from its reducts to the components of P in a similar way. The following compositionality result for structured hidden specifications can then be derived.

Theorem 43. *Let* $\mathsf{p} : \mathsf{I}^{\mathsf{op}} \to \mathsf{Spec}$ *denote a finite diagram having* $(P, (\phi_i : \mathsf{p}(i) \to P)_{i\in|\mathsf{I}|})$ *as a colimit. Then,* $(\mathsf{Alg}_D(P), (\mathsf{U}_{\phi_i} : \mathsf{Alg}_D(P) \to \mathsf{Alg}_D(\mathsf{p}(i)))_{i\in|\mathsf{I}|})$ *is a limit for the diagram* $\mathsf{d} : \mathsf{I} \to \mathsf{Cat}$ *given by:*

- $\mathsf{d}(i) = \mathsf{Alg}_D(\mathsf{p}(i))$ *for* $i \in |\mathsf{I}|$
- $\mathsf{d}(l) = \mathsf{U}_{\mathsf{p}(\bar{l})}$ *for* $(l : i \to j) \in \|\,\mathsf{I}\,\|$.

That is, the semantics of a structured hidden specification can be expressed exclusively in terms of the semantics of its component specifications.

Finally, we note that a version of the above result in which the categories $\mathsf{Alg}_D(P)$ and $\mathsf{Alg}_D(\mathsf{p}(i))$ with $i \in |\mathsf{I}|$ are replaced by $\mathsf{Fam}(\mathsf{Alg}_D(P))$ and respectively $\mathsf{Fam}(\mathsf{Alg}_D(\mathsf{p}(i)))$ can also be formulated.

5 Relation to Other Approaches

This section discusses the relation between hidden algebra and other existing approaches to system specification, based either exclusively on coalgebra or on a combination of algebra and coalgebra, focusing on their expressiveness as well as on their strengths/limitations.

Hidden algebra employs an algebraic syntax to specify system behaviour. However, its semantics is intrinsically coalgebraic: hidden algebraic structures can be regarded as coalgebraic structures, and taking this view yields a canonical characterisation of behavioural equivalence as coalgebraic bisimilarity, as well as the existence of final algebras for hidden signatures of destructors. Furthermore, it is precisely the absence of any purely algebraic features that ensures the existence of final/cofree algebras in the restricted version of hidden algebra considered in [Cîr98]: by ruling out operations with more than one hidden argument, hidden signatures can be regarded as coalgebraic signatures of observers; and by ruling out equations with more than one hidden-sorted variable, the only use of equations is to relate different observations of the same state.

The extension of the results regarding the existence of final/cofree algebras to arbitrary specifications crucially depends on the triviality of the purely algebraic aspects of hidden specifications (such as the use of structured domains for operations, or the use of equations with arbitrarily many hidden-sorted variables). The triviality of the constructor operations is crucial for the existence of final families of algebras, ensuring uniqueness of homomorphisms into (an algebra of) the final family. Also, the restriction to equations with at most one hidden-sorted variable is necessary to guarantee the existence of filtered colimits, and hence of a final family, in the category of hidden algebras of a specification containing constructors. Results similar to the ones presented in Section 3 could not, for instance, be formulated for recent extensions of hidden algebra [Dia98], [RG] that accommodate arbitrary constructors, or for approaches that combine algebraic and coalgebraic concepts without substantially restricting the algebraic ones (such as for instance the approach in [Rei98] based on dialgebras).

As far as the observational aspect of state-based systems is concerned, purely coalgebraic approaches to system specification benefit from greater generality than hidden algebra (where the use of an algebraic syntax prevents operations from having structured results). Particularly worth mentioning in this respect are coalgebraic approaches involving the use of power sets or coproducts in the endofunctors considered, approaches which allow the specification of nondeterministic systems and respectively of systems whose structure is variable. Moreover, exceptions can be naturally handled by approaches involving coproducts: the destructors `empty? : Buffer` $\rightarrow$ `Bool` and `val : Buffer` $\rightarrow$ `Val` used in Example 36 could, for instance, be replaced in such an approach by a single destructor `val : Buffer` $\rightarrow$ `1 + Val`, thus avoiding any redundancy in the information contained in states.

On the other hand, apart from state observers, hidden algebra is also able to accommodate basic state constructors, and to capture the relationship between constructing and observing system states. Moreover, [Dia98], [RG] further increase the generality of (the algebraic aspect of) hidden algebra, by accommodating arbitrary constructors. Finally (and most importantly), hidden algebra benefits from great simplicity and efficiency of proofs, as a result of using a (finitary) algebraic syntax.

A possible limitation of hidden algebra is related to the expressive power of equational approaches to system specification. Such approaches have been shown in [Cor98] to be insufficiently expressive to yield a Birkhoff-style characterisability result for (classes of) coalgebras. This makes them less expressive than, for instance, coalgebraic approaches generalising modal logics (see [Mos]), approaches which, at the expense of using infinitary sentences, are able to provide characterising formulae for states. However, equational sentences appear to be better suited for *concisely* specifying properties quantified over the entire state space of the system being considered, whereas coalgebraic modal logic seems more suitable for characterising individual system states.

