

# Comodules over relative comonads for streams and infinite matrices

Benedikt Ahrens

joint work with Régis Spadotti

Institut de Recherche en Informatique de Toulouse  
Université Paul Sabatier

CMCS 2014

- Category-theoretic semantics of **co**inductive data types in intensional Martin-Löf type theory (IMLTT)
- ↪ Develop a notion of “coalgebra” for the signature of a codata type
- Incorporate canonical cosubstitution

- ① Syntax: inductives and substitution
- ② Homogeneous cosyntax: streams
- ③ Heterogeneous cosyntax: infinite triangular matrices

- 1 Syntax: inductives and substitution
- 2 Homogeneous cosyntax: streams
- 3 Heterogeneous cosyntax: infinite triangular matrices

Motivation: **binding syntax** in MLTT

$$\text{Lc} : \text{Type} \rightarrow \text{Type}$$
$$\frac{x : X}{\text{var}(x) : \text{Lc}(X)} \quad \frac{s, t : \text{Lc}X}{\text{app}(s, t) : \text{Lc}X} \quad \frac{t : \text{Lc}(X + 1)}{\text{abs}(t) : \text{Lc}X}$$

Heterogeneity of **abs**:

recursive argument with bigger parameter  $X + 1$

Substitution:

$$\text{subst}_{X,Y} : (X \rightarrow \text{Lc}Y) \rightarrow \text{Lc}X \rightarrow \text{Lc}Y$$

Avoiding capture:

$$\text{shift}_{X,Y} : (X \rightarrow \text{Lc}Y) \rightarrow X + 1 \rightarrow \text{Lc}(Y + 1)$$

Initial semantics for lambda calculus: Fiore, Plotkin & Turi '99

- characterizes not only data type but also **substitution**
- reformulated using **monads** by Hirschowitz & Maggesi '07

Basis for this reformulation:

Lemma (Substitution is monadic: Altenkirch & Reus '99)

$(Lc, var, subst)$  *forms a monad (in Kleisli form)*

Definition (Algebra for signature of Lc, H & M '07)

- a monad  $(T, \text{unit}, \text{bind})$  on Type
- two **morphisms of modules over  $T$** ,

$$\text{App} : T \times T \rightarrow T$$

$$\text{Abs} : T(\_ + 1) \rightarrow T$$

Definition (Algebra for signature of Lc, H & M '07)

- a monad  $(T, \text{unit}, \text{bind})$  on Type
- two **morphisms of modules over  $T$** ,

$$\text{App} : T \times T \rightarrow T$$

$$\text{Abs} : T(\_ + 1) \rightarrow T$$

“Module morphism” expresses **commutativity with bind**:

$$\text{bind } f \circ \text{App} = \text{App} \circ (\text{bind } f)^2$$

$$\text{bind } f \circ \text{Abs} = \text{Abs} \circ \text{bind } (\text{shift } f)$$



# Initial semantics for $\lambda$ -calculus using monads

Definition (Algebra for signature of Lc, H & M '07)

- a monad  $(T, \text{unit}, \text{bind})$  on Type
- two **morphisms of modules over  $T$** ,

$$\text{App} : T \times T \rightarrow T$$

$$\text{Abs} : T(\_ + 1) \rightarrow T$$

“Module morphism” expresses **commutativity with bind**:

$$\text{bind } f \circ \text{App} = \text{App} \circ (\text{bind } f)^2$$

$$\text{bind } f \circ \text{Abs} = \text{Abs} \circ \text{bind } (\text{shift } f)$$

Lemma (Initial semantics for Lc, H & M '07)

$(\text{Lc}, \text{app}, \text{abs})$  is the initial algebra, where  $\text{Lc} = (\text{Lc}, \text{var}, \text{subst})$

