Unfolding Nested Patterns and Copatterns

Anton Setzer

(Swansea, UK)

Types 2013 Toulouse
Thursday, 25 April 2013

Theorem Regarding Undecidabilty of Equality

Theorem

Assume the following:

- *There exists a decidable subset* $\mathrm{Stream} \subseteq \mathbb{N}$,
- *computable functions*
  $\mathrm{head} : \mathrm{Stream} \to \mathbb{N}$, $\mathrm{tail} : \mathrm{Stream} \to \mathrm{Stream}$,
- *a decidable equality* $\_ == \_$ *on* Stream *which is congruence,*
- *the possibilty to define elements of* Stream *by guarded recursion based on primitive recursive functions* $f, g : \mathbb{N} \to \mathbb{N}$, *such that the standard equalities related to guarded recursion hold.*

Then it is not possible to fulfil the following condition:

$$\forall s, s' : \mathrm{Stream}.\mathrm{head}\ s = \mathrm{head}\ s' \wedge \mathrm{tail}\ s == \mathrm{tail}\ s' \to s == s' \qquad (*)$$

Consequences for Codata Approach

Remark

Condition $(*)$ *is fulfilled if we have an operation*
$\mathrm{cons} : \mathbb{N} \to \mathrm{Stream} \to \mathrm{Stream}$ *preserving equalities s.t.*

$$\forall s : \mathrm{Stream}. s = \mathrm{cons}\ (\mathrm{head}\ s)\ (\mathrm{tail}\ s)$$

So we cannot have a type theory with streams, decidable type checking and decidable equality on streams such that

$$\forall s. \exists n, s'. s == \mathrm{cons}\ n\ s'$$

as assumed by the codata approach.

Proof of Theorem

- Assume we had the above.
- By

$$s \approx n_0 :: n_1 :: n_2 :: \cdots n_k :: s'$$

  we mean the equations using head, tail expressing that s behaves as the stream indicated on the right hand side.
- Define by guarded recursion $l : \mathrm{Stream}$

$$l \approx 1 :: 1 :: 1 :: \cdots$$

Proof of Theorem

- For e code for a Turing machine define by guarded recursion based on primitive recursion functions f, g s.t. if e terminates after n steps and returns result k then

$$f\ e \quad \approx \quad \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n \text{ times}} :: l$$

$$g\ e \quad \approx \quad \begin{cases} \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n \text{ times}} :: l & \text{if } k = 0 \\ \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n+1 \text{ times}} :: l & \text{if } k > 0 \end{cases}$$

Proof of Theorem

$$f\ e \quad \approx \quad \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n \text{ times}} :: l$$

$$g\ e \quad \approx \quad \begin{cases} \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n \text{ times}} :: l & \text{if } k = 0 \\ \underbrace{0 :: 0 :: 0 :: \cdots :: 0}_{n+1 \text{ times}} :: l & \text{if } k > 0 \end{cases}$$

- If e terminates after n steps with result 0 then

$$f\ e == g\ e$$

- If e terminates after n steps with result > 0 then

$$\neg(f\ e == g\ e)$$

Proof of Theorem

- So
$$\lambda e.(f\ e == g\ e)$$
separates the TM with result 0 from those with result > 0.
- But these two sets are inseparable.

Operators for Primitive (Co)Recursion

$$
\begin{array}{lcl}
\mathrm{P}_{\mathbb{N},A} : A \rightarrow (\mathbb{N} \rightarrow A \rightarrow A) \rightarrow \mathbb{N} \rightarrow A & & \\
\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_\mathrm{S}\ 0 & = & \mathrm{step}_0 \\
\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_\mathrm{S}\ (\mathrm{S}\ n) & = & \mathrm{step}_\mathrm{S}\ n\ (\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_\mathrm{S}\ n)
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{coP}_{\mathrm{Stream},A} : (A \rightarrow \mathbb{N}) \rightarrow (A \rightarrow (\mathrm{Stream} + A)) \rightarrow A \rightarrow \mathrm{Stream} & & \\
\mathrm{head}\ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_\mathrm{head}\ \mathrm{step}_\mathrm{tail}\ a) & = & \mathrm{step}_\mathrm{head}\ a \\
\mathrm{tail}\ \ \ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_\mathrm{head}\ \mathrm{step}_\mathrm{tail}\ a) & = & \\
\quad \mathrm{case}_{\mathrm{Stream},A,\mathrm{Stream}}\ \mathrm{id}\ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_\mathrm{head}\ \mathrm{step}_\mathrm{tail})\ (\mathrm{step}_\mathrm{tail}\ a) & &
\end{array}
$$