Finally, we briefly comment on the difference in handling nondeterminism between hidden algebra and coalgebraic approaches involving power sets. The form of nondeterminism captured by hidden specifications corresponds to *underspecification* (with the behaviour of some of the constructors not being fully determined by specifications, and with algebras resolving the nondeterminism in specifications in particular ways), as opposed to the *true nondeterminism* captured by coalgebraic approaches involving power sets. However, this difference merely reflects a difference in the kinds of systems these approaches aim to specify, namely *active* in the case of coalgebraic approaches involving power sets, and respectively *reactive* in the case of hidden algebra.

6 Conclusions and Future Work

Hidden specifications comprising both algebraic and coalgebraic structure and maps between such specifications have been considered, and final, respectively cofree families of hidden algebras have been shown to provide appropriate denotations for them. A canonical way of constructing algebras for structured specifications from algebras of their component specifications has also been derived. Finally, a compositionality result for structured hidden specifications has been formulated.

The use of an algebraic syntax in conjunction with a coalgebraic semantics restricts the form of constructors and destructors that one can specify in hidden algebra. Other ways of combining algebra and coalgebra for objects should also be investigated, possibly by making the separation between their algebraic and coalgebraic aspects more explicit, in order to allow the specification of more general behaviours.

References

[Bor94] F. Borceux. *Handbook of Categorical Algebra*, volume I. Cambridge University Press, 1994.

[Cîr98] C. Cîrstea. Coalgebra semantics for hidden algebra: parameterised objects and inheritance. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS*. Springer, 1998.

[Cor98] Andrea Corradini. A complete calculus for equational deduction in coalgebraic specification. In F. Parisi-Presicce, editor, *Recent Trends in Algebraic Development Techniques*, volume 1376 of *Lecture Notes in Computer Science*. Springer, 1998.

[Dia98] Răzvan Diaconescu. Behavioural coherence in object-oriented algebraic specification. Technical report, Japan Advanced Institute for Science and Technology, 1998.

[Die79] Y. Diers. Familles universelles de morphismes. *Annales de la Société Scientifique de Bruxelles*, 93(3):175–195, 1979.

[EBO93] H. Ehrig, M. Baldamus, and F. Orejas. New concepts for amalgamation and extension in the framework of specification logics. In M. Bidoit and C. Choppy, editors, *Recent Trends in Data Type Specification*, volume 655 of *LNCS*. Springer, 1993.

[EM85] H. Ehrig and B. Mahr. Fundamentals of algebraic specification 1: Equations and initial semantics. In W. Brauer, G. Rozenberg, and A. Salomaa, editors, *EATCS Monographs on TCS, Volume 6*. Springer, 1985.

[GD94] J. Goguen and R. Diaconescu. Towards an algebraic semantics for the object paradigm. In H. Ehrig and F. Orejas, editors, *Recent Trends in Data Type Specification*, volume 785 of *Lecture Notes in Computer Science*, pages 1–29. Springer, 1994.

[GM97] J. Goguen and G. Malcolm. A hidden agenda. Technical Report CS97-538, UCSD, 1997.

[Gog91] Joseph Goguen. Types as theories. In G. M. Reed, A. W. Roscoe, and R. F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford University Press, 1991.

[GTW78] J. Goguen, J. Thatcher, and E. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R. Yeh, editor, *Current Trends in Programming Methodology. Data Structuring*, volume 4, pages 80–149. Prentice-Hall, 1978.

[Her93] C. Hermida. On fibred adjunctions and completeness for fibred categories. In H. Ehrig and F. Orejas, editors, *Recent Trends in Data Type Specification*, volume 785 of *LNCS*. Springer, 1993.

[Jac96] B. Jacobs. Objects and classes, coalgebraically. In B. Freitag, C. B. Jones, C. Lengauer, and H.-J. Schek, editors, *Object Orientation with Parallelism and Persistence*, pages 83–103. Kluwer Academic Publishers, 1996.

[MG94] G. Malcolm and J. Goguen. Proving correctness of refinement and implementation. Technical Monograph PRG-114, Oxford University, 1994.

[Mos] Lawrence S. Moss. Coalgebraic logic. To appear in the Annals of Pure and Applied Logic.

[Rei98] Horst Reichel. Dialgebraic logics. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Proceedings, Coalgebraic Methods in Computer Science*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 247–255. Elsevier Science, 1998.

[RG] Grigore Roşu and Joseph Goguen. Hidden congruent deduction. To appear in Lecture Notes in Artificial Inteligence, 1999.

[Rut96] J. Rutten. Universal coalgebra: a theory of systems. Technical Report CS-R9652, CWI, 1996.