
The Art of Saying “No” – How to Politely Eat Your Way Through an Infinite Meal

Dirk Pattinson, Imperial College London
(joint work with Neil Ghani and Peter Hancock)

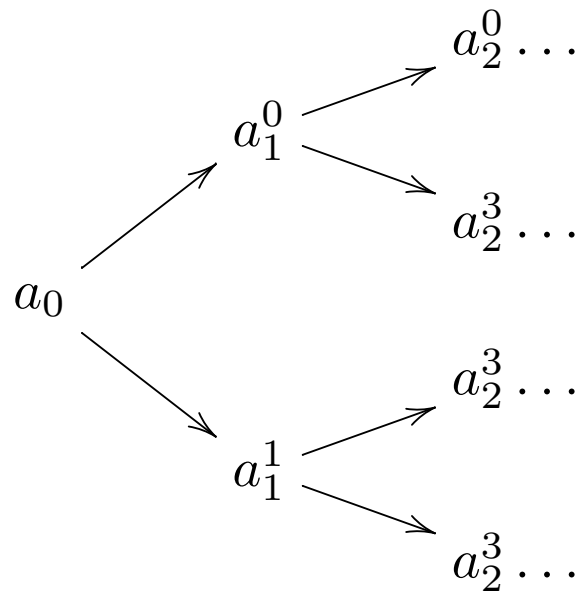
CCC 2009, Cologne

Infinite Objects: Examples

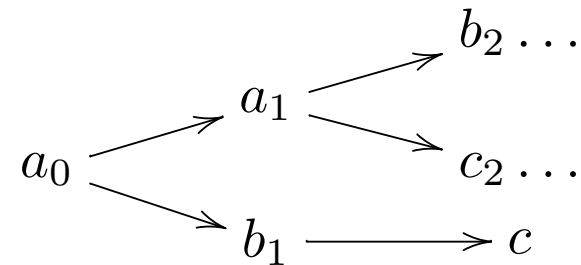
Infinite Streams over A

$$a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \dots$$

Infinite Binary Trees over A



Signatures (variable branching)



Coalgebraic Representation

Infinite Streams. Coalgebras of type $S \rightarrow A \times S$

- every $s \in S$ generates a label $\sigma_0(s) \in A$ and one successor $\sigma_1(s) \in S$

Infinite Binary Trees. Coalgebras of type $S \xrightarrow{\sigma} S \times A \times S$

- every $s \in S$ generates a label $\sigma_0(s)$ and two successors $\sigma_1(s)$ and $\sigma_2(s)$

Signatures. Coalgebras of type $S \xrightarrow{\sigma} A \times S^2 + B \times S + C$

- every $s_0 \in S$ generates
 - either a label $\sigma_0(s) \in A$ and two successors $\sigma_1(s), \sigma_2(s) \in S$, or
 - a label $\sigma_0(s) \in B$ and one successor $\sigma_1(s) \in S$, or
 - a label $\sigma_0(s) \in C$ and no successors.

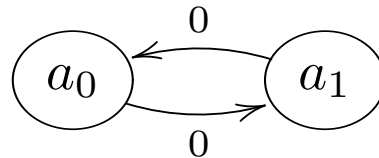
Coalgebras in General. $S \xrightarrow{\sigma} T(S)$ for a functor $T : \text{Set} \rightarrow \text{Set}$

Representation of Infinite Objects

Idea. In a coalgebra $S \xrightarrow{\sigma} TS$

- every $s \in S$ represents “infinite behaviour” – but not uniquely
- not every “type of infinite behaviour” corresponds to some $s \in S$

Example. Take $S = \{a_0, a_1\}$ and $S \rightarrow \{0, 1\} \times S$



- both a_0 and a_1 represent $(0, 0, \dots)$, but none represents e.g. $(0, 1, \dots)$

Final Coalgebras. (Z, ζ) final if every “behaviour” has a unique representative

$$\begin{array}{ccc} S & \xrightarrow{\exists! u} & Z \\ \forall \sigma \downarrow & & \zeta \downarrow \\ TS & \xrightarrow{Tu} & TZ \end{array}$$