Example of Mixed Pattern and Copattern Matching

$$
\begin{array}{llll}
f : \mathbb{N} \to \mathrm{Stream} & & & \\
\mathrm{head} & (f\ 0\quad) & = & 0 \\
\mathrm{head}\ (\mathrm{tail} & (f\ 0\quad)) & = & 0 \\
\mathrm{tail}\ \ (\mathrm{tail} & (f\ 0\quad)) & = & f\ N \\
\mathrm{head} & (f\ (\mathrm{S}\ n)) & = & \mathrm{S}\ n \\
\mathrm{head}\ (\mathrm{tail} & (f\ (\mathrm{S}\ n))) & = & \mathrm{S}\ n \\
\mathrm{tail}\ \ (\mathrm{tail} & (f\ (\mathrm{S}\ n))) & = & f\ n
\end{array}
$$

This example can be reduced to primitive (co)recursion.
Step 1: Following the development of the (co)pattern matching definition, unfold it into simulteneous non-nested (co)pattern matching definitions.

$$
\begin{array}{lcllcl}
 & & f : \mathbb{N} \to \mathrm{Stream} & & & \\
 & & \mathrm{head} & (f\ n) & = & g\ n \\
 & & \mathrm{tail} & (f\ n) & = & h\ n \\
 & & & & & \\
 & & g : \mathbb{N} \to \mathbb{N} & & & \\
(\mathrm{head}\ (f\ 0) & =) & g\ 0 & & = & 0 \\
(\mathrm{head}\ (f\ (\mathrm{S}\ n)) & =) & g\ (\mathrm{S}\ n) & & = & \mathrm{S}\ n \\
 & & & & & \\
 & & h : \mathbb{N} \to \mathrm{Stream} & & & \\
(\mathrm{tail}\ (f\ 0) & =) & h\ 0 & & = & b_0 \\
(\mathrm{tail}\ (f\ (\mathrm{S}\ n)) & =) & h\ (\mathrm{S}\ n) & & = & h_0\ n \\
 & & & & & \\
 & & b_0 : \mathrm{Stream} & & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ 0)) & =) & \mathrm{head} & b_0 & = & 0 \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ 0)) & =) & \mathrm{tail} & b_0 & = & f\ N \\
 & & & & & \\
 & & h_0 : \mathbb{N} \to \mathrm{Stream} & & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & =) & \mathrm{head} & (h_0\ n) & = & \mathrm{S}\ n \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & =) & \mathrm{tail} & (h_0\ n) & = & f\ n
\end{array}
$$

Step 2: Reduction to Primitive (Co)recursion

- We can always after step 2 replace the recursion by full (co)recursion operators.
- Reduction to primitive (co)recursion – if it is possible – requires more work:
- First the functions f, b_0, h_0 defined by copattern matching can be defined simultaneously:

$$
\begin{array}{lll}
f : \mathbb{N} \to \mathrm{Stream} & & \\
f\ n & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n) \\
 & & \\
(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream} & & \\
\mathrm{head}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & g\ n \\
\mathrm{head}\quad ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & 0 \\
\mathrm{head}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & \mathrm{S}\ n \\
\mathrm{tail}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & h\ n \\
\mathrm{tail}\quad ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ N) \\
\mathrm{tail}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n) \\
 & & \\
g : \mathbb{N} \to \mathbb{N} & & \\
g\ 0 & = & 0 \\
g\ (\mathrm{S}\ n) & = & \mathrm{S}\ n \\
 & & \\
h : \mathbb{N} \to \mathrm{Stream} & & \\
h\ 0 & = & (f + b_0 + h_0)\ (\underline{\mathrm{b}_0}) \\
h\ (\mathrm{S}\ n) & = & (f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)
\end{array}
$$

Unfolding of the Pattern Matchings

- h has recursive calls allowed by primitive corecursion on Stream. We replace h by a function h' return the argument for the recursive call.

$$
\begin{array}{lll}
f : \mathbb{N} \to \mathrm{Stream} & & \\
f\ n & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)
\end{array}
$$

$$
\begin{array}{llcl}
\multicolumn{4}{l}{(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream}} \\
\mathrm{head} & ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & g\ n \\
\mathrm{head} & ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & 0 \\
\mathrm{head} & ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & \mathrm{S}\ n \\
\mathrm{tail} & ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & (\mathrm{id} + (f + b_0 + h_0))\ (h'\ n) \\
\mathrm{tail} & ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ N) \\
\mathrm{tail} & ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)
\end{array}
$$

$$
\begin{array}{lll}
g : \mathbb{N} \to \mathbb{N} & & \\
g\ 0 & = & 0 \\
g\ (\mathrm{S}\ n) & = & \mathrm{S}\ n
\end{array}
$$

$$
\begin{array}{lll}
\multicolumn{3}{l}{h' : \mathbb{N} \to (\underline{\mathrm{return}}(\mathrm{Stream}) + (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})))} \\
h'\ 0 & = & \underline{\mathrm{b}_0} \\
h'\ (\mathrm{S}\ n) & = & \underline{\mathrm{h}_0}\ n
\end{array}
$$

