Reflecting Algebraically Compact Functors

Vladimir Zamdzhiev
Université de Lorraine, CNRS, Inria, LORIA, F 54000 Nancy, France

A *compact* T-algebra is an initial T-algebra whose inverse is a final T-coalgebra. Functors with this property are said to be *algebraically compact*. This is a very strong property used in programming semantics which allows one to interpret recursive datatypes involving mixed-variance functors, such as function space. The construction of compact algebras is usually done in categories with a zero object where some form of a limit-colimit coincidence exists. In this paper we consider a more abstract approach and show how one can construct compact algebras in categories which have neither a zero object, nor a (standard) limit-colimit coincidence by reflecting the compact algebras from categories which have both. In doing so, we provide a *constructive* description of a large class of algebraically compact functors (satisfying a compositionality principle) and show our methods compare quite favorably to other approaches from the literature.

1 Introduction

Inductive datatypes for programming languages can be used to represent important data structures such as lists, trees, natural numbers and many others. When providing a denotational interpretation for such languages, type expressions correspond to functors and one has to be able to construct their initial algebras in order to model inductive datatypes [9]. If the admissible datatype expressions allow only pairing and sum types, then the functors induced by these expressions are all polynomial functors, i.e., functors constructed using only coproducts and (tensor) product connectives, and the required initial algebra may usually be constructed using Adámek's celebrated theorem [2].

However, if one also allows function types as part of the admissible datatype expressions, then we talk about *recursive datatypes* and their denotational interpretation requires additional structure. A solution advocated by Freyd [8] and Fiore and Plotkin [6] is based on *algebraically compact functors*, i.e., functors F which have an initial F-algebra whose inverse is a final F-coalgebra. F-algebras with this property are called *compact* within this paper.

The celebrated limit-colimit coincidence theorem [17] and other similar theorems are usually used for the construction of compact algebras with starting point a zero object of the category where the language is interpreted. However, if one is interested in semantics for mixed linear/non-linear lambda calculi, then it becomes necessary to also solve recursive domain equations within categories that do not have a zero object.

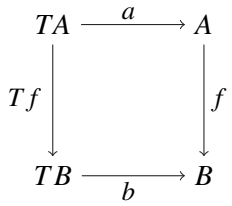
In this paper, we demonstrate how one can construct compact algebras in categories which do not have a zero object and we do so without (explicitly) assuming the existence of any limits or colimits whatsoever. Our methods are based on *enriched* category theory and we show how this allows us to reflect compact algebras from categories with strong algebraic compactness properties into categories without such properties. The results which we present are also compositional and this allows us to provide constructive descriptions of large classes of algebraically compact functors using formal grammars.

John Baez and Bob Coecke (Eds.): Applied Category Theory 2019
EPTCS 323, 2020, pp. 15–23, doi:10.4204/EPTCS.323.2

2 A Reflection Theorem for Algebraically Compact Functors

In this section we show how initial algebras, final coalgebras and compact algebras may be reflected.

Definition 1. *Given an endofunctor $T : \mathbf{C} \to \mathbf{C}$, a* T-algebra *is a pair (A,a), where A is an object of $\mathbf{C}$ and $TA \xrightarrow{a} A$ is a morphism of $\mathbf{C}$. A T-algebra morphism $f : (A,a) \to (B,b)$ is a morphism $f : A \to B$ of $\mathbf{C}$, such that the following diagram:*



commutes. The dual notion is called a T-coalgebra.

Obviously, T-(co)algebras form a category. A T-(co)algebra is initial (final) if it is initial (final) in that category.

Definition 2. *An endofunctor $T : \mathbf{C} \to \mathbf{C}$ is (1)* algebraically complete *if it has an initial T-algebra (2)* algebraically cocomplete *if it has a final T-coalgebra and (3)* algebraically compact *if it has an initial T-algebra $T\Omega \xrightarrow{\omega} \Omega$, such that $T\Omega \xleftarrow{\omega^{-1}} \Omega$ is a final T-coalgebra.*

Next, we recall a lemma first observed by Peter Freyd.

Lemma 3 ([8, pp. 100])**.** *Let $\mathbf{C}$ and $\mathbf{D}$ be categories and $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$ functors. If $GF\Omega \xrightarrow{\omega} \Omega$ is an initial GF-algebra, then $FGF\Omega \xrightarrow{F\omega} F\Omega$ is an initial FG-algebra.*

By dualising the above lemma, we obtain the next one.