Examples

Streams, i.e. T -coalgebras for $TS = A \times S$

$$\begin{array}{ccc} S & \xrightarrow{u} & A^\omega \\ \sigma \downarrow & & \downarrow \langle \text{hd}, \text{tl} \rangle \\ A \times S & \xrightarrow{A \times u} & A \times A^\omega \end{array}$$

where $u(s)_n = \sigma_0 \circ \underbrace{\sigma_1 \circ \dots \circ \sigma_1}_{n \text{ times}}(s)$ is the unique morphism

- *final coalgebra*: infinite streams, i.e. $(A^\omega, \langle \text{hd}, \text{tl} \rangle)$

Infinite Binary Trees, i.e. coalgebras for $TS = A \times S^2$

- *final coalgebra*: $(\text{Tree}(A), \langle \text{lab}, \text{lft}, \text{rt} \rangle)$ where lab/lft/rt are label and subtrees

Enter Topology ...

Intuition. Continuous Functions $f : \nu X.TX \rightarrow B$

- output $b \in B$ after reading *finite amount* of information in $\nu X.TX$

Example. Infinite Streams, or coalgebraically $\nu X.A \times X \rightarrow B$

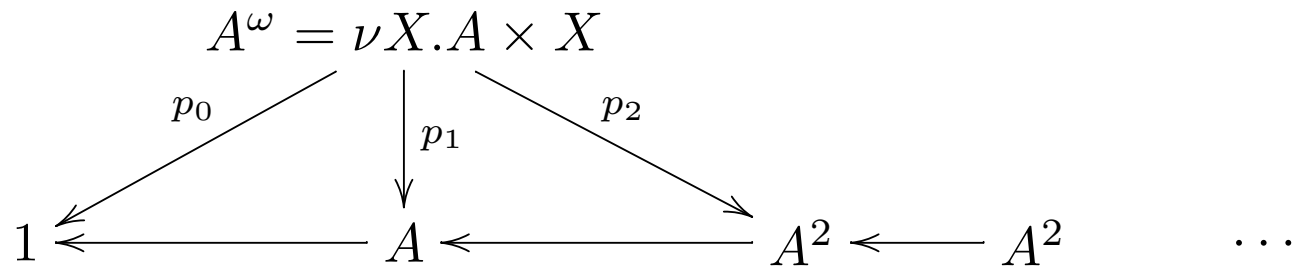
- $f(\alpha)$ depends on finite initial prefix of α

Conceptually. This is the Cantor topology on A^ω (with A discrete)

- generated by $\bar{\alpha} \cdot A^\omega$ where $\bar{\alpha} \in A^*$

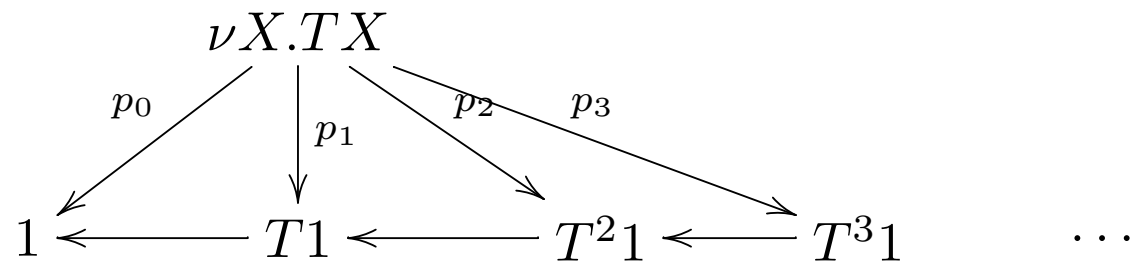
Coalgebraic View

Final Coalgebras arise as *infinite limits*: e.g. streams



Topology generated by $p_i^{-1}(o)$, $o \subseteq A^i$ open

Coalgebraic Generalisation. Suppose $\nu X. TX \xrightarrow{\sigma} T(\nu X. TX)$



where $p_{i+1} = Tp_i \circ \sigma$. Topology generated by $p_i^{-1}(o)$, “ $o \subseteq T^i 1$ open”

Continuous Functions: The Case of Streams

Goal. Characterise continuous functions of type $A^\omega \rightarrow B$ with B discrete.

Continuity. (A and B discrete) $f : A^\omega \rightarrow B$ is continuous ...

