Reflecting Algebraically Compact Functors

Vladimir Zamdzhiev

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

Applied Category Theory
University of Oxford
17 July 2019

Introduction

- This talk is about categorical semantics of inductive / recursive types.
- (Inductive datatypes $\iff$ polynomial functors) can be modelled by initial algebras.
- (Recursive datatypes $\iff$ mixed-variance functors) can be modelled by compact algebras, i.e., initial algebras whose inverse is a final coalgebra.
- The known constructions of compact algebras are based on limit-colimit coincidence results.
- In this talk we present a more abstract method for their construction.
- Application in semantics for mixed linear/non-linear type systems.

Background: Initial and final (co)algebras

Definition

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:

$$
\begin{tikzcd}
TA \arrow[r, "a"] \arrow[d, "Tf"'] & A \arrow[d, "f"] \\
TB \arrow[r, "b"'] & B
\end{tikzcd}
$$

- The dual notion is called a T-coalgebra.
- T-(co)algebras form a category.
- A T-(co)algebra is initial (final) if it is initial (final) in that category.

Initial and final (co)algebras

Theorem (Lambek)

If (TA, a) is an initial (final) T-(co)algebra, then a is an isomorphism.

Theorem (Adámek)

Let $T : \mathbf{C} \to \mathbf{C}$ be an endofunctor. Assume that the colimit of the initial sequence of T:

$$\varnothing \xrightarrow{\iota} T\varnothing \xrightarrow{T\iota} T^2\varnothing \xrightarrow{T^2\iota} \cdots$$

exists and is preserved by T. Then T has an initial T-algebra.

Theorem (coAdámek)

Let $T : \mathbf{C} \to \mathbf{C}$ be an endofunctor. Assume that the limit of the final sequence of T:

$$1 \xleftarrow{\iota} T1 \xleftarrow{T\iota} T^2 1 \xleftarrow{T^2\iota} \cdots$$

exists and is preserved by T. Then T has a final T-coalgebra.

Categorical Semantics of Inductive Datatypes

- *Inductive datatypes* are an important programming concept.
  - Data structures such as natural numbers, lists, trees, etc.
- Type expressions made from constants, $\otimes$ and $+$ (polynomial endofunctors).
- In *programming semantics* inductive datatypes are modelled via initial algebras.

Example

- Natural numbers are defined by the type expression $\mathrm{Nat} \equiv \mu X.I + X$.
- To interpret it, we need an object $[\![\mathrm{Nat}]\!] \cong I + [\![\mathrm{Nat}]\!]$.
- Consider the functor $T(X) = I + X : \mathbf{C} \to \mathbf{C}$.
- Solution: $[\![\mu X.I + X]\!] := Y(T)$, the carrier of the initial algebra of T.

Categorical Semantics of Recursive Datatypes

- *Recursive datatypes* also allow type expressions involving function space.
  - *Lazy* datatypes, such as streams.
  - Example: $\mu X.1 \to \mathrm{Nat} \times X$, a stream of natural numbers (in a non-linear setting).
- Type expressions made from constants, $\otimes, +, \multimap$ (and possibly ! in linear settings).
- The semantic treatment is considerably more complicated and requires additional structure.
- One approach is based on *algebraic compactness*, i.e., the property of a functor to have an initial algebra whose inverse is a final coalgebra.
- Under some reasonable conditions, this property carries over to endofunctors $T : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \to \mathbf{C}^{\mathrm{op}} \times \mathbf{C}$ which allows one to interpret recursive types.

Algebraic Compactness

Definition

An endofunctor $T : \mathbf{C} \to \mathbf{C}$ is

- *algebraically complete* if it has an initial T-algebra;
- *algebraically cocomplete* if it has a final T-coalgebra;
- *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. We say ω is a *compact* T-algebra.

Definition

A category $\mathbf{C}$ is *algebraically compact* if every endofunctor $T : \mathbf{C} \to \mathbf{C}$ is algebraically compact.

Compact algebra constructions in the literature

Problem

How can one construct compact algebras?

Solution

Require that the initial and final sequences of a functor coincide (limit-colimit coincidence).

Example

The terminal category $\mathbf{1}$ is algebraically compact.

Example (Barr)

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.

Enriched Algebraic Compactness

There are a few issues with algebraic compactness as presented:

- Very few known algebraically compact categories.
- Algebraically compact functors do not compose.

Solution

Consider a class of algebraically compact functors which is well-behaved. Usually, in an enriched sense.

Definition

Given a $\mathbf{V}$-category $\mathcal{C}$, a $\mathbf{V}$-functor $\mathcal{T} : \mathcal{C} \to \mathcal{C}$ is algebraically compact if its underlying functor $T : \mathbf{C} \to \mathbf{C}$ is algebraically compact.

Definition

A $\mathbf{V}$-category $\mathcal{C}$ is $\mathbf{V}$-algebraically compact if every $\mathbf{V}$-endofunctor acting on it is algebraically compact.