Lemma 4. *Let $\mathbf{C}$ and $\mathbf{D}$ be categories and $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$ functors. If $GF\Omega \xleftarrow{\omega} \Omega$ is a final GF-coalgebra, then $FGF\Omega \xleftarrow{F\omega} F\Omega$ is a final FG-coalgebra.*

By using the two lemmas above, the next theorem follows immediately.

Theorem 5. *Let $\mathbf{C}$ and $\mathbf{D}$ be categories and $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$ functors. Then FG is algebraically complete/cocomplete/compact iff GF is algebraically complete/cocomplete/compact, respectively.*

In order to avoid cumbersome repetition, all subsequent results are stated for algebraic compactness. However, all results presented in this section and the next one (excluding Non-Example 29) also hold true when all instances of "algebraic compactness" are replaced with "algebraic completeness" or with "algebraic cocompleteness".

Assumption 6. *Throughout the rest of the paper we assume we are given an arbitrary cartesian closed category $\mathbf{V}(1,\times,\to)$ which we will use as the base of enrichment. $\mathbf{V}$-categories are written using capital calligraphic letters $(\mathscr{C},\mathscr{D},\ldots)$ and their underlying categories using a corresponding bold capital letter $(\mathbf{C},\mathbf{D},\ldots)$. $\mathbf{V}$-functors are also written with calligraphic letters $(\mathscr{F},\mathscr{G} : \mathscr{C} \to \mathscr{D})$ and their underlying functors using a corresponding capital letter $F,G : \mathbf{C} \to \mathbf{D}$.*

Definition 7. *A $\mathbf{V}$-endofunctor $\mathscr{T} : \mathscr{C} \to \mathscr{C}$ is* algebraically compact *if its underlying endofunctor $T : \mathbf{C} \to \mathbf{C}$ is algebraically compact.*

Definition 8 ([6, Definition 5.3])**.** *A $\mathbf{V}$-category $\mathscr{C}$ is* $\mathbf{V}$-algebraically compact *if every $\mathbf{V}$-endofunctor $\mathscr{T} : \mathscr{C} \to \mathscr{C}$ is algebraically compact.*

In particular, a **Set**-algebraically compact category is a locally small category $\mathbf{C}$, such that every endofunctor $T : \mathbf{C} \to \mathbf{C}$ is algebraically compact. In this case we simply say $\mathbf{C}$ is algebraically compact.

Example 9. *Let λ be a cardinal and let $\mathbf{Hilb}_{\lambda}^{\leq 1}$ be the category whose objects are the Hilbert spaces with dimension at most λ and whose morphisms are the linear maps of norm at most 1. Then $\mathbf{Hilb}_{\lambda}^{\leq 1}$ is algebraically compact [3, Theorem 3.2].*

For the next (very important) example, recall that a *complete partial order* (cpo) is a poset such that every increasing chain has a supremum. A cpo is *pointed* if it has a least element. A monotone map $f : X \to Y$ between two cpo's is *Scott-continuous* if it preserves suprema. If, in addition, X and Y are pointed and f preserves the least element of X, then we say that f is *strict*. We denote with **CPO** the category of cpo's and Scott-continuous functions and we denote with $\mathbf{CPO}_{\perp !}$ the category of pointed cpo's and strict Scott-continuous functions. The category **CPO** is cartesian closed, $\mathbf{CPO}_{\perp !}$ is symmetric monoidal closed (when equipped with the smash product and strict function space) and both categories are complete and cocomplete [1]. We will see both categories as **CPO**-categories when equipped with the standard pointwise order on functions.

Therefore, a **CPO**-category ($\mathbf{CPO}_{\perp !}$-category) is simply a category whose homsets have the additional structure of a (pointed) cpo and for which composition is a (strict) Scott-continuous operation in both arguments. A **CPO**-functor ($\mathbf{CPO}_{\perp !}$-functor) then is simply a functor whose action on hom-cpo's is a (strict) Scott-continuous function. The notion of a **CPO**-natural transformation coincides with that of $\mathbf{CPO}_{\perp !}$-natural transformation which also coincides with the ordinary notion. Because of these reasons, it is standard in the programming semantics literature to use the same notation for $\mathbf{CPO}_{(\perp !)}$-enriched categorical notions and their ordinary underlying counterparts. We do the same in this paper.