- iff f locally constant.

$$(\forall (a_0, a_1, \dots) \in A^\omega) (\exists n \in \omega) f \text{ constant on } (a_0, a_1, \dots, a_n) \cdot A^\omega$$

- iff f is in the least class C closed under

$$\frac{f \text{ constant}}{C(f)} \qquad \frac{(\forall a \in A) C(f(a : _))}{C(f)}$$

Proof (\Leftarrow) locally constant functions are so closed.

Proof (\Rightarrow) classical logic and dependent choice.

Representation of Continuous Stream Functions

Idea. Proofs of Continuity define the *least class* of functions

$$\frac{f \text{ constant}}{C(f)} \qquad \frac{(\forall a \in A)C(f(a : _))}{C(f)}$$

and can be represented as an *inductive* data type:

$$R = \mu X. B + (A \rightarrow X) \cong B + (A \rightarrow R)$$

with two constructors: $\text{Ret} : B \rightarrow R$ and $\text{Rd} : (A \rightarrow R) \rightarrow R$

from which a continuous function can be extracted:

$$\begin{aligned} \text{eat} : \quad & \mu X. B + (A \rightarrow X) \quad \rightarrow A^\omega \quad \rightarrow B \\ \text{eat} \quad & (\text{Ret } b) \quad (a : \alpha) = b \\ \text{eat} \quad & (\text{Rd } f) \quad (a : \alpha) = \text{eat}(f \ a)\alpha \end{aligned}$$

Theorem. If \rightarrow_c is continuous functions, then $\text{eat} : R \rightarrow (A^\omega \rightarrow_c B)$ is onto.

From Streams to Trees

Goal. Classify functions $\text{Tree}(A) \rightarrow_c B$ where $\text{Tree}(A) = \nu X. A \times X^2$

Idea. Let R denote the type of representatives with constructors Rd and Ret .

$$\begin{aligned} \text{eat} : R &\rightarrow \text{Tree}(A) \rightarrow B \\ \text{eat} (\text{Ret } b) (a, l, r) &= b \\ \text{eat} (\text{Rd } f) (a, l, r) &= \text{eat}(f a)(l, r) \end{aligned}$$

Observation. $\text{eat}(f a) : \text{Tree}(A)^2 \rightarrow B$, so $f(a)$ represents $\text{Tree}(A)^2 \rightarrow B$

Mathematical Obfuscation. R_n represents $\text{Tree}(A)^n \rightarrow B$

$$\begin{aligned} \text{eat}_n : R_n &\rightarrow \text{Tree}(A)^n \rightarrow B \\ \text{eat}_n (\text{Ret } b) (t_1, \dots, t_n) &= b \\ \text{eat}_n (\text{Rd}_i f) (t_1, \dots, t_n) &= \text{eat}_{n+1}(f a_i)(t_1, \dots, t_{i-1}, l, r, t_{i+1}, \dots, t_n) \end{aligned}$$

where l, r are the left/right subtree of t_i . **Constructors.** $\text{Ret}, \text{Rd}_1, \dots, \text{Rd}_n$

Escaping the Underworld of Indices

Desired (Inductive) Type with constructors $\text{Ret}, \text{Rd}_1, \dots, \text{Rd}_n$ as above.

$$R_n \cong B + \sum_{i \in n} (A \rightarrow R_{n+1})$$

Realisation Mapping.

$$\text{eat}_n : R_n \rightarrow \text{Tree}(A)^n \rightarrow B$$

$$\text{eat}_n (\text{Ret } b) (t_1, \dots, t_n) = b$$

$$\text{eat}_n (\text{Rd}_i f) (t_1, \dots, t_n) = \text{eat}_{n+1}(f \ a_i)(t_1, \dots, t_{i-1}, l, r, t_{i+1}, \dots, t_n)$$

Taking Indices Seriously.

$$R(n) \cong B + \sum_{i \in n} (A \rightarrow R(n+1))$$

Observation. Now R has type $\text{Set} \rightarrow \text{Set}$ – and we want the *least* such

$$R = \mu F : \text{Set} \rightarrow \text{Set}. \Lambda I : \text{Set}. B + I \times (A \rightarrow R(I+1))$$

