Dependent Inductive and Coinductive Types via Dialgebras in Fibrations

Henning Basold
Radboud University, Nijmegen and CWI, Amsterdam
Joint work with Herman Geuvers

TYPES Meeting
18 May 2014
Outline

1 Introduction

2 Introductory Example: Vectors

3 Dependent Data Types – Fibrationally
1 Introduction

2 Introductory Example: Vectors

3 Dependent Data Types – Fibrationally
Introduction

Goal
Categorical Semantics for (Co)Inductive families in Agda

Hagino’s idea, '87
- Very simple type system (3 type constructors)
- Based on dialgebras
- Can represent product and coproduct (as adjoint functors)
- Flaw: function space requires special treatment

Move to dialgebras on dependent types
- Foundation for dependent data types
- $\Sigma$- and $\Pi$-types representable (as adjoints)
- Correspond to Agda’s inductive data types, and extend its coinductive data types
1 Introduction

2 Introductory Example: Vectors

3 Dependent Data Types – Fibrationally
Introductory Example: Vectors

Example (Vectors over set $A$)

- Lists of length $n$ over $A$: $A^n = A \times \cdots \times A$
- Vector data type: $\text{Vec } A = \{ A^n \}_{n \in \mathbb{N}}$

Definition (Category of set families indexed by $I$)

$$\textbf{Set}^I = \left\{ \begin{array}{l} \text{objects} \\ \text{morphisms} \end{array} \right\}$$

- Objects: $X = \{ X_i \}_{i \in I}$
- Morphisms: $f = \{ f_i : X_i \to Y_i \}_{i \in I}$

Example

$\text{Vec } A$ is an object in $\textbf{Set}^\mathbb{N}$
Example (Constructors of Vec $A$)

Vectors come with

- $\text{nil}_*: 1 \to A^0$ – empty vector
- $\text{cons}_n: A \times A^n \to A^{n+1}$ – prefixing

where $1 = \{\ast\}$ (final object).

These are morphisms in $\textbf{Set}^1$ and $\textbf{Set}^\mathbb{N}$

- $\text{nil} = \{\text{nil}_*\}: \{1\}_{\ast \in 1} \to \{A^0\}_{\ast \in 1}$
- $\text{cons} = \{\text{cons}_n\}_{n \in \mathbb{N}}: \{A \times A^n\}_{n \in \mathbb{N}} \to \{A^{n+1}\}_{n \in \mathbb{N}}$
Example (Constructors of Vec $A$)

Vectors come with

- $\text{nil}_*: \mathbf{1} \rightarrow A^0$ – empty vector
- $\text{cons}_n: A \times A^n \rightarrow A^{n+1}$ – prefixing

where $\mathbf{1} = \{\ast\}$ (final object).

These are morphisms in $\textbf{Set}^1$ and $\textbf{Set}^\mathbb{N}$

- $\text{nil} = \{\text{nil}_*\}: \{\mathbf{1}\}_{\ast \in \mathbf{1}} \rightarrow \{A^0\}_{\ast \in \mathbf{1}}$
- $\text{cons} = \{\text{cons}_n\}_{n \in \mathbb{N}}: \{A \times A^n\}_{n \in \mathbb{N}} \rightarrow \{A^{n+1}\}_{n \in \mathbb{N}}$

Example (Vectors as dialgebra)

Let $F, G: \textbf{Set}^\mathbb{N} \rightarrow \textbf{Set}^1 \times \textbf{Set}^\mathbb{N}$ be functors to the product category

\[
F(X) = (\{\mathbf{1}\}, \{A \times X_n\}_{n \in \mathbb{N}}) \text{ and } 
G(X) = (\{X_0\}, \{X_{n+1}\}_{n \in \mathbb{N}})
\]

Then $(\text{nil}, \text{cons}) : F(\text{Vec} A) \rightarrow G(\text{Vec} A)$ is the initial $(F, G)$-dialgebra.
Example (Constructors of Vec A)

Vectors come with

- $\text{nil}_* : 1 \to A^0$ – empty vector
- $\text{cons}_n : A \times A^n \to A^{n+1}$ – prefixing

where $1 = \{ \ast \}$ (final object).