Replacement by Combinators

$$f : \mathbb{N} \to \mathrm{Stream}$$
$$f \quad = \quad \lambda n.(f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)$$

$$(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream}$$
$$\begin{aligned}(f + b_0 + h_0) = \\ \quad \mathrm{coP}_{\mathrm{Stream},(\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ &(\mathrm{case}_{(\underline{\mathrm{f}}(\mathbb{N})+(\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})))} \\ &\ \ g \\ &\ \ (\mathrm{case}_{\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})}\ (\lambda\_.0)\ \mathrm{S}))) \\ &(\mathrm{case}_{(\underline{\mathrm{f}}(\mathbb{N})+(\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})))} \\ &\ \ h' \\ &\ \ (\mathrm{case}_{\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})}\ (\lambda\_.\underline{\mathrm{f}}\ N)\ \underline{\mathrm{f}}))\end{aligned}$$

$$g : \mathbb{N} \to \mathbb{N}$$
$$g = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ 0\ (\lambda n, ih.\mathrm{S}\ n)$$

$$h' : \mathbb{N} \to (\underline{\mathrm{return}}(\mathbb{N}) + \underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N}))$$
$$h' = \mathrm{P}_{\mathbb{N},(\underline{\mathrm{return}}(\mathbb{N})+\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ \underline{\mathrm{b}_0}\ (\lambda n, ih.\underline{\mathrm{h}_0}\ n)$$

Conclusion

- Codata types make the assumption

$$\forall s : \text{Stream}.\exists n, s'.s = \text{cons}\ n\ s'$$

which cannot be combined with a decidable equality.
- One can reduce certain cases of recursive nested (co)pattern matching to primitive (co)recursion.
  - Systematic treatment needs still to be done.
  - Cases which can be reduced should be those to be accepted by a termination checker.
  - If the reduction succeeds we get a normalising version (by Mendler and Geuvers).
  - Therefore a termination checked version of the calculus is normalising.

Example of Mixed Pattern and Copattern Matching

We consider operators for full and primitive (co)recursion:

$$
\begin{array}{lcl}
\mathrm{P}_{\mathbb{N},A} : A \to (\mathbb{N} \to A \to A) \to \mathbb{N} \to A & & \\
\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}\ 0 & = & \mathrm{step}_0 \\
\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}\ (\mathrm{S}\ n) & = & \mathrm{step}_{\mathrm{S}}\ n\ (\mathrm{P}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}\ n)
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{R}_{\mathbb{N},A} : ((\mathbb{N} \to A) \to A) \to ((\mathbb{N} \to A) \to \mathbb{N} \to A) \to \mathbb{N} \to A & & \\
\mathrm{R}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}\ 0 & = & \mathrm{step}_0\ (\mathrm{R}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}) \\
\mathrm{R}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}}\ (\mathrm{S}\ n) & = & \mathrm{step}_{\mathrm{S}}\ (\mathrm{R}_{\mathbb{N},A}\ \mathrm{step}_0\ \mathrm{step}_{\mathrm{S}})\ n
\end{array}
$$

Operators for full/primitive (co)recursion

$$
\begin{array}{l}
\mathrm{coP}_{\mathrm{Stream},A} : (A \to \mathbb{N}) \to (A \to (\mathrm{Stream} + A)) \to A \to \mathrm{Stream} \\
\mathrm{head}\ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}}\ a) \quad = \quad \mathrm{step}_{\mathrm{head}}\ a \\
\mathrm{tail}\ \ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}}\ a) \quad = \\
\quad \mathrm{case}_{\mathrm{Stream},A,\mathrm{Stream}}\ \mathrm{id}\ (\mathrm{coP}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}})\ (\mathrm{step}_{\mathrm{tail}}\ a)
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{coR}_{\mathrm{Stream},A} : ((A \to \mathrm{Stream}) \to A \to \mathbb{N}) & & \\
\qquad \to ((A \to \mathrm{Stream}) & & \\
\qquad \to A \to \mathrm{Stream}) \to \mathrm{Stream} & & \\
\mathrm{head}\ (\mathrm{coR}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}}\ a) & = & \mathrm{step}_{\mathrm{head}} \\
 & & (\mathrm{coR}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}})\ a \\
\mathrm{tail}\ (\mathrm{coR}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}}\ a) & = & \mathrm{step}_{\mathrm{tail}} \\
 & & (\mathrm{coR}_{\mathrm{Stream},A}\ \mathrm{step}_{\mathrm{head}}\ \mathrm{step}_{\mathrm{tail}})\ a
\end{array}
$$

