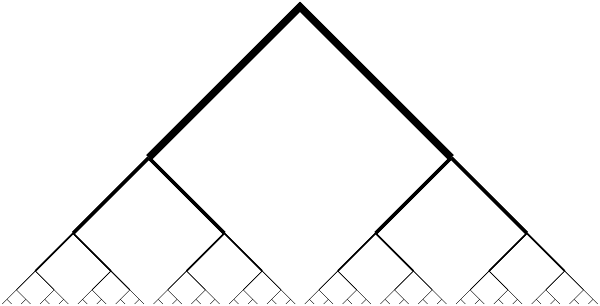
# M-types and Bisimulation



**Henning Basold and Daniël Otten**

# Introduction

Motivation:

- Induction is well-studied but not all results translate to coinduction.
- Induction principle uses dependent functions which we can't dualise.
- Adding $\eta$-reduction for streams makes type checking undecidable:

$$s \coloneqq \text{head } s :: \text{tail } s.$$

We study known definitions of the coinduction principle on $\mathsf{M}$-types: how do we formulate them and under which assumptions are they equivalent?

# Conventions

The polynomial functor for $A : \text{Type}$ and $B : A \to \text{Type}$ is given by:

$$P\ X :\equiv \textstyle\sum_{a:A}(B\,a \to X),$$
$$P\ f\ :\equiv \lambda(a, c).\,(a, f \circ c).$$

The $P$-coalgebras are given by:

$$\text{Coalg} :\equiv \textstyle\sum_{X:\text{Type}}(X \to P\,X),$$
$$\text{CoalgMor}\ (X, d)\ (Y, e) :\equiv \textstyle\sum_{f:X\to Y}(e \circ f \equiv P\,f \circ d).$$

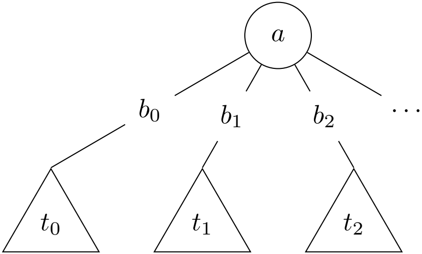
$$\begin{tikzcd} X \arrow[d, "d"'] \\ P\,X \end{tikzcd} \qquad \begin{tikzcd} X \arrow[r, "f"] \arrow[d, "d"'] & Y \arrow[d, "e"] \\ P\,X \arrow[r, "P\,f"'] & P\,Y \end{tikzcd}$$

# $\mathsf{M}$-type

Intuitively, the $\mathsf{M}$-type for $A : \mathrm{Type}$ and $B : A \to \mathrm{Type}$ is

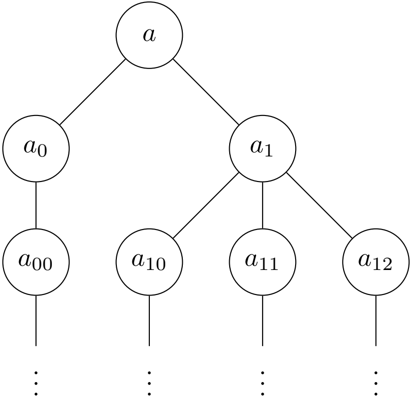
- the type of potentially infinite trees
- whose nodes each have a label $a : A$ and
- precisely one child for every $b : B\,a$.

This gives rise to a coalgebra $\mathrm{des} : \mathsf{M} \to P\,\mathsf{M}$.



# Coinduction

For any coalgebra $d : X \to P\,X$ and $x : X$ we get a tree:




This gives a unique morphism $(f, c) : \mathrm{CoalgMor}\,(X, d)\,(\mathsf{M}, \mathrm{des})$.

# Final $\mathsf{M}$-type

For $(\mathsf{M}, \mathrm{des}) : \mathrm{Coalg}$ we define:

$$\mathrm{IsFinM}\,(\mathsf{M}, \mathrm{des}) \coloneqq \prod_{(X,d):\mathrm{Coalg}} \mathrm{IsContr}\,(\mathrm{CoalgMor}\,(X, d)\,(\mathsf{M}, \mathrm{des})).$$

$$(X, d) \overset{!(f,c)}{\dashrightarrow} (\mathsf{M}, \mathrm{des})$$

# Coherent $\mathsf{M}$-type

For $(\mathsf{M}, \mathrm{des}) : \mathrm{Coalg}$ we define:

$$
\begin{aligned}
\mathrm{IsCohM}\,(\mathsf{M}, \mathrm{des}) \coloneqq\; & \textstyle\prod_{(X,d):\mathrm{Coalg}} \\
& \mathrm{CoalgMor}\,(X, d)\,(\mathsf{M}, \mathrm{des}) \times \\
& \textstyle\prod_{(f_0,c_0),(f_1,c_1):\mathrm{CoalgMor}\,(X,d)\,(\mathsf{M},\mathrm{des})} \\
& \textstyle\sum_{p:(f_0 \equiv f_1)} \prod_{x:X} \text{(the diagram commutes)}
\end{aligned}
$$

