What is the Category for Haskell?

Makoto Hamana

IPL, 2007, September
We roughly believe ... 

- not just \textbf{Set}

- maybe \textbf{CPO}, due to lazyness, and mixed variance
  \((D \cong D \Rightarrow D)\)

- Folklore: initial algebra = finial coalgebra
  e.g. List datatype in Haskell is given by an initial algebra but
  contains all infinite lists

- But: there are subtle \textcolor{orange}{\textit{categorical}} differences in several variations
  of cpos

- Clarify these!
We clarify ...

▷ why coincidence?
  - which is a suitable category of cpos?

▷ why infinite data?

▷ how can we give a datatype functor?
  - a non-strict continuous function given by a strict function

▷ what is Haskell’s algebraic datatype?
Categories of CPOs

Def.
A cpo is a poset such that every $\omega$-chain has a lub.

NOTE: it may not have $\perp$.

Consider slightly different categories: Cpo, Cppo, Cppo$_\perp$
All have cpos as objects.
<table>
<thead>
<tr>
<th>Category</th>
<th>Cpo</th>
<th>Cppo</th>
<th>Cppo⊥</th>
</tr>
</thead>
<tbody>
<tr>
<td>bottom?</td>
<td>no</td>
<td>⊥</td>
<td>⊥</td>
</tr>
<tr>
<td>functions cont.</td>
<td>cont.</td>
<td>cont.</td>
<td>strict</td>
</tr>
<tr>
<td>$D \times E$</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>$[D \to E]$</td>
<td>+</td>
<td>+</td>
<td>✓</td>
</tr>
<tr>
<td>$D + E$</td>
<td>+</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Initial object</td>
<td>∅</td>
<td>no</td>
<td>{⊥}</td>
</tr>
<tr>
<td>Terminal object</td>
<td>{*}</td>
<td>{⊥}</td>
<td>{⊥}</td>
</tr>
<tr>
<td>$D \otimes E$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$[D \to ⊥ E]$</td>
<td>✓</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>$D \oplus E$</td>
<td>✓</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>closed structure</td>
<td>ccc</td>
<td>ccc</td>
<td>monoidal closed</td>
</tr>
<tr>
<td>(co)completeness</td>
<td>✓</td>
<td>no</td>
<td>✓</td>
</tr>
<tr>
<td>i.alg.⇒final coalg.</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
General Construction of Initial Algebras

**Thm. (Basic Lemma [Smyth-Plotkin])**

Let $\mathcal{C}$ be a category with initial object $0$, and $F : \mathcal{C} \rightarrow \mathcal{C}$ a functor. If $F$ preserves the colimit of the $\omega$-chain $D$

$$
0 \xrightarrow{!} F(0) \xrightarrow{F!} F^2(0) \xrightarrow{F^2!} \cdots
$$

then, the **initial $F$-algebra** exists and is given by the mediating arrow from the colimit of the $\omega$-chain $F(0) \rightarrow F^2(0) \rightarrow \cdots$

$$
\text{colim } FD \cong F(\text{colim } D) \rightarrow \text{colim } D
$$


- Why cocompleteness? At least, if $\mathcal{C}$ is cocomplete (existence of any colimit)

$$
\Rightarrow \quad 0 \rightarrow F(0) \rightarrow F^2(0) \rightarrow \cdots \text{ has a colimit}
$$
Why Cocompleteness?

Cor.
Let $\mathcal{C}$ be a cocomplete category. If $F$ is $\omega$-cocontinuos functor, the initial $F$-algebra exists.

Note: If $\mathcal{C}$ is a cpo, this is Knaster-Tarski fixpoint theorem.
Construction of Colimits

**Thm. [Mac Lane V.2.2]**
Let $\mathcal{C}$ be a category with coproducts and coequalizers, and $D : \mathbb{I} \to \mathcal{C}$ a diagram in $\mathcal{C}$.
Then, $\text{colim } D$ exists and is given by the coequalizer

\[
\bigsqcup_{(f : j \to k) \in \text{arr}\mathbb{I}} D(j) \xrightarrow{[\phi_f]} \bigsqcup_{i \in \mathbb{I}} D(i) \rightarrow \text{colim } D
\]

where

\[
\phi_f : D(j) \xrightarrow{D(f)} D(k) \xrightarrow{i} \bigsqcup_{i \in \mathbb{I}} D(i)
\]

\[
\iota_f : D(j) \xrightarrow{i} \bigsqcup_{i \in \mathbb{I}} D(i)
\]
Example: List Types in $\text{Cppo}_\bot$

**Remark**
Non-strict functions can be represented by strict functions in $\text{Cppo}_\bot$

$$[X \rightarrow Y] \cong (X)_\bot \rightarrow_\bot Y$$

Type constructors are $\times \otimes \oplus (\neg)_\bot$

In $\text{Set}$, the list functor is

$$FX = 1 + \mathbb{N} \times X$$
In $\mathbf{Cppo}_\bot$, assume $1, \mathbb{N}$ to be flat cpos.