These are morphisms in $\textbf{Set}^1$ and $\textbf{Set}^\mathbb{N}$

- $\text{nil} = \{ \text{nil}_* \} : \{1\}_* \times 1 \to \{ A^0 \}_* \times 1$
- $\text{cons} = \{ \text{cons}_n \}_n : \{ A \times A^n \}_n \times \mathbb{N} \to \{ A^{n+1} \}_n \times \mathbb{N}$

Example (Vectors as dialgebra)

Let $F, G : \textbf{Set}^\mathbb{N} \to \textbf{Set}^1 \times \textbf{Set}^\mathbb{N}$ be functors to the product category

$$F(X) = (\{1\}, \{ A \times X_n \}_n \times \mathbb{N}) \text{ and } G(X) = (\{X_0\}, \{X_{n+1} \}_n \times \mathbb{N})$$

Then $(\text{nil}, \text{cons}) : F(\text{Vec} A) \to G(\text{Vec} A)$ is the initial $(F, G)$-dialgebra.
Example (Constructors of Vec $A$)

Vectors come with

- $\text{nil}_*: 1 \to A^0$ – empty vector
- $\text{cons}_n: A \times A^n \to A^{n+1}$ – prefixing

where $1 = \{ * \}$ (final object).

These are morphisms in $\text{Set}^1$ and $\text{Set}^\mathbb{N}$

- $\text{nil} = \{ \text{nil}_* \} : \{ 1 \}_{* \in 1} \to \{ A^0 \}_{* \in 1}$
- $\text{cons} = \{ \text{cons}_n \}_{n \in \mathbb{N}} : \{ A \times A^n \}_{n \in \mathbb{N}} \to \{ A^{n+1} \}_{n \in \mathbb{N}}$

Example (Vectors as dialgebra)

Let $F, G : \text{Set}^\mathbb{N} \to \text{Set}^1 \times \text{Set}^\mathbb{N}$ be functors to the product category

$$F(X) = (\{ 1 \}, \{ A \times X_n \}_{n \in \mathbb{N}}) \text{ and }$$
$$G(X) = (\{ X_0 \}, \{ X_{n+1} \}_{n \in \mathbb{N}})$$

Then $(\text{nil}, \text{cons}) : F(\text{Vec } A) \to G(\text{Vec } A)$ is the initial $(F, G)$-dialgebra.
1 Introduction

2 Introductory Example: Vectors

3 Dependent Data Types – Fibrationally
Definition (Fibres of functor \( P : E \to B \))

Fibre above \( I \in B \):

\[
P_I = \begin{cases} 
\text{objects} & A \in E \text{ with } P(A) = I \\
\text{morphisms} & f : A \to B \text{ with } P(f) = \text{id}_I 
\end{cases}
\]
Definition (Fibres of functor $P : E \to B$)

Fibre above $I \in B$:

$$
P_I = \begin{cases} 
\text{objects} & A \in E \text{ with } P(A) = I \\
\text{morphisms} & f : A \to B \text{ with } P(f) = \text{id}_I
\end{cases}
$$

Definition (Cloven Fibration)

A functor $P : E \to B$ with

- a functor $u^* : P_J \to P_I$ for each $u : I \to J$ in $B$,
- $\text{id}_I^* \cong \text{Id}_{P_I}$ and 
- $u^* \circ v^* \cong (v \circ u)^*$

plus coherence conditions.

We call $u^*$ the reindexing or substitution along $u$. 