# Goal: characterize **codata** types with **cosubstitution**

## Goal

dualize techniques of H & M to characterize

- **codata** types in **intensional ML type theory** with
- **cosubstitution**

as **terminal** object

## In this talk

- streams
- infinite triangular matrices

- ① Syntax: inductives and substitution
- ② Homogeneous cosyntax: streams
- ③ Heterogeneous cosyntax: infinite triangular matrices

# Streams over a base type

- Streams = infinite lists over some base type  $A$

$$\begin{array}{c|ccc} a_0 & a_1 & a_2 & \dots \\ \hline \text{head} & \text{tail} & & \end{array}$$

- Specified by destructors

$$\frac{s : \text{Stream } A}{\text{head}_A(s) : A}$$

$$\frac{s : \text{Stream } A}{\text{tail}_A(s) : \text{Stream } A}$$

- “Sameness” = bisimilarity in IMLTT

$$\frac{s \sim s'}{\text{head}(s) = \text{head}(s')}$$

$$\frac{s \sim s'}{\text{tail}(s) \sim \text{tail}(s')}$$

- $(\text{Stream}A, \sim)$  is a *setoid*,

$$\text{Stream} : \text{Type} \rightarrow \text{Setoid}$$

- Canonical cosubstitution

$$\text{cosubst}_{A,B} : (\text{Stream}A \rightarrow B) \rightarrow \text{Stream}A \rightarrow \text{Stream}B$$

is compatible with bisimilarity:

$$\text{cosubst}_{A,B} : \text{Setoid}(\text{Stream}A, \text{eq}B) \rightarrow \text{Setoid}(\text{Stream}A, \text{Stream}B)$$

with

$$\text{eq} : \text{Type} \rightarrow \text{Setoid} \quad \text{eq} \dashv \text{forget}$$

## Lemma

$(\text{Stream}, \text{head}, \text{cosubst})$  is a *comonad relative to*  
 $\text{eq} : \text{Type} \rightarrow \text{Setoid}$ .

Definition (**Relative** (co)monad, Alten., Chapm. & Uust. '10)

- underlying functor is **not** necessarily **endo**
- needs “mediating” functor (above:  $\text{eq}$ )

## Morphisms of modules over monads

characterize commutativity of substitution with constructors

$$\text{app} : \text{Lc} \times \text{Lc} \rightarrow \text{Lc}$$

$$\text{subst } f \circ \text{app} = \text{app} \circ (\text{subst } f)^2$$

## Morphisms of **comodules** over **relative comonads**

characterize commutativity of **cosubstitution** with destructors

$$\text{tail} : \text{Stream} \rightarrow \text{Stream}$$

$$\text{tail} \circ \text{cosubst } f = \text{cosubst } f \circ \text{tail}$$

## Definition (Category of coalgebras)

A coalgebra for the signature of `Stream` is given by a pair  $(S, t)$ :

- a comonad  $S$  relative to  $\text{eq} : \text{Type} \rightarrow \text{Setoid}$
- a morphism of comodules over  $S$

$$t : S \rightarrow S$$

Morphisms: ...

## Lemma

$(\text{Stream}, \text{tail})$  is the terminal object in the above category.



- ① Syntax: inductives and substitution
- ② Homogeneous cosyntax: streams
- ③ Heterogeneous cosyntax: infinite triangular matrices

# An example of cosyntax: infinite triangular matrices

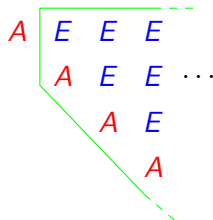
Tri: the codata type of infinite triangular matrices

- omit redundant information below the diagonal
- have a **variable** type  $A$  of diagonal elements
  - e.g. invertible elements
- a fixed type  $E$  of elements for rest of matrix
- usage: Pascal matrices (binomial coefficients), mathematical physics (infinite-dim. problems)

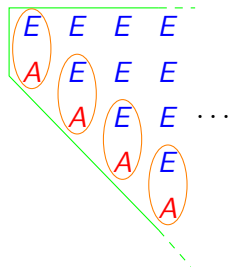
$$\begin{array}{cccc} A & E & E & E \\ & A & E & E \dots \\ & & A & E \\ & & & A \end{array}$$

