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)
Infinite Objects are Coalgebras

Infinite Streams over $A$: $\nu X.A \times X$

$$a_0 \to a_1 \to a_2 \to \ldots$$

Infinite Binary Trees over $A$:
$$\nu X.A \times X^2$$

Signatures (variable branching):
$$\nu X.A \times X^2 + B \times X + C$$
Goal. Algebraic Treatment of continuous functions \( \nu T \to \nu S \)

- e.g. representatives of reals: \( \{-1, 0, 1\}^\omega \sim [-1, 1] \)
- clean (co)inductive definitions and proofs

Discrete Codomain. Continuous Functions \( f : \nu X.TX \to B \)

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

Example. Infinite Streams, or coalgebraically \( \nu X.A \times X \to B \)

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

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

- generated by \( \overline{\alpha} \cdot A^\omega \) where \( \overline{\alpha} \in A^* \)
Coalgebraic View

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

\[
A^\omega = \nu X. A \times X
\]

\[
\begin{array}{c}
1 \leftarrow A \leftarrow A^2 \leftarrow A^2 \leftarrow \cdots
\end{array}
\]

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

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

\[
\nu X.TX
\]

\[
\begin{array}{c}
1 \leftarrow T^1 \leftarrow T^2 \leftarrow T^3 \leftarrow \cdots
\end{array}
\]

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 $A^\omega \to B$ with $B$ discrete.

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

- iff $f$ locally constant.
  \[
  (\forall (a_0, a_1, \ldots) \in A^\omega)(\exists n \in \omega) f \text{ constant on } (a_0, a_1, \ldots, a_n) \cdot A^\omega
  \]

- iff $f$ is in the least class $C$ closed under

\[
\begin{array}{c}
\text{f constant} \\
\hline
\text{C(f)}
\end{array}, \quad \begin{array}{c}
(\forall a \in A)C(f(a : _)) \\
\hline
C(f)
\end{array}
\]

Proof ($\iff$) 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

\[
\begin{array}{cc}
\text{\(f\) constant} & \quad \text{\((\forall a \in A) C(f(a : \_))\)} \\
C(f) & \quad C(f)
\end{array}
\]

and can be represented as an inductive data type:

\[
R = \mu X. B + (A \to X) \cong B + (A \to R)
\]

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

from which a continuous function can be extracted:

\[
eat: \quad \mu X. B + (A \to X) \to A^\omega \to B
\]

\[
eat (\text{Ret} b) (a : \alpha) = b
\]

\[
eat (\text{Rd} f) (a : \alpha) = \text{eat}(f \ a) \alpha
\]

Theorem. If \(\to_c\) is continuous functions, then \(\text{eat} : R \to (A^\omega \to_c B)\) 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 $\text{Rd}$ and $\text{Ret}$.

$$\text{eat} : \quad R \quad \rightarrow \quad \text{Tree}(A) \quad \rightarrow \quad B$$

$$\text{eat} \quad (\text{Ret} \ b) \quad (a, l, r) \quad = \quad b$$

$$\text{eat} \quad (\text{Rd} \ f) \quad (a, l, r) \quad = \quad \text{eat}(f a)(l, r)$$

**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 $\text{Tree}(A)^n \rightarrow B$

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

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

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

where $l, r$ are the left/right subtree of $t_i$. **Constructors.** Ret, Rd$_1$, \ldots, Rd$_n$
Escaping the Underworld of Indices

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

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

**Realisation Mapping.**

\[
eat_n : R_n \rightarrow \text{Tree}(A)^n \rightarrow B
\]

\[
eat_n (\text{Ret } b) (t_1, \ldots, t_n) = b
\]

\[
eat_n (\text{Rd}_i f) (t_1, \ldots, t_n) = \eat_{n+1}(f a_i)(t_1, \ldots, t_{i-1}, l, r, t_{i+1}, \ldots, t_n)
\]

**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

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

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

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

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

**Container Functors.** (Abbot, Altenkirch, Ghani)

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

- \(S : \text{Set}\) 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.**

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

**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

\[
\begin{align*}
(R) & \quad \forall a (C(f(a : _))) \\
(W) & \quad C(f) \wedge C(\lambda \alpha. b : f(\alpha))
\end{align*}
\]