Conceptual Digression

Streams. Represent $\text{Stream}(A)^S \rightarrow B$ by $R(S)$ where

$$R(S) = \mu X. B + S \times (A \rightarrow X)$$

- each $R(S)$ is an initial algebra for a functor of type $\text{Set} \rightarrow \text{Set}$
- $\text{eat}(S)$ defined by initiality of $R(S)$ – *separately* for all arities

Linearity: Family of Inductive Types

Trees. Represent $\text{Tree}(A)^S \rightarrow_c B$ by $R(S)$ where

$$R = \mu F : \text{Set} \rightarrow \text{Set}. \Lambda S : \text{Set}. B + S \times (A \rightarrow R(X + 1))$$

- R is an initial algebra for a functor $(\text{Set} \rightarrow \text{Set}) \rightarrow (\text{Set} \rightarrow \text{Set})$
- eat is natural and defined by initiality of R – *simultaneously* for all arities

Nonlinearity: Inductive Family of Types

Infinite Objects of Container Type

Container Functors. (Abbot, Altenkirch, Ghani)

$$(S \triangleleft P)(X) = \sum_{s \in S} X^{P(s)}$$

- S : Set is a set of *shapes*, each of which stores data
- $P : S \rightarrow \text{Set}$ associates a set of *positions* to every shape

Continuous Functions of type $(\nu X.(S \triangleleft P)X)^I \rightarrow B$

$$R = \mu F : \text{Set} \rightarrow \text{Set} . \Lambda I : \text{Set} . B + \sum_{i \in I} \prod_{s \in S} F(I + P(s))$$

Unfolding Isomorphisms.

$$R(I) \cong B + \sum_{i \in I} \prod_{s \in S} F(I + P(s))$$

Intuition.

- if not constant, select tree ($i \in I$), extract root ($s \in S$), behead and continue

Discrete Codomains are Boring

Next Goal. Represent $A^\omega \rightarrow_c B^\omega$

Idea. $f : A^\omega \rightarrow B^\omega$ is continuous iff we have an *infinite* proof

$$(R) \frac{\forall a (C(f(a : -)))}{C(f)} \qquad (W) \frac{C(f)}{C(\lambda \alpha. b : f(\alpha))}$$

where, on any branch in a proof, the right hand rule occurs infinitely often.

Induced Data Type. Wrap up finite occurrences of (R) using a μ

$$R \cong \nu X. \mu Y. B \times X + (A \rightarrow Y) \cong B \times R + (A \rightarrow R)$$

with constructors $\text{Ret} : B \times R \rightarrow R$ and $\text{Rd} : (A \rightarrow R) \rightarrow R$

Extracted Continuous Function.

$$\begin{aligned} \text{eat} : \quad & \nu X. \mu Y. B \times X + (A \rightarrow Y) \quad \rightarrow A^\omega \quad \rightarrow B^\omega \\ \text{eat} \quad & (\text{Ret } (b, r)) \quad (a : \alpha) = b : \text{eat } r \ (a : \alpha) \\ \text{eat} \quad & (\text{Rd } f) \quad (a : \alpha) = \text{eat } (f \ a) \ \alpha \end{aligned}$$

Alternative Computational Representation

We know. Continuous functions of type $A^\omega \rightarrow B$ are represented by

$$A^\omega \rightarrow_C B \rightsquigarrow R = \mu X. B + (A \rightarrow X)$$

Idea. Re-start the computation as soon as a digit has been produced

$$A^\omega \rightarrow_C B^\omega \rightsquigarrow \nu X. \mu Y. B \times X + (A \rightarrow Y)$$