Example 10. *The category $\mathbf{CPO}_{\perp !}$ is **CPO**-algebraically compact [5, Corollary 7.2.4].*

Next, we show how to reflect algebraically compact **V**-functors.

Definition 11. *We shall say that a **V**-endofunctor $\mathscr{T} : \mathscr{C} \to \mathscr{C}$ has a* **V**-algebraically compact factorisation *if there exists a **V**-algebraically compact category $\mathscr{D}$ and **V**-functors $\mathscr{F} : \mathscr{C} \to \mathscr{D}$ and $\mathscr{G} : \mathscr{D} \to \mathscr{C}$ such that $\mathscr{T} \cong \mathscr{G} \circ \mathscr{F}$.*

Theorem 12. *If a **V**-endofunctor $\mathscr{T} : \mathscr{C} \to \mathscr{C}$ has a **V**-algebraically compact factorisation, then it is algebraically compact.*

Proof. Taking $\mathscr{D}, \mathscr{F}, \mathscr{G}$ as in Definition 11, we get a **V**-endofunctor $\mathscr{F} \circ \mathscr{G} : \mathscr{D} \to \mathscr{D}$. Since $\mathscr{D}$ is **V**-algebraically compact, then its underlying endofunctor $F \circ G : \mathbf{D} \to \mathbf{D}$ is algebraically compact. Theorem 5 shows that $G \circ F : \mathbf{C} \to \mathbf{C}$ is algebraically compact. Algebraic compactness is preserved by natural isomorphisms and therefore $T \cong G \circ F$ is also algebraically compact. □

Using the two examples above, we easily get two corollaries.

Corollary 13. *Any endofunctor $T : \mathbf{Set} \to \mathbf{Set}$ which factors through $\mathbf{Hilb}_{\lambda}^{\leq 1}$ is algebraically compact.*

Corollary 14. *Any **CPO**-endofunctor $T : \mathbf{CPO} \to \mathbf{CPO}$ which factors through $\mathbf{CPO}_{\perp !}$ via a pair of **CPO**-functors, is algebraically compact. Thus the lifting functor $(-)_{\perp} : \mathbf{CPO} \to \mathbf{CPO}$ (given by freely adding a least element) is algebraically compact.*

Note that (ordinary) algebraically compact functors are *not* closed under composition. However, using the additional structure we have introduced, we can prove the following compositionality result.

Proposition 15. *Let $\mathscr{H} : \mathscr{C} \to \mathscr{C}$ be a **V**-endofunctor and $\mathscr{T} : \mathscr{C} \to \mathscr{C}$ be a **V**-endofunctor with a **V**-algebraically compact factorisation. Then $\mathscr{H} \circ \mathscr{T}$ also has a **V**-algebraically compact factorisation and is thus algebraically compact.*

Proof. If $\mathscr{T} \cong \mathscr{G} \circ \mathscr{F}$, then $\mathscr{H} \circ \mathscr{T} \cong (\mathscr{H} \circ \mathscr{G}) \circ \mathscr{F}$. □

3 Constructive Classes of Algebraically Compact Functors

Assumption 16. *Throughout the rest of the section, we assume we are given the following data. A* **V***-category* $\mathscr{C}$*, a* **V***-algebraically compact category* $\mathscr{D}$ *together with* **V***-functors* $\mathscr{F} : \mathscr{C} \to \mathscr{D}$ *and* $\mathscr{G} : \mathscr{D} \to \mathscr{C}$ *and a* **V***-endofunctor* $\mathscr{T} \cong \mathscr{G} \circ \mathscr{F}$*.*

Consider the following grammar:

$$A, B ::= \mathscr{T}X \mid \mathscr{H}(A_1, \ldots, A_n) \tag{1}$$

where X is simply a type variable, n ranges over the natural numbers (including zero) and $\mathscr{H}$ ranges over **V**-functors $\mathscr{H} : \mathscr{C}^n \to \mathscr{C}$. Every such type expression induces a **V**-endofunctor $[\![X \vdash A]\!] : \mathscr{C} \to \mathscr{C}$, defined by:

$$\begin{aligned} [\![X \vdash \mathscr{T}X]\!] &= \mathscr{T} \\ [\![X \vdash \mathscr{H}(A_1, \ldots, A_n)]\!] &= \mathscr{H} \circ \langle [\![X \vdash A_1]\!], \ldots, [\![X \vdash A_n]\!] \rangle \end{aligned}$$

