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

$$
\begin{array}{c}
TA \xrightarrow{a} A \\
\downarrow Tf \\
TB \xrightarrow{b} B \\
\end{array}
$$

• 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 \(T : C \rightarrow C\) be an endofunctor. Assume that the colimit of the initial sequence of \(T\):

\[
\emptyset \xrightarrow{\iota} T \emptyset \xrightarrow{T\iota} T^2 \emptyset \xrightarrow{T^2\iota} \ldots
\]

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

Theorem (coAdámek)

Let \(T : C \rightarrow C\) be an endofunctor. Assume that the limit of the final sequence of \(T\):

\[
1 \xleftarrow{\iota} T1 \xleftarrow{T\iota} T^2 1 \xleftarrow{T^2\iota} \ldots
\]

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 $\mathbb{Nat} \equiv \mu X. I + X$.
• To interpret it, we need an object $\llbracket \mathbb{Nat} \rrbracket \cong I + \llbracket \mathbb{Nat} \rrbracket$.
• Consider the functor $T(X) = I + X : C \to C$.
• Solution: $\llbracket \mu X. I + X \rrbracket := Y(T)$, the carrier of the initial algebra of $T$. 

Recursive datatypes also allow type expressions involving function space.

- Lazy datatypes, such as streams.
- Example: $\mu X.1 \rightarrow \text{Nat} \times X$, a stream of natural numbers (in a non-linear setting).

Type expressions made from constants, $\otimes$, $+$, $\rightarrow$ (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 $T : C^{\text{op}} \times C \rightarrow C^{\text{op}} \times C$ which allows one to interpret recursive types.
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 $\omega$ 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 $\lambda$ be a cardinal and let $\text{Hilb}_{\lambda}^{\leq 1}$ be the category whose objects are the Hilbert spaces with dimension at most $\lambda$ and whose morphisms are the linear maps of norm at most 1. Then $\text{Hilb}_{\lambda}^{\leq 1}$ 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 $T : \mathcal{C} \to \mathcal{C}$ is algebraically compact if its underlying functor $T : \mathcal{C} \to \mathcal{C}$ 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.
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 $f : X \rightarrow Y$ 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.

CPO⊥ is the category of pointed cpo’s and strict Scott-continuous functions. It is complete, cocomplete and symmetric monoidal closed.
Order-enriched Category Theory

- **CPO-enriched** and **CPO\(\perp\!\)-enriched** categories are often used in programming semantics to interpret recursion and recursive types.

- A **CPO\(\perp\!\)-category** \(C\) is a category where \(C(A, B)\) is a (pointed) cpo and where \((-\circ-) : C(B, C) \times C(A, B) \to C(A, C)\) is a (strict) Scott-continuous function.

- A **CPO\(\perp\!\)-functor** \(T : C \to D\) is a functor whose action on hom-cpo’s \(T_{A,B} : C(A, B) \to D(TA, TB)\) is a (strict) Scott-continuous function.

- In a **CPO-category** \(C\), 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 \(p \circ e = \text{id}\) and \(e \circ p \leq \text{id}\).
The limit-colimit coincidence theorem

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

**Theorem**

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

**Example**

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

Many other examples in semantics.
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 $GF\Omega \xrightarrow{\varphi} \Omega$ is an initial $GF$-algebra, then $FGF\Omega \xrightarrow{F\varphi} F\Omega$ 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 $GF\Omega \xleftarrow{\varphi} \Omega$ is a final $GF$-coalgebra, then $FGF\Omega \xleftarrow{F\varphi} F\Omega$ 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 $T : \mathcal{C} \to \mathcal{C}$ 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 $T \cong \mathcal{G} \circ \mathcal{F}$.

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

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

Corollary
Any $\text{CPO}$-endofunctor $T : \text{CPO} \to \text{CPO}$ which factors through a $\text{CPO}$-algebraically compact category (like $\text{CPO}_{\bot!}$) in an enriched sense, is algebraically compact. Thus the lifting functor $(-)_{\bot} : \text{CPO} \to \text{CPO}$ is algebraically compact.
A compositionality principle

Proposition

Let $\mathcal{H} : C \to C$ be a $\mathbf{V}$-endofunctor and $\mathcal{T} : C \to C$ 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 $\text{CPO} \xleftarrow{F} \xrightarrow{G} D$.
- A bit more structure which is irrelevant for this talk.
- Let $T := G \circ F : \text{CPO} \to \text{CPO}$.
An application of the theory

Consider the following formal grammar:

\[
A, B ::= c \mid TX \mid HA \mid A + B \mid A \times B \mid A \rightarrow B
\]

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 

\[
\exists X \vdash A : \mathbf{CPO}^{\text{op}} \times \mathbf{CPO} \rightarrow \mathbf{CPO}^{\text{op}} \times \mathbf{CPO},
\]

when interpreted in the standard way.

Theorem
Every \(\exists X \vdash A : \mathbf{CPO}^{\text{op}} \times \mathbf{CPO} \rightarrow \mathbf{CPO}^{\text{op}} \times \mathbf{CPO}\) is algebraically compact.

Remark
The above result also holds when \(\mathbf{CPO}\) is replaced with a \(\mathbf{CCC} \mathbf{V}\) and where \(D\) 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!