<table>
<thead>
<tr>
<th>$FX$</th>
<th>$\text{cons}$ : $\mathbb{N} \times L \rightarrow \bot L$</th>
<th>$\text{how strict}$</th>
<th>$\text{i. alg.}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 \oplus \mathbb{N} \times X$</td>
<td>$\mathbb{N} \times L \rightarrow \bot L$</td>
<td>$\text{cons}(\bot, \bot) = \bot$ but $\text{cons}(a, \bot) = a: \bot$</td>
<td>$\text{fin.}&amp;\text{ inf.}$</td>
</tr>
<tr>
<td>$1 \oplus \mathbb{N} \otimes X$</td>
<td>$\mathbb{N} \otimes L \rightarrow \bot L$</td>
<td>$\text{cons}(\bot, l) = \text{cons}(a, \bot) = \bot$</td>
<td>$\mathbb{N}^*$</td>
</tr>
<tr>
<td>$1 \oplus \mathbb{N} \otimes X_\bot$</td>
<td>$\mathbb{N} \otimes L_\bot \rightarrow \bot L$</td>
<td>$\text{cons}(\bot, l) = \bot$ but $\text{cons}(a, \bot) = a: \bot$</td>
<td>$\text{fin.}&amp;\text{ inf.}$</td>
</tr>
<tr>
<td>$1 \oplus \mathbb{N}<em>\bot \otimes X</em>\bot$</td>
<td>$\mathbb{N}<em>\bot \otimes L</em>\bot \rightarrow \bot L$</td>
<td>$\text{cons}(\bot, \bot) = \bot: \bot \neq \bot$</td>
<td>$\text{fin.}&amp;\text{ inf.}$</td>
</tr>
</tbody>
</table>

**Remark**

1. $(\bot, \bot)$ is the least element of $\mathbb{N} \times L$
2. $\text{cons}$ is a bistrict function
3. $\text{cons}$ is a left strict function
4. $\mathbb{N}_\bot \otimes X_\bot \cong (\mathbb{N} \times X)_\bot$, $\text{cons}$ is continuous. "continuous algebra". Haskell’s datatype
Others

- Initial algebra construction via repeated application of $F$
- Note: in $\mathbf{Cppo}_\bot$, infinite coproduct should contain lub of $\omega$-chain while finite coproduct is just a crushed disjoint sum
Why Initial Algebra $=\ Final\ Coalgebra$?

**Def.** Injection-Projection pair $(e, p)$

$$
\begin{array}{c}
C \xrightarrow{e} D \\
\downarrow p
\end{array}
$$

s.t. $p \circ e = \text{id}$, $e \circ p \sqsubseteq \text{id}$.

If such a pair $(e, p)$ exists, $p$ is **uniquely determined** by $e$.

Why coincidence:

$$
\begin{array}{c}
0 \xrightarrow{!} F(0) \xrightarrow{F!} F^2(0) \xrightarrow{F^2!} \ldots
\end{array}
$$

$$
\begin{array}{c}
\mu_0 \\
\mu_1 \\
\mu_2 \\
\mu_3 \\
\exists! m
\end{array}
$$

$$
\begin{array}{c}
\lim \ D \cong \ \text{colim} \ D \cong \ F(\text{colim} \ D) \cong F(\lim \ D)
\end{array}
$$

Actually, $\mu_i = F^i!$ and injective, so has projection
Datatypes and Cocomplete Categories

- **Set** for polynomial types

- **Cppo⊥** for initial/finial (co)algebra coincidence & mixed variant types ("fold" given by Johan Glimming [CALCO’07] for $\mathcal{C}$ that is algebraically compact, has products, coproducts, smc, generator)

- **Cpo** for polynomial with monad types $T$ ("fold" given by Filinski & Stovring [ICFP’07])

- $[\mathcal{C}, \mathcal{C}]$ for nested datatypes, where $\mathcal{C}$ is $\omega$-cocomplete ("generalised fold" given by Johann & Ghani [TLCA’07])

- **Set$^S$** for simple GADT (e.g. phantom type of Expr)

**Conclusion:** **Cppo⊥** for core Haskell
Further Complication: Incorporating Function Types

To treat mixed-variant functor, e.g.,

\[ F X = (X \to \bot X) \bot \]

local continuity is easier to chk than cocontinuity of datatype functor.

This means that we need to use order-enriched category \( \text{Cppo}_\bot \).

**Thm.** Every locally continuous functor

\[ F : \text{Cppo}^{op}_\bot \times \text{Cppo}_\bot \to \text{Cppo}^{op}_\bot \times \text{Cppo}_\bot \]

has an initial algebra, which is also a final coalgebra.
   ▶ Basic lemma (dual is for final coalgebra)
   ▶ Embedding-projection pair construction for mixed variance in enriched setting ⇒ limit=colimit

   ▶ More detailed account for initial algebras for datatypes, such as lists, trees, stack, queues

   ▶ Survey and modern account

(5) Simpson, Plotkin, Complete Axioms for Categorical Fixed-point Operators, LICS’00.