Henning Dependent Inductive and Coinductive Types via Dialgebras in Fibrat 18.05.15 9
Example

\[
\text{Fam}(\text{Set}) = \begin{cases} 
\text{objects} & (I, X = \{X_i\}_{i \in I}) \\
\text{morphisms} & (u, f) : (I, X) \to (J, Y) \text{ with } \\
& u : I \to J \text{ and } f = \{f_i : X_i \to Y_{u(i)}\}_{i \in I}
\end{cases}
\]

\[
P : \text{Fam}(\text{Set}) \to \text{Set}
\]

\[
P(I, X) = I \quad P(u, f) = u
\]

Using that \( P_I \cong \text{Set}^I \), reindexing along \( u : I \to J \) is given by

\[
u^* : \text{Set}^J \to \text{Set}^I
\]

\[
u^*(X) = \{X_{u(i)}\}_{i \in I}
\]

\[
u^*(f) = \{f_{u(i)} : X_{u(i)} \to Y_{u(i)}\}_{i \in I}
\]
Example (Vectors abstractly)

Recall

\[ G(X) = (\{X_0\}, \{X_{n+1}\}_{n \in \mathbb{N}}) \]

Define

\[ z : 1 \rightarrow \mathbb{N} \quad s : \mathbb{N} \rightarrow \mathbb{N} \]

\[ z(*) = 0 \quad s(n) = n + 1 \]

So that

\[ z^* : \text{Set}^\mathbb{N} \rightarrow \text{Set}^1 \quad s^* : \text{Set}^\mathbb{N} \rightarrow \text{Set}^\mathbb{N} \]

\[ z^*(X) = \{X_0\} \quad s^*(X) = \{X_{n+1}\}_{n \in \mathbb{N}}. \]

and

\[ G = \langle z^*, s^* \rangle : \text{Set}^\mathbb{N} \rightarrow \text{Set}^1 \times \text{Set}^\mathbb{N} \]
A data type signature is a pair \((F, G)\) with

- \(F : C \times P_I \to P_{J_1} \times \cdots \times P_{J_n}\) functor with parameters \(C = P_{L_1} \times \cdots \times P_{L_m}\)

- \(G : P_I \to P_{J_1} \times \cdots \times P_{J_n}\) with \(G = \langle u_1^*, \cdots , u_n^* \rangle\) for \(u_k : J_k \to I\) in \(B\)

- Dependent inductive data type: initial \((\hat{F}, \hat{G})\)-dialgebra, with a functor \(C \to P_I\) as carrier

- Dependent coinductive data type: final \((\hat{G}, \hat{F})\)-dialgebra, with a functor \(C \to P_I\) as carrier
Definition (Data Types in Cloven Fibration \( P : E \to B \))

A data type signature is a pair \((F, G)\) with

- \(F : C \times P_I \to P_{J_1} \times \cdots \times P_{J_n}\) functor with parameters \(C = P_{L_1} \times \cdots \times P_{L_m}\)
- \(G : P_I \to P_{J_1} \times \cdots \times P_{J_n}\) with \(G = \langle u_1^*, \cdots, u_n^* \rangle\) for \(u_k : J_k \to I\) in \(B\)

- **Dependent inductive data type**: initial \((\hat{F}, \hat{G})\)-dialgebra, with a functor \(C \to P_I\) as carrier
- **Dependent coinductive data type**: final \((\hat{G}, \hat{F})\)-dialgebra, with a functor \(C \to P_I\) as carrier
Example (Dependent product $\prod_A : \mathbf{P}_{I \times A} \to I$)

- $(\prod_A, \xi : \pi_A^* \circ \prod_A \Rightarrow \text{Id}_{\mathbf{P}_{I \times A}})$ is final $(\hat{\pi}_A^*, \hat{p}_1)$-dialgebra, where

  \[
  \begin{align*}
  p_1 : \mathbf{P}_{I \times A} \times \mathbf{P}_I &\to \mathbf{P}_{I \times A} \\
  \pi_A^* : \mathbf{P}_I &\to \mathbf{P}_{I \times A}
  \end{align*}
  \]