with the same computational interpretation

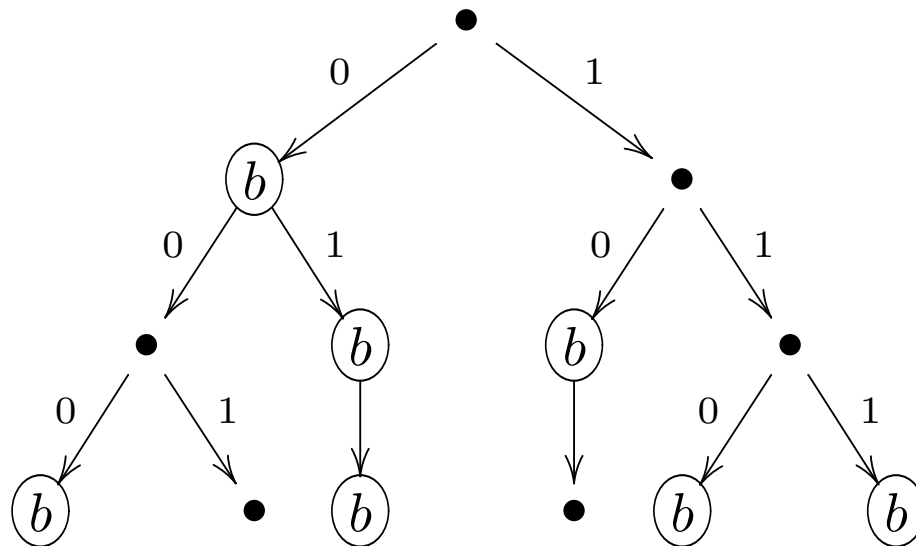
$$\begin{array}{l} \text{eat} : \quad \nu X. \mu Y. B \times X + (A \rightarrow Y) \quad \rightarrow A^\omega \quad \rightarrow B^\omega \\ \text{eat} \quad \quad \quad (\text{Ret } (b, r)) \quad \quad \quad (a : \alpha) \quad = b : \text{eat } r (a : \alpha) \\ \text{eat} \quad \quad \quad (\text{Rd } f) \quad \quad \quad (a : \alpha) \quad = \text{eat } (f a) \alpha \end{array}$$

Note. Occurrence of $B \times X$ suggests that “codomain slots in”

Stream Functions are Trees

Observation. First-Order Functions $A^\omega \rightarrow B^\omega$ are *trees*

$$R = \nu X. \mu Y. B \times X + (A \rightarrow Y)$$



Initiality guarantees infinitely many labels on every path

General Codomain

More Ambitious Goal. Represent $A^\omega \rightarrow \nu X.(S \triangleleft P)X = \nu X. \sum_{s \in S} X^{P(s)}$

By Analogy.

$$R = \nu X. \mu Y. \sum_{s \in S} X^{P(s)} + (A \rightarrow Y) \cong \sum_{s \in S} R^{P(s)} + (A \rightarrow R)$$

with constructors $\text{Ret}_s : (P(s) \rightarrow R) \rightarrow R$ and $\text{Rd} : (A \rightarrow R) \rightarrow R$

Associated Functional.

$$\text{eat} : \nu X. \mu Y. \sum_{s \in S} X^{P(s)} + (A \rightarrow Y) \rightarrow A^\omega \rightarrow \nu Z. (P \triangleleft S)Z$$

$$\text{eat} \quad (\text{Ret}_s (r_i)) \quad (a : \alpha) = (s, (\text{eat } r_i (a : \alpha))_{i \in P(s)})$$

$$\text{eat} \quad (\text{Rd } f) \quad (a : \alpha) = \text{eat } (f a) \alpha$$

Observation.

- *codomain* just “slots in”, more general *domains* by same recipe

Induction Meets Coinduction

Example. Continuous Stream Functions

$$f : A^\omega \rightarrow_c B^\omega$$

are represented by

$$\underbrace{\nu X. \overbrace{\mu Y. B \times X + Y^A}^{T_A(B \times X)}}_{P_A(B)}$$

Lambek's Lemma.

$$P_A(B) = (\nu X)(\mu Y)B \times X + Y^A \cong (\mu Y)B \times P_A(B) + Y^A$$

Pleasant Mathematical Theory.