Domain Theory

- A *complete partial order* (cpo) is a poset where every increasing chain has a supremum.
- A *pointed* cpo is a cpo with a least element.
- A (strict) Scott-continuous function $f : X \to Y$ between two (pointed) cpo's is a monotone function which preserves suprema of chains (and the least element).
- **CPO** is the category of cpo's and Scott-continuous functions. It is complete, cocomplete and cartesian closed.
- $\mathbf{CPO}_{\perp !}$ is the category of pointed cpo's and strict Scott-continuous functions. It is complete, cocomplete and symmetric monoidal closed.

Order-enriched Category Theory

- $\mathbf{CPO}$-enriched and $\mathbf{CPO}_{\perp!}$-enriched categories are often used in programming semantics to interpret recursion and recursive types.
- A $\mathbf{CPO}_{(\perp!)}$-category $\mathbf{C}$ is a category where $\mathbf{C}(A, B)$ is a (pointed) cpo and where $(- \circ -) : \mathbf{C}(B, C) \times \mathbf{C}(A, B) \to \mathbf{C}(A, C)$ is a (strict) Scott-continuous function.
- A $\mathbf{CPO}_{(\perp!)}$-functor $T : \mathbf{C} \to \mathbf{D}$ is a functor whose action on hom-cpo's $T_{A,B} : \mathbf{C}(A, B) \to \mathbf{D}(TA, TB)$ is a (strict) Scott-continuous function.
- In a $\mathbf{CPO}$-category $\mathbf{C}$, an *embedding* is a morphism $e : A \to B$, for which there exists a (necessarily unique) morphism $p : B \to A$, called a *projection*, such that $p \circ e = \mathrm{id}$ and $e \circ p \leq \mathrm{id}$.

The limit-colimit coincidence theorem

A classical result in domain theory (see [Smyth & Plotkin 1982] and [Fiore & Plotkin 1994]):

Theorem

Let $\mathbf{C}$ *be a* $\mathbf{CPO}$*-category with* ω*-colimits (over embeddings) and a zero object* 0 *such that each* $e : 0 \to A$ *is an embedding. Then* $\mathbf{C}$ *is* $\mathbf{CPO}$*-algebraically compact.*

Example

The category $\mathbf{CPO}_{\perp !}$ is $\mathbf{CPO}$-algebraically compact.

Many other examples in semantics.

Semantics for mixed linear/non-linear type systems

- To interpret mixed linear/non-linear recursive types, one also has to provide an interpretation within a cartesian closed category.
- Existing methods for the construction of compact algebras do not work well in CCCs.
- This talk: we address this issue.

A Reflection Theorem for Algebraically Compact Functors

Lemma (Freyd)

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.*

Lemma (coFreyd)

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.*

Theorem

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.*

A factorisation result

Definition

A $\mathbf{V}$-endofunctor $\mathcal{T} : \mathcal{C} \to \mathcal{C}$ has a $\mathbf{V}$-*algebraically compact factorisation* if there exists a $\mathbf{V}$-algebraically compact category $\mathcal{D}$ and $\mathbf{V}$-functors $\mathcal{F} : \mathcal{C} \to \mathcal{D}$ and $\mathcal{G} : \mathcal{D} \to \mathcal{C}$ such that $\mathcal{T} \cong \mathcal{G} \circ \mathcal{F}$.

Theorem

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

Corollary

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

Corollary

Any $\mathbf{CPO}$*-endofunctor* $T : \mathbf{CPO} \to \mathbf{CPO}$ *which factors through a* $\mathbf{CPO}$*-algebraically compact category (like* $\mathbf{CPO}_{\perp !}$*) in an enriched sense, is algebraically compact. Thus the lifting functor* $(-)_{\perp} : \mathbf{CPO} \to \mathbf{CPO}$ *is algebraically compact.*

A compositionality principle

Proposition

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

A couple of notes

- Most results are stated for algebraic compactness, but many of them also hold for algebraic completeness / cocompleteness.
- For the next slide, consider a model of a mixed linear/non-linear lambda calculus with recursive types. It is given by the following data:
- A **CPO**-algebraically compact category **D**;
- A **CPO**-symmetric monoidal adjunction $\begin{tikzcd} \mathbf{CPO} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$.
- A bit more structure which is irrelevant for this talk.
- Let $T := G \circ F : \mathbf{CPO} \to \mathbf{CPO}$.

An application of the theory

Consider the following formal grammar:

$$A, B ::= c \mid TX \mid HA \mid A + B \mid A \times B \mid A \to B$$

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

Theorem

Every $[\![X \vdash A]\!] : \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO} \to \mathbf{CPO}^{\mathrm{op}} \times \mathbf{CPO}$ *is algebraically compact.*

Remark

The above result also holds when $\mathbf{CPO}$ *is replaced with a CCC* $\mathbf{V}$ *and where* $\mathbf{D}$ *is parameterised* $\mathbf{V}$*-algebraically compact.*

Conclusion

- New method for establishing algebraic completeness/cocompleteness/compactness which does not rely on limits, colimits or their coincidence.
- Simple compositionality principle.
- Applications for semantics of mixed linear/non-linear type systems with inductive/recursive datatypes.
- Easy to establish *constructive* classes of algebraically compact functors with the new method.
- The new method nicely complements other approaches from the literature.

Thank you for your attention!