# Matrices through trapezia: the destructors of Tri

$$\frac{t : \text{Tri}A}{\text{top}_A(t) : A}$$

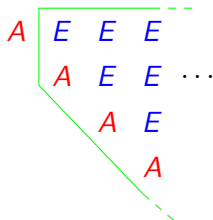


$$\frac{t : \text{Tri}A}{\text{rest}_A(t) : \text{Tri}(E \times A)}$$

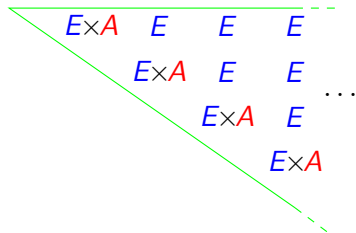


# Matrices through trapezia: the destructors of Tri

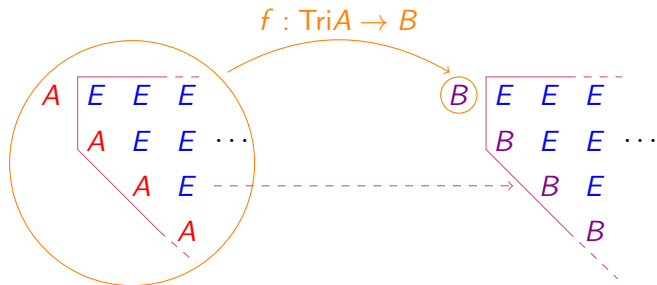
$$\frac{t : \text{Tri}A}{\text{top}_A(t) : A}$$



$$\frac{t : \text{Tri}A}{\text{rest}_A(t) : \text{Tri}(E \times A)}$$



$$\text{redec}_{A,B} : (\text{Tri}A \rightarrow B) \rightarrow (\text{Tri}A \rightarrow \text{Tri}B)$$



$$\text{top} \circ \text{redec } f := f \quad \text{and}$$

$$\text{rest} \circ \text{redec } f := \text{redec } (\text{lift } f) \circ \text{rest}$$

with  $\text{lift } f : \text{Tri}(E \times A) \rightarrow E \times B$

# Tri is a weak constructive comonad

Sameness = **bisimilarity**

Bisimilarity  $\sim$  coinductively defined via destructors

$$\frac{t \sim t'}{\text{top}(t) = \text{top}(t')} \qquad \frac{t \sim t'}{\text{rest}(t) \sim \text{rest}(t')}$$

Lemma (Matthes and Picard '11)

$(\text{Tri} : \text{Type} \rightarrow \text{Type}, \text{top}, \text{redec})$  forms a “weak constructive comonad”.

$\rightsquigarrow$  “weak constructive” refers to compatibility conditions with bisimilarity

# Tri is a relative comonad

Alternatively,  $\text{Tri}A$  is a **setoid** rather than a (plain) type

$$\text{top}_A : \text{Setoid}(\text{Tri}A, \text{eq}A)$$
$$\text{redec}_{A,B} : \text{Setoid}(\text{Tri}A, \text{eq}B) \rightarrow \text{Setoid}(\text{Tri}A, \text{Tri}B)$$

with  $\text{eq} : \text{Type} \rightarrow \text{Setoid}$

# Tri is a relative comonad

Alternatively,  $\text{Tri}A$  is a **setoid** rather than a (plain) type

$$\text{top}_A : \text{Setoid}(\text{Tri}A, \text{eq}A)$$
$$\text{redec}_{A,B} : \text{Setoid}(\text{Tri}A, \text{eq}B) \rightarrow \text{Setoid}(\text{Tri}A, \text{Tri}B)$$

with  $\text{eq} : \text{Type} \rightarrow \text{Setoid}$

Lemma (Reformulation of Matthes and Picard '11)

$(\text{Tri} : \text{Type} \rightarrow \text{Setoid}, \text{top}, \text{redec})$  forms a *comonad relative to*  $\text{eq} : \text{Type} \rightarrow \text{Setoid}$ .