Remark 17. *Since the base of enrichment* **V** *is cartesian, tuples of* **V***-functors, as above, are also* **V***-functors and the above assignment is well-defined. Also,* **V***-algebraically compact categories have been studied only for cartesian* **V***. Because of these two reasons, Assumption 6 cannot be relaxed to a symmetric monoidal closed* **V***.*

Theorem 18. *Any functor* $[\![X \vdash A]\!] : \mathscr{C} \to \mathscr{C}$ *factors through* $\mathscr{F}$ *and is therefore algebraically compact.*

Proof. By induction. For the base case we have $\mathscr{T} \cong \mathscr{G} \circ \mathscr{F}$. The step case is given by

$$\begin{aligned} [\![X \vdash \mathscr{H}(A_1, \ldots, A_n)]\!] &= \mathscr{H} \circ \langle [\![X \vdash A_1]\!], \ldots, [\![X \vdash A_n]\!] \rangle \\ &\cong \mathscr{H} \circ \langle \mathscr{G}_1 \circ \mathscr{F}, \ldots, \mathscr{G}_n \circ \mathscr{F} \rangle \\ &= \mathscr{H} \circ \langle \mathscr{G}_1, \ldots, \mathscr{G}_n \rangle \circ \mathscr{F}, \end{aligned}$$

for some **V**-functors $\mathscr{G}_i : \mathscr{D} \to \mathscr{C}$. □

Example 19. *The* **V***-functor* $\mathscr{T}$ *is algebraically compact.*

Example 20. *Any constant functor* $K_c : \mathbf{C} \to \mathbf{C}$ *is, of course, algebraically compact. This is captured by our theorem, because* K_c *is the underlying functor of the constant c* **V***-endofunctor* $\mathscr{K}_c : \mathscr{C} \to \mathscr{C}$*, which may be constructed using our grammar.*

Example 21. *If* $\mathscr{B}_1 : \mathscr{C} \times \mathscr{C} \to \mathscr{C}$ *and* $\mathscr{B}_2 : \mathscr{C} \times \mathscr{C} \to \mathscr{C}$ *are two* **V***-bifunctors, and* $\mathscr{E} : \mathscr{C} \to \mathscr{C}$ *is a* **V***-endofunctor, then the endofunctors* $\mathscr{E} \circ \mathscr{T}$ *and* $\mathscr{B}_1 \circ \langle \mathscr{T}, \mathscr{T} \rangle$ *and* $\mathscr{B}_2 \circ \langle \mathscr{E} \circ \mathscr{T}, \mathscr{B}_1 \circ \langle \mathscr{T}, \mathscr{T} \rangle \rangle$ *are algebraically compact (among many other combinations).*

3.1 Special Case: Models of Mixed Linear/Non-linear Lambda Calculi

As a special case, our development can be applied to models of mixed linear/non-linear lambda calculi with recursive types, as we shall now explain.

In a **CPO**-category, an *embedding-projection pair* is a pair of morphisms (e, p), such that $e \circ p \leq \mathrm{id}$ and $p \circ e = \mathrm{id}$. The morphism e is called an *embedding* and the morphism p a *projection*. An *e-initial object* is an initial object 0, such that every initial map with it as source is an embedding.

Definition 22. *A model of the linear/nonlinear fixpoint calculus (LNL-FPC) [12] is given by the following data:*

1. *A* **CPO***-symmetric monoidal closed category* **D** *with finite* **CPO***-coproducts, such that* **D** *has an e-initial object and all* ω*-colimits over embeddings;*
2. *A* **CPO***-symmetric monoidal adjunction* $\mathbf{CPO} \underset{G}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{D}$.

In the above situation, the category **D** is necessarily **CPO**-algebraically compact, so it is an ideal setting for constructing compact algebras of **CPO**-functors. We will now show that the monad T of this adjunction also induces a large class of algebraically compact functors on **CPO** (which is not **CPO**-algebraically compact). But first, two examples of the above situation.