- supports both *inductive* and *coinductive* definitions and proofs.
- similar for other (co)domains

Inductive Maps Between Coinductive Types

Example. Composition: $P_B(C) \times P_A(B) \rightarrow P_A(C)$ where

$$T_A(B) = \mu X. B + (A \rightarrow X) \quad \text{and} \quad P_A(B) = \nu X. T_A(B \times X)$$

Operation on *representatives*

$$\begin{array}{ccc} P_B(C) \times P_A(B) & \longrightarrow & P_A(C) \\ \downarrow & & \downarrow \\ (B^\omega \rightarrow_c C^\omega) \times (A^\omega \rightarrow_c B^\omega) & \longrightarrow & (A^\omega \rightarrow_c C^\omega) \end{array}$$

As $P_A(B) \cong T_A(B \times P_A(B))$ is bi-inductive: composition

$$\gamma : S = T_B(C \times P_B C) \times T_A(B \times P_A B) \rightarrow T_A(C \times S)$$

is an *inductively defined* map between *coinductive* types

More on Composition

Inductive definition of composition

$$\begin{aligned} \gamma : S = T_B(C \times P_B C) \times T_A(B \times P_A B) &\rightarrow T_A(C \times S) \\ \langle \mathbf{Ret} \langle c, p_{bc} \rangle, t_{ab} \rangle &\mapsto \mathbf{Ret} \langle c, \mathbf{out} p_{bc}, t_{ab} \rangle \\ \langle \mathbf{Rd} \phi, \mathbf{Ret} \langle b, p_{ab} \rangle \rangle &\mapsto \gamma \langle \phi b, \mathbf{out} p_{ab} \rangle \\ \langle t_{bc}, \mathbf{Rd} \psi \rangle &\mapsto \mathbf{Rd} \lambda a. \gamma \langle t_{bc}, \psi a \rangle \end{aligned}$$

whose *coinductive* cousin ($\mathbf{out} : \nu F \rightarrow F(\nu F)$)

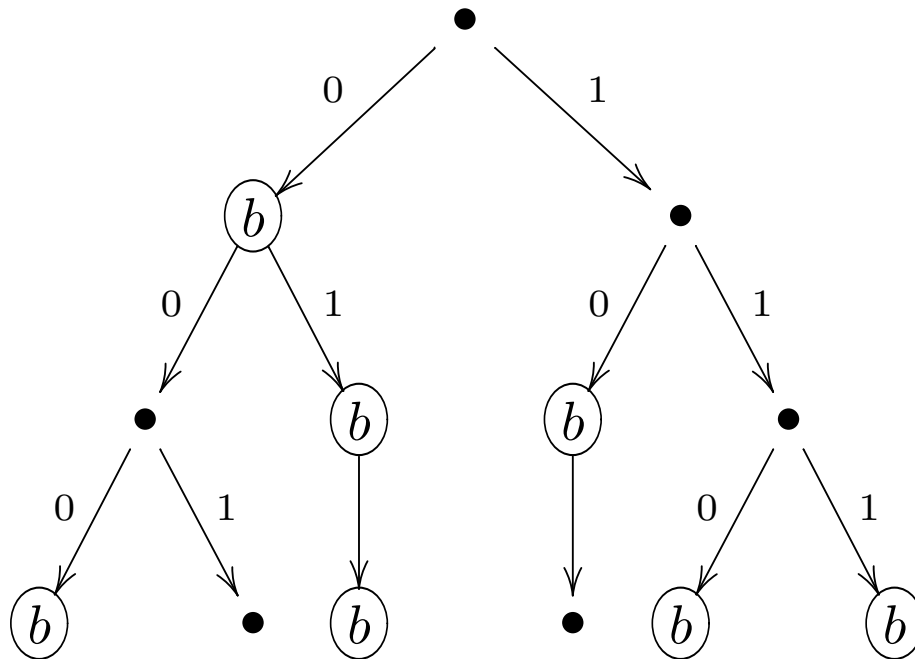
$$\begin{aligned} \chi : P_B(C) \times P_A(B) &\rightarrow P_A(C) \\ \langle post, pre \rangle &\mapsto (\mathbf{unfold} \gamma) \langle \mathbf{out} post, \mathbf{out} pre \rangle \end{aligned}$$

represents composition.

This is *output centered* – alternatives are possible.

Higher-Order Functions

Observation. First-Order Functions $A^\omega \rightarrow B^\omega$ are *trees*



Idea.