## Morphisms of modules over monads

characterize commutativity of substitution with constructors

$$\begin{aligned} \text{abs} &: \text{Lc}(\_ + 1) \rightarrow \text{Lc} \\ \text{subst } f \circ \text{abs} &= \text{abs} \circ \text{subst } (\text{shift } f) \end{aligned}$$

## Morphisms of **comodules** over **relative comonads**

characterize commutativity of **cosubstitution** with destructors

$$\begin{aligned} \text{rest} &: \text{Tri} \rightarrow \text{Tri}(E \times \_) \\ \text{rest} \circ \text{redec } f &= \text{redec } (\text{lift } f) \circ \text{rest} \end{aligned}$$

## Definition (Category of coalgebras)

A coalgebra for the signature of Tri is given by a pair  $(T, r)$ :

- a comonad  $T$  relative to  $\text{eq} : \text{Type} \rightarrow \text{Setoid}$
- a morphism of comodules over  $T$

$$r : T \rightarrow T(E \times \_)$$

Morphisms: ...

## Lemma

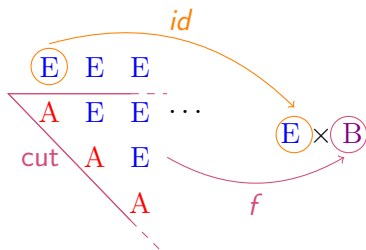
$(\text{Tri}, \text{rest})$  is the terminal object in the above category.

That's **almost** how it works ...

Definition of

$$\text{lift}_A : (\text{Tri}A \rightarrow B) \rightarrow \text{Tri}(E \times A) \rightarrow E \times B$$

requires auxiliary function  $\text{cut}_A : \text{Tri}(E \times A) \rightarrow \text{Tri}A$



# A specified cut for any coalgebra

- we were not able to define cut categorically
- fix: every coalgebra  $(T, r)$  comes with a **specified**

$$c_A : T(E \times A) \rightarrow TA$$

and equations characterizing  $c$

- $c^{\text{Tri}} := \text{cut}$  for  $\text{Tri}$  uniquely determined by these equations

Lemma (for real this time)

$(\text{Tri}, \text{cut}, \text{rest})$  is terminal in the category of coalgebras  $(T, c, r)$ .

## Mechanization in the proof assistant Coq

- helped to find a mistake we made (already made by Abel, Matthes & Uustalu '05)
- reuses code by Matthes & Picard '11
- 3000 lines of code, among which 1500 by MP '11
- fully constructive and axiom-free
- tedious: coinduction in Coq cumbersome

- Coinductive type + bisimilarity as setoid in IMLTT
- **Stream** and **Tri** are **relative** comonads
- Develop comodules over relative comonads
- Terminal coalgebra semantics for **Stream** and **Tri**
- Bisimilarity and redecoration are part of universal object
- **Tri** not as straightforward as the  $\lambda$ -calculus because of **cut**
- Mechanization in **Coq**

- Coinductive type + bisimilarity as setoid in IMLTT
- **Stream** and **Tri** are **relative** comonads
- Develop comodules over relative comonads
- Terminal coalgebra semantics for **Stream** and **Tri**
- Bisimilarity and redecoration are part of universal object
- **Tri** not as straightforward as the  $\lambda$ -calculus because of **cut**
- Mechanization in **Coq**

Thanks for your attention

## Some references

- Altenkirch, Chapman & Uustalu: *Monads need not be endofunctors*
- Hirschowitz & Maggesi: *Modules over Monads and Linearity*
- Matthes & Picard: *Verification of Redecoration for Infinite Triangular Matrices using Coinduction*
- preprint about this work on the arXiv

TikZ pictures used with permission from Matthes and Picard