Example 23. *The adjunction* $\mathbf{CPO} \underset{U}{\overset{(-)_\bot}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{CPO}_{\bot!}$ *, where the left adjoint is given by domain-theoretic lifting and the right adjoint is the forgetful functor, has the required structure. The induced monad* $T : \mathbf{CPO} \to \mathbf{CPO}$ *is called* lifting *(see Corollary 14). This adjunction is in fact a computationally adequate model of LNL-FPC [12].*

Example 24. *Let* $\mathbf{M}$ *be a small* $\mathbf{CPO}_{\bot!}$*-symmetric monoidal category and let* $\widehat{\mathbf{M}} = [\mathbf{M}^{\mathrm{op}}, \mathbf{CPO}_{\bot!}]$ *be the indicated* $\mathbf{CPO}_{\bot!}$*-functor category. There exists an adjunction* $\mathbf{CPO}_{\bot!} \underset{\widehat{\mathbf{M}}(I,-)}{\overset{-\circledcirc I}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \widehat{\mathbf{M}}$ *, where the left adjoint is the* $\mathbf{CPO}_{\bot!}$*-copower with the tensor unit* I *and the right adjoint is the representable functor (see [4, §6]). Composing the two adjunctions* $\mathbf{CPO} \underset{U}{\overset{(-)_\bot}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{CPO}_{\bot!} \underset{\widehat{\mathbf{M}}(I,-)}{\overset{-\circledcirc I}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \widehat{\mathbf{M}}$ *yields a LNL-FPC model. By making suitable choices for* $\mathbf{M}$*, this data also becomes a model of Proto-Quipper-M, a quantum programming language [16] and also a model of ECLNL, a programming language for string diagrams [10, 11].*

Since **D** is **CPO**-algebraically compact, we can now construct a large class of algebraically compact functors via Theorem 18. For instance, such a subclass is given by the following corollary.

Corollary 25. *Any endofunctor on* **CPO** *constructed using constants,* T*,* $\times$ *and* $+$*, and such that all occurrences of the functorial variable in its definition are surrounded by* T*, is algebraically compact.*

Remark 26. *To make this more precise, one should specify a formal grammar like* (1) *to indicate the admissible functorial expressions, but it should be clear that* (1) *can be easily specialised to handle this.*

Next, let us consider some example endofunctors on **CPO**.

Example 27. *The endofunctor* $H(X) = TX + TX$ *is algebraically compact. Indeed, observe that* $H = + \circ \langle T, T \rangle = [\![X \vdash TX + TX]\!]$.

Example 28. *The endofunctor* $H(X) = TX + T(TX \times TX)$ *is algebraically compact. To see it, observe that* $H = + \circ \langle T, T \circ \times \circ \langle T, T \rangle \rangle = [\![X \vdash TX + T(TX \times TX)]\!]$.

Non-Example 29. *The endofunctor* $H(X) = X \times TX$ *is not algebraically compact (its initial algebra is* $\varnothing \times T\varnothing = \varnothing \xrightarrow{id} \varnothing$ *). Our results do not apply to it, because the left occurrence of* X *does not have* T *applied to it. For the same reason, the identity functor* $Id(X) = X$ *is also not algebraically compact and not covered by our development.*

4 Algebraically Compact Mixed-Variance Functors

As mentioned in the introduction, algebraic compactness allows us to model recursive datatypes which include mixed-variance functors such as function space. In this section we show that our methods are also compatible with recursive datatypes.

Consider a mixed-variance bifunctor $H : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{C}$. Since H is not an endofunctor, then clearly we cannot talk about H-algebras or H-coalgebras. A more appropriate notion is that of a *H-dialgebra*, which we will not introduce here, because of a lack of space and because the category of H-dialgebras is isomorphic to the category of $\breve{H}$-algebras [7, §4], where

$$\breve{H} = \langle H^{\mathrm{op}} \circ \langle \Pi_2, \Pi_1 \rangle, H \rangle : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{C}^{\mathrm{op}} \times \mathbf{C}.$$

Because of this, it is standard to model recursive datatypes as endofunctors $\breve{H} : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{C}^{\mathrm{op}} \times \mathbf{C}$ [6].

If a category $\mathscr{D}$ is $\mathbf{V}$-algebraically complete, then $\mathscr{D}^{\mathrm{op}}$ is $\mathbf{V}$-algebraically cocomplete and vice versa. Thus, $\mathbf{V}$-algebraic compactness is a self-dual notion. Unlike the previous sections, the results presented here do not hold for algebraically complete or cocomplete functors and categories.