$$
\begin{tikzcd}
\mathrm{des}\,(f_0\,x) \arrow[rr, "\text{cong-app}\,c_0\,x"] \arrow[d, "\mathrm{cong}\,(\lambda f.\,\mathrm{des}\,(f\,x))\,p"'] & & P\,f_0\,(d\,x) \arrow[d, "\mathrm{cong}\,(\lambda f.\,P\,f\,(d\,x))\,p"] \\
\mathrm{des}\,(f_1\,x) \arrow[rr, "\text{cong-app}\,c_1\,x"'] \arrow[rru, phantom, "\equiv"] & & P\,f_1\,(d\,x)
\end{tikzcd}
$$

# Span Bisimulation

For $(X, d)$ : Coalg we define:

$$\mathrm{SpanBisim}\,(X, d) :\equiv \sum_{(R,b):\mathrm{Coalg}} (\mathrm{CoalgMor}\,(R, b)\,(X, d))^2$$

$$(X, d) \xleftarrow{(\rho_0, c_0)} (R, b) \xrightarrow{(\rho_1, c_1)} (X, d)$$

We can view $((R, b), (\rho_0, c_0), (\rho_1, c_1))$ : $\mathrm{SpanBisim}\,X$ as a relation on $X$:

$$x_0 \sim x_1 :\equiv \sum_{r:R} ((\rho_0\, r \equiv x_0) \times (\rho_1\, r \equiv x_1)).$$

Propositional equality is a bisimulation:

$$\equiv_{(X,d)} :\equiv ((X, d), (\mathrm{id}, \mathrm{refl}), (\mathrm{id}, \mathrm{refl})).$$

# Span Bisimulation Morphism

For $(X, d) : \mathrm{Coalg}$ and $\sim, \approx : \mathrm{SpanBisim}\,(X, d)$ we define:

$$\mathrm{SpanBisimMor} \sim \approx :\equiv \sum_{(f,c):\mathrm{CoalgMor}\,(R,b)\,(S,c)} (\text{the diagram commutes}).$$

$$\begin{tikzcd}
 & (R, b) \arrow[dl, "(\rho_0, c_0)"'] \arrow[dr, "(\rho_1, c_1)"] \arrow[dd, "(f,c)"] & \\
(X, d) & & (X, d) \\
 & (S, c) \arrow[ul, "(\sigma_0, d_0)"] \arrow[ur, "(\sigma_1, d_1)"'] &
\end{tikzcd}$$

In particular this is an inclusion of the relations.

# Span Bisimilarity $\mathsf{M}$-type

For $(\mathsf{M}, \mathrm{des}) : \mathrm{Coalg}$ we define:

$$\begin{aligned}\mathrm{IsSpanBisimM}\,(\mathsf{M}, \mathrm{des}) \coloneqq\ & \left(\prod_{(X,d):\mathrm{Coalg}} \mathrm{CoalgMor}\,(X, d)\,(\mathsf{M}, \mathrm{des})\right) \times \\ & \prod_{(\sim:\mathrm{SpanBisim}\,(\mathsf{M},\mathrm{des}))} \\ & \mathrm{IsContr}\,(\mathrm{SpanBisimMor}\ \sim\ \equiv_{(X,d)})\end{aligned}$$

$$\begin{tikzcd}
 & (R, b) \arrow[dl, "{(\rho_0, c_0)}"'] \arrow[dr, "{(\rho_1, c_1)}"] \arrow[dd, dashed, "{!(f, c)}"] & \\
(\mathsf{M}, \mathrm{des}) & & (\mathsf{M}, \mathrm{des}) \\
 & (\mathsf{M}, \mathrm{des}) \arrow[ul, "{(\mathrm{id}, \mathrm{refl})}"] \arrow[ur, "{(\mathrm{id}, \mathrm{refl})}"'] &
\end{tikzcd}$$

# Implications

We have the following functions:

$$
\begin{tikzcd}
\mathrm{IsFinM}\,(\mathsf{M}, \mathrm{des}) \arrow[r, shift left] \arrow[d, shift left] & \mathrm{IsSpanBisimM}\,(\mathsf{M}, \mathrm{des}) \arrow[l, shift left] \\
\mathrm{IsCohM}\,(\mathsf{M}, \mathrm{des}) \arrow[u, shift left, "\mathrm{funext}"] \arrow[r] & \mathrm{IsLiftingBisimM}\,(\mathsf{M}, \mathrm{des})
\end{tikzcd}
$$

- The arrow marked with ‘funext’ uses function extensionality.
- IsLiftingBisimM would have to add coherences to be equivalent.
- These results have largely been formalised in `Agda`.