- Note that $\pi_A^* (\prod_A \mathbf{X})_{(i, a)} = (\prod_A \mathbf{X})_i$

- Denote $g \in (\prod_A \mathbf{X})_i$ by $g : (a : A) \to \mathbf{X}_{(i, a)}$

- Application: $g(a) := \xi_{(i, a)}(g)$
Example (Dependent product $\prod_A : \mathbb{P}_{I \times A} \rightarrow \mathbb{P}_I$)

- $(\prod_A, \xi : \pi_A^* \circ \prod_A \Rightarrow \text{Id}_{\mathbb{P}_{I \times A}})$ is final $(\hat{\pi}_A^*, \hat{\rho}_1)$-dialgebra, where

  $$p_1 : \mathbb{P}_{I \times A} \times \mathbb{P}_I \rightarrow \mathbb{P}_{I \times A}$$
  $$\pi_A^* : \mathbb{P}_I \rightarrow \mathbb{P}_{I \times A}$$

- Note that $\pi_A^* (\prod_A X)_{(i,a)} = (\prod_A X)_i$

- Denote $g \in (\prod_A X)_i$ by $g : (a : A) \rightarrow X_{(i,a)}$

- Application: $g(a) := \xi_{(i,a)}(g)$
Theorem

There are isomorphisms

\[ \text{DiAlg}(F, G) \cong \text{Alg} \left( \bigoplus_{u_1} \circ F_1 + I \cdots + I \bigoplus_{u_n} \circ F_n \right) \]

and

\[ \text{DiAlg}(G, F) \cong \text{CoAlg} \left( \bigotimes_{u_1} \circ F_1 \times I \cdots \times I \bigotimes_{u_n} \circ F_n \right). \]

where \( F_k = \pi_k \circ F \) is the kth component of \( F \) and \( G = \langle u_1^*, \ldots, u_n^* \rangle \). In particular, the existence of inductive data types and initial algebras, and coinductive data types and final coalgebras, respectively, coincide.
Interpretation of Syntactic Data Types

Definition

A fibration \( P : E \rightarrow B \) is \textit{data type complete} all initial \((F, G)\)-dialgebras and final \((G, F)\)-dialgebras for functors \( F \) of the following form exist.

\[
F ::= K^I_A \mid \pi_k \mid f^* \mid F_2 \circ F_1 \mid \langle F_1, F_2 \rangle \mid \mu(\widehat{F}, \widehat{G}) \mid \nu(\widehat{G}, \widehat{F}),
\]
Interpretation of Syntactic Data Types

Definition

A fibration $P : E \to B$ is \textit{data type complete} all initial $(F, G)$-dialgebras and final $(G, F)$-dialgebras for functors $F$ of the following form exist.

$$F ::= K^I_A \mid \pi_k \mid f^* \mid F_2 \circ F_1 \mid \langle F_1, F_2 \rangle \mid \mu(\hat{F}, \hat{G}) \mid \nu(\hat{G}, \hat{F}),$$

- Combined with comprehension, we can interpret syntactic data types.
Interpretation of Syntactic Data Types

Definition

A fibration $P : \mathbf{E} \to \mathbf{B}$ is data type complete all initial $(F, G)$-dialgebras and final $(G, F)$-dialgebras for functors $F$ of the following form exist.

$$F ::= K_A^I \mid \pi_k \mid f^* \mid F_2 \circ F_1 \mid \langle F_1, F_2 \rangle \mid \mu(\widehat{F}, \widehat{G}) \mid \nu(\widehat{G}, \widehat{F}),$$

- Combined with comprehension, we can interpret syntactic data types.
- I’m working on computation of data types via (co)limits of chains.
Conclusion etc.

Missing from this Talk

- Induction and Coinduction, see draft on my homepage: www.cs.ru.nl/~hbasold/publications/
- Beck-Chevalley condition for sound interpretation, see same draft
- Encoding of W- and M-types, see Agda standard library and https://github.com/hbasold/Sandbox/tree/master/TypeTheory/Container

Future Work

- Already mentioned: finish proof of data type completeness via chain constructions.
- Are there groupoids that are data type complete?
Usually I thank here, but now we can just go to the next set of slides!