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 $\mu$

\[
R \sqsubseteq \nu X. \mu Y. B \times X + (A \rightarrow Y) \sqsubseteq B \times R + (A \rightarrow R)
\]

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

Extracted Continuous Function.

\[
eat : \quad \nu X. \mu Y. B \times X + (A \rightarrow Y) \rightarrow A^\omega \rightarrow B^\omega
\]

\[
eat (\text{Ret}(b, r)) (a : \alpha) = b : \text{eat} \ r (a : \alpha)
\]

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

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 $A^\omega \to \nu X. (S \triangleleft P) X = \nu X. \sum_{s \in S} X^{P(s)}$

**By Analogy.**

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

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

**Associated Functional.**

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

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

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

**Observation.**

- *codomain* just “slots in”, more general *domains* by same recipe
Example. Continuous Stream Functions

\[ f : A^\omega \rightarrow_c B^\omega \]

are represented by

\[ \nu X. \mu Y B \times X + Y^A \]

\[ P_A(B) \]

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

$$P_B(C) \times P_A(B) \rightarrow P_A(C)$$

$$\downarrow \quad \downarrow$$

$$(B^\omega \rightarrow_c C^\omega) \times (A^\omega \rightarrow_c B^\omega) \rightarrow (A^\omega \rightarrow_c C^\omega)$$

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 \textit{inductively defined} map between \textit{coinductive} types
More on Composition

*Inductive* definition of composition

\[ \gamma : S = T_B(C \times P_B C) \times T_A(B \times P_A B) \to T_A(C \times S) \]

\[ \langle \text{Ret} \langle c, p_{bc} \rangle , t_{ab} \rangle \mapsto \text{Ret} \langle c, \text{out } p_{bc}, t_{ab} \rangle \]

\[ \langle \text{Rd } \phi \rangle \mapsto \text{Rd} \langle b, p_{ab} \rangle \mapsto \gamma \langle \phi \ b, \text{out } p_{ab} \rangle \]

\[ \langle t_{bc} \rangle \mapsto \text{Rd } \langle \psi \rangle \mapsto \text{Rd } \lambda a. \gamma \langle t_{bc}, \psi \ a \rangle \]

whose *coinductive* cousin (\text{out} : \nu F \to F(\nu F))

\[ \chi : P_B(C) \times P_A(B) \to P_A(C) \]

\[ \langle \text{post} , \text{pre} \rangle \mapsto (\text{unfold } \gamma) \langle \text{out } \text{post} , \text{out } \text{pre} \rangle \]

represents composition.

This is *output centered* – alternatives are possible.
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 $\nu Z. (S \triangleleft P) Z = \nu Z. \sum_{s \in S} Z^{P(s)}$
**Question.** What’s the natural topology on $A^\omega \to B^\omega$?

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

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

![Diagram](attachment:tree_diagram.png)

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

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

- elements of $T^n 1$ are layers of $A$-branching trees with labels in $B$
- single trees define compact-open constraints
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/Alternkirch/Ghani) Containers are closed under $\mu, \nu$.

More precisely: for every $n$-ary container

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

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

$$D(X_1, \ldots, X_{n-1}) = \mu X_n.C(X_1, \ldots, X_n)$$
**Wanted.** Solutions of

$$\mu Y.B \times X + (A \to Y) \cong (S \triangle P)X = \sum_{s \in S} X^{P(s)}$$

**Observation.** We see *trees* with payload at the leaves.

---

**Shapes.**

$$S = \mu X.B + (A \to X)$$

**Positions.**

$$P(s) = \{ \text{paths in } S \text{ from root to leaves} \}$$
Order-Two Example

Representatives. (Recall: $S = \mu X.B + (A \to X)$ and $P(s) =$ paths )

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

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

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

Induced Representation.

$$\begin{align*}
eat(I) : & \quad R & \to T^I & \to \nu Z.C \times Z \\
eat(I) & (\Ret(c, r)) & (\phi) & = c : (\text{eat } r \phi) \\
eat(I) & (\Rd(i, f)) & (\phi) & = \text{eat } (f(\text{root } \phi(i))) \ [\phi, \text{debris}(\phi(i))] \\
\end{align*}$$

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?