If a category $\mathscr{D}$ is $\mathbf{V}$-algebraically compact in a parameterised sense, then so is $\mathscr{D}^{\mathrm{op}} \times \mathscr{D}$. We omit the details of parameterised algebraic compactness, but the interested reader may consult [6]. We point out that the notions of **CPO**-algebraic compactness and parameterised **CPO**-algebraic compactness coincide [5, Corollary 7.2.5] and we shall consider such a **CPO**-example to illustrate our methods. But we emphasise that our methods can be adapted to the general setting of a parameterised $\mathbf{V}$-algebraically compact category $\mathscr{D}$.

Let us assume we are given an LNL-FPC model $\mathbf{CPO} \underset{G}{\overset{F}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{D}$ as in Subsection 3.1 with $T = G \circ F$. In this situation, the category $\mathbf{D}^{\mathrm{op}} \times \mathbf{D}$ is also **CPO**-algebraically compact and we can thus reuse Theorem 12 and Proposition 15, where we choose the **CPO**-algebraically compact factorisation $T^{\mathrm{op}} \times T = (G^{\mathrm{op}} \times G) \circ (F^{\mathrm{op}} \times F)$.

Consider the following grammar:

$$A, B ::= c \mid TX \mid HA \mid A + B \mid A \times B \mid A \to B, \tag{2}$$

where c ranges over the objects of **CPO** and H ranges over **CPO**-endofunctors on **CPO**. Every such type expression induces a **CPO**-endofunctor $[\![X \vdash A]\!] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO}$, defined by:

$$\begin{aligned}
[\![X \vdash TX]\!] &= T^{\mathrm{op}} \times T \\
[\![X \vdash c]\!] &= K_{(c,c)} \\
[\![X \vdash HA]\!] &= (H^{\mathrm{op}} \times H) \circ [\![X \vdash A]\!] \\
[\![X \vdash A + B]\!] &= (+ \circ \langle \Pi_2 [\![X \vdash A]\!], \Pi_2 [\![X \vdash B]\!] \rangle)^{\smile} \\
[\![X \vdash A \times B]\!] &= (\times \circ \langle \Pi_2 [\![X \vdash A]\!], \Pi_2 [\![X \vdash B]\!] \rangle)^{\smile} \\
[\![X \vdash A \to B]\!] &= ([- \to -] \circ \langle \Pi_1 [\![X \vdash A]\!], \Pi_2 [\![X \vdash B]\!] \rangle)^{\smile},
\end{aligned}$$

where $K_{(c,c)}$ is the constant (c,c) endofunctor on $\mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO}$ and $[- \to -] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}$ is the internal-hom.

Remark 30. *The last three cases in the above assignment are essentially the same as the standard interpretation of types within FPC [6, Definition 6.2].*

Theorem 31. *Every functor* $[\![X \vdash A]\!] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO}$ *factors through* $F^{\mathrm{op}} \times F$ *and is therefore algebraically compact.*

Proof. Simple proof by induction. The first three cases are obvious. For the last three cases, simply use the fact that $(H \circ (F^{\mathrm{op}} \times F))^{\smile} = \overset{\smile}{H} \circ (F^{\mathrm{op}} \times F)$, which can be proved after recognising that $(-)^{\mathrm{op}}$ is a *covariant* operation with respect to functor composition. □

Example 32. *Consider the functor* $H(X,Y) = [TX \to TY] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}$*. Then the functor* $\overset{\smile}{H} : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO}$ *is algebraically compact, because:*

$$\begin{aligned}\overset{\smile}{H} = ([- \to -] \circ (T^{\mathrm{op}} \times T))^{\smile} &= ([- \to -] \circ \langle \Pi_1, \Pi_2 \rangle \circ (T^{\mathrm{op}} \times T))^{\smile} \\ &= ([- \to -] \circ \langle \Pi_1 \circ (T^{\mathrm{op}} \times T), \Pi_2 \circ (T^{\mathrm{op}} \times T) \rangle)^{\smile} \\ &= [\![X \vdash TX \to TX]\!].\end{aligned}$$

Non-Example 33. *Consider the internal-hom functor* $[- \to -] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}$*. Then* $[- \overset{\smile}{\to} -]$ *is not algebraically compact, because its initial algebra is given by*

$$\left([- \overset{\smile}{\to} -](1,\varnothing) = ([\varnothing \to 1],[1 \to \varnothing]) = (1,\varnothing)\right) \xrightarrow{id} (1,\varnothing),$$

which is not its final coalgebra. Our results do not apply to $[- \overset{\smile}{\to} -]$*, because* T *does not occur anywhere in its definition.*

5 Comparison with Limit-Colimit Coincidence Results

The focus in this paper is to study algebraically compact endofunctors on categories which do not necessarily have a zero object. In [3] Michael Barr considers this situation and he presents a more general version of the standard limit-colimit coincidence theorem [17]. The increased generality allows him to establish the existence of algebraically compact endofunctors on categories that do not have a zero object. In this section, we will compare his results about **CPO**-categories with ours.