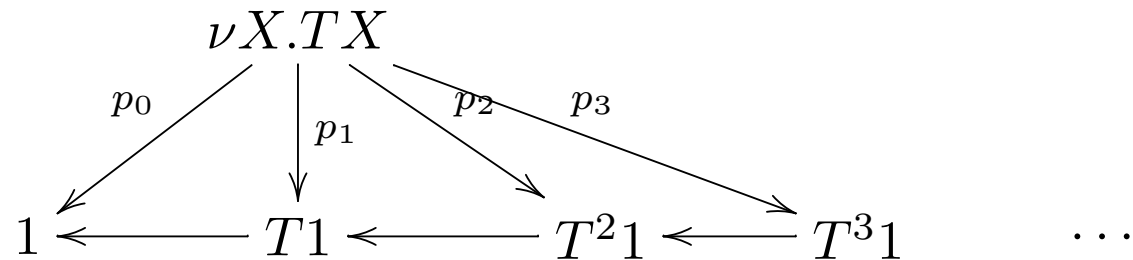
- represent higher-order functions as functions on trees
- *but:* domain doesn't fit into $\nu Z.(S \triangleleft P)Z = \nu Z. \sum_{s \in S} Z^{P(s)}$

Topological Excursion

Question. What's the natural topology on $A^\omega \rightarrow B^\omega$?

Topology on Representatives $R = \nu X.\mu Y.B \times X + (A \rightarrow Y)$.

- consider $TX = \mu Y.B \times X + (A \rightarrow Y)$ and $\sigma : R \rightarrow TR$
- topology given by the inverse limit



where $p_{i+1} = Tp_i \circ \sigma$. Topology generated by $p_i^{-1}(o)$, $o \subseteq T^i 1$ open

Induced Topology on $(A^\omega \rightarrow B^\omega)$ is compact-open:

- elements of $T^n 1$ are layers of A -branching trees with labels in B
- single trees define compact-open constraints

Container Magic

Summary so far.

- have representation of functions $\nu Z.(S \triangleleft P)Z \rightarrow X$
- want: representations of $\nu X.\mu Y.(B \times X) + (A \rightarrow Y) \rightarrow X$

Container Translation. Representations for free – if we solve

$$\mu Y.(B \times X) + (A \rightarrow Y) = (S \triangleleft P)X$$

Theorem. (Abbot/Altenkirch/Ghani) Containers are closed under μ, ν .

More precisely: for every n -ary container

$$C(X_1, \dots, X_n) = \sum_{s \in S} X_1^{P_1(s)} \times \dots \times X_n^{P_n(s)}$$

there is an $n - 1$ -ary container $D(X_1, \dots, X_{n-1})$ that satisfies

$$D(X_1, \dots, X_{n-1}) = \mu X_n.C(X_1, \dots, X_n)$$

Order-Two Example

Representatives. (Recall: $S = \mu X.B + (A \rightarrow X)$ and $P(s) = \text{paths}$)

$$R = \nu F.\mu G.\Lambda I.C \times F(I) + I \times \prod_{s \in S} (I + P(s))$$

Unfolding Isomorphisms. (Recall: $R(I)$ represents $(A^\omega \rightarrow B^\omega)^I \rightarrow C^\omega$)

$$R(I) \cong C \times R(I) + I \times \prod_{s \in S} R(I + P(s))$$

Induced Representation.

$$\text{eat}(I) : \quad R \quad \rightarrow T^I \quad \rightarrow \nu Z.C \times Z$$

$$\text{eat}(I) \quad (\text{Ret}(c, r)) \quad (\phi) \quad = c : (\text{eat } r \phi)$$

$$\text{eat}(I) \quad (\text{Rd}(i, f)) \quad (\phi) \quad = \text{eat} (f(\text{root } \phi(i))) [\phi, \text{debris}(\phi(i))]$$

Notation. For $t = (r, d) \in T = \nu X.(S \triangleleft P)X \cong \sum_{s \in S} T^{P(s)}$

$$\text{root}(r, d) = r \quad \text{and} \quad \text{debris}(r, d) = d$$

Conclusions

Tree Eating.

- linear structures (streams) \rightsquigarrow family of inductive types
- nonlinear structures (trees) \rightsquigarrow inductive families of types
- in both cases: sound and complete representation of continuous functions

Higher Order Functions.

- reducible to tree case – but with coding
- possibly very inefficient in practice – try out

Open Questions.

- more combinators (e.g. buffering, currying)
- concrete case studies – in particular integration
- complexity of (higher order) stream functions?