Theorem 34 ([3, Theorem 5.4])**.** *Let* $\mathbf{C}$ *be a* **CPO***-category with initial object* $\varnothing$ *and terminal object* 1*. Assume further* $\mathbf{C}$ *has colimits of initial sequences of* **CPO***-endofunctors. Then the class of* **CPO***-endofunctors for which there is a morphism* $l : 1 \to H\varnothing$ *such that* $\left(H1 \to 1 \xrightarrow{l} H\varnothing \xrightarrow{Hh} H1\right) \leq id_{H1}$*, where* $h : \varnothing \to 1$ *is the unique arrow, is algebraically compact.*

First, a necessary condition in the above situation.

Proposition 35. *In the situation of Theorem 34, the hom-cpo* $\mathbf{C}(H1,H1)$ *is pointed.*

Proof. Let $\bot = \left(H1 \to 1 \xrightarrow{l} H\varnothing \xrightarrow{Hh} H1\right)$. Let $f : H1 \to H1$ be an arbitrary morphism. Then

$$\bot = \bot \circ f \leq \mathrm{id} \circ f = f.$$

□

We may now see that Barr's theorem does not behave well when dealing with constant functors or with functors involving coproducts.

Example 36. *Consider the constant functor $K_2 : \mathbf{CPO} \to \mathbf{CPO}$ where 2 is any two point cpo equipped with the discrete order. As we explained in Example 20, our development captures the fact that K_2 is algebraically compact. However, Barr's theorem does not show this, because $\mathbf{CPO}(2,2)$ is not pointed.*

Example 37. *Consider the functor $H(X) = X_\perp + X_\perp : \mathbf{CPO} \to \mathbf{CPO}$ where $(-)_\perp$ is given by lifting. Our development showed in Example 27 that this functor is algebraically compact. However, Barr's theorem does not show this, because $\mathbf{CPO}(1_\perp + 1_\perp, 1_\perp + 1_\perp)$ is not pointed.*

A natural question to ask is whether there exists an algebraically compact functor described by Theorem 34, but not captured by the methods presented here. We leave this for future work.

We also provided a compositionality result (Proposition 15) which then allowed us to present a *constructive* description of large classes of algebraically compact functors (Section 3). So far this has not been done with Barr's results.

6 Related Work

The solution of recursive domain equations is based on the construction of initial algebras [2, 9] and on the construction of compact algebras when mixed-variance functors are involved [5, 6, 7, 8]. The limit-colimit coincidence theorem [17] for **CPO**-enriched categories with sufficient structure is perhaps the most common way of constructing such compact algebras. In this paper we have focused on the construction of compact algebras within categories with little structure that does not admit utilising the above mentioned approaches. Our motivation for doing this is to consider denotational interpretations of mixed linear/non-linear recursive types.

Another approach for modelling mixed linear/non-linear recursive types is described in [12, 13] where the authors interpret non-linear types within a carefully constructed subcategory of **CPO**. That method works only for **CPO**-categories whereas the techniques presented here work for arbitrary **V**-categories. Also, the set of type expressions that can be interpreted with the methods from [12, 13] is incomparable with the one presented here (neither is a subset of the other). However, the main idea in [12, 13] is to reflect the initial algebra structure from certain (sub)categories and not necessarily the compact algebra structure. This method has found further applications in constructing denotational models for quantum programming [14, 15] and for affine type systems [18].

7 Conclusion

We established new results about algebraically compact functors without relying on limits, colimits or their coincidence. We arrived at these results in a more abstract way by observing that any enriched endofunctor is algebraically compact, provided that it factors through a category which is algebraically compact in an enriched sense. This then allowed us to establish large classes of algebraically compact functors which also admit a constructive description. Our results are compositional and nicely complement other existing approaches in the literature which do rely on a limit-colimit coincidence.

Acknowledgements. The author wishes to thank the anonymous reviewers for their feedback and he gratefully acknowledges financial support from the French projects ANR-17-CE25-0009 SoftQPro and PIA-GDN/Quantex.

References

[1] S. Abramsky & A. Jung (1994): *Domain Theory*. *Handbook of Logic in Computer Science (Vol. 3)*, pp. 1–168. Available at http://dl.acm.org/citation.cfm?id=218742.218744.

[2] Jiří Adámek (1974): *Free algebras and automata realizations in the language of categories*. *Commentationes Mathematicae Universitatis Carolinae* 15(4), pp. 589–602.

[3] M. Barr (1992): *Algebraically compact functors*. *Journal of Pure and Applied Algebra* 82(3), pp. 211 – 231, doi:10.1016/0022-4049(92)90169-G.

[4] F. Borceux (1994): *Handbook of Categorical Algebra 2: Categories and Structures*. Cambridge University Press, doi:10.1017/CBO9780511525865.

[5] M. P. Fiore (1994): *Axiomatic domain theory in categories of partial maps*. Ph.D. thesis, University of Edinburgh, UK.

[6] Marcelo Fiore & Gordon Plotkin (1994): *An Axiomatization of Computationally Adequate Domain Theoretic Models of FPC*. In: *LICS*, doi:10.1109/LICS.1994.316083.

[7] P. Freyd (1990): *Recursive types reduced to inductive types*. In: *LICS 1990*, pp. 498–507, doi:10.1109/LICS.1990.113772.

[8] P. Freyd (1991): *Algebraically complete categories*. In: *Category Theory: Proceedings of the International Conference held in Como, Italy*, doi:10.1007/BFb0084215.

[9] Daniel J Lehmann & Michael B Smyth (1981): *Algebraic specification of data types: A synthetic approach*. *Mathematical Systems Theory*, doi:10.1007/BF01752392.

[10] Bert Lindenhovius, Michael Mislove & Vladimir Zamdzhiev (2018): *Enriching a Linear/Non-linear Lambda Calculus: A Programming Language for String Diagrams*. In: *LICS 2018*, ACM, doi:10.1145/3209108.3209196.

[11] Bert Lindenhovius, Michael Mislove & Vladimir Zamdzhiev (2020): *Semantics for a Lambda Calculus for String Diagrams*. Available at https://homepages.loria.fr/VZamdzhiev/papers/lambda-calculus-string-diagrams.pdf. Preprint.

[12] Bert Lindenhovius, Michael W. Mislove & Vladimir Zamdzhiev (2019): *Mixed linear and non-linear recursive types*. *Proc. ACM Program. Lang.* 3(ICFP), pp. 111:1–111:29, doi:10.1145/3341715.

[13] Bert Lindenhovius, Michael W. Mislove & Vladimir Zamdzhiev (2020): *LNL-FPC: The Linear/Non-linear Fixpoint Calculus*. Available at http://arxiv.org/abs/1906.09503. Preprint.

[14] Romain Péchoux, Simon Perdrix, Mathys Rennela & Vladimir Zamdzhiev (2020): *Quantum Programming with Inductive Datatypes*. Available at https://homepages.loria.fr/VZamdzhiev/papers/qpl-inductive.pdf. Preprint.

[15] Romain Péchoux, Simon Perdrix, Mathys Rennela & Vladimir Zamdzhiev (2020): *Quantum Programming with Inductive Datatypes: Causality and Affine Type Theory*. In Jean Goubault-Larrecq & Barbara König, editors: *Foundations of Software Science and Computation Structures - 23rd International Conference, FOSSACS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings*, *Lecture Notes in Computer Science* 12077, Springer, pp. 562–581, doi:10.1007/978-3-030-45231-5_29.

[16] Francisco Rios & Peter Selinger (2017): *A categorical model for a quantum circuit description language*. In: *QPL 2017*, pp. 164–178, doi:10.4204/EPTCS.266.11.

[17] M.B. Smyth & G.D. Plotkin (1982): *The Category-theoretic Solution of Recursive Domain Equations*. *Siam J. Comput.*, doi:10.1137/0211062.

[18] Vladimir Zamdzhiev (2020): *Semantics for first-order affine inductive data types via slice categories*. In: *Coalgebraic Methods in Computer Science, to appear*. Available at https://arxiv.org/abs/2001.06905.