Step 1: Unnesting of Nested (Co)Pattern Matching

We follow the steps in the pattern matching:
We start with

$$
\begin{array}{l}
f : \mathbb{N} \to \mathrm{Stream} \\
\begin{array}{lllll}
\mathrm{head} & (f \ n) & = & ? \\
\mathrm{tail} & (f \ n) & = & ?
\end{array}
\end{array}
$$

Pattern matching on first n:

$$\begin{array}{lcl} f : \mathbb{N} \to \mathrm{Stream} & & \\ \mathrm{head}\ (f\ 0) & = & ? \\ \mathrm{head}\ (f\ (\mathrm{S}\ n)) & = & ? \\ \mathrm{tail}\ \ (f\ n) & = & ? \end{array}$$

corresponds to

$$\begin{array}{lclcl} & & f : \mathbb{N} \to \mathrm{Stream} & & \\ & & \mathrm{head}\ \ (f\ n) & = & g\ n \\ & & \mathrm{tail}\ \ \ \ (f\ n) & = & ? \\ & & & & \\ & & g : \mathbb{N} \to \mathbb{N} & & \\ (\mathrm{head}\ (f\ 0) & =) & g\ 0 & = & ? \\ (\mathrm{head}\ (f\ (\mathrm{S}\ n)) & =) & g\ (\mathrm{S}\ n) & = & ? \end{array}$$

Pattern matching on second $n : \mathbb{N}$:

$$\begin{array}{lll} f : \mathbb{N} \to \mathrm{Stream} & & \\ \mathrm{head}\ (f\ 0) & = & ? \\ \mathrm{head}\ (f\ (\mathrm{S}\ n)) & = & ? \\ \mathrm{tail}\ \ \ (f\ 0) & = & ? \\ \mathrm{tail}\ \ \ (f\ (\mathrm{S}\ n)) & = & ? \end{array}$$

corresponds to

$$\begin{array}{lll} f : \mathbb{N} \to \mathrm{Stream} & & \\ \mathrm{head}\ \ (f\ n) & = & g\ n \\ \mathrm{tail}\ \ \ \ (f\ n) & = & h\ n \end{array}$$

$$\begin{array}{llll} & g : \mathbb{N} \to \mathbb{N} & & \\ (\mathrm{head}\ (f\ 0) & =)\ \ g\ 0 & = & ? \\ (\mathrm{head}\ (f\ (\mathrm{S}\ n)) & =)\ \ g\ (\mathrm{S}\ n) & = & ? \end{array}$$

$$\begin{array}{llll} & h : \mathbb{N} \to \mathrm{Stream} & & \\ (\mathrm{tail}\ (f\ 0) & =)\ \ h\ 0 & = & ? \\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n)) & =)\ \ h\ (\mathrm{S}\ n) & = & ? \end{array}$$

Copattern matching on $\mathrm{tail}\ (f\ 0) : \mathrm{Stream}$

$$
\begin{array}{llll}
f : \mathbb{N} \to \mathrm{Stream} & & & \\
\mathrm{head} & (f\ 0) & = & ? \\
\mathrm{head} & (f\ (\mathrm{S}\ n)) & = & ? \\
\mathrm{head}\ (\mathrm{tail} & (f\ 0)) & = & ? \\
\mathrm{tail}\ (\mathrm{tail} & (f\ 0)) & = & ? \\
\mathrm{tail} & (f\ (\mathrm{S}\ n)) & = & ?
\end{array}
$$

which corresponds to

$$
\begin{array}{llll}
 & f : \mathbb{N} \to \mathrm{Stream} & & \\
 & \mathrm{head} \quad (f \; n) & = & g \; n \\
 & \mathrm{tail} \quad\;\; (f \; n) & = & h \; n \\
 & & & \\
 & g : \mathbb{N} \to \mathbb{N} & & \\
(\mathrm{head}\ (f\ 0) & =) \quad g\ 0 & = & ? \\
(\mathrm{head}\ (f\ (\mathrm{S}\ n)) & =) \quad g\ (\mathrm{S}\ n) & = & ? \\
 & & & \\
 & h : \mathbb{N} \to \mathrm{Stream} & & \\
(\mathrm{tail}\ (f\ 0) & =) \quad h\ 0 & = & b_0 \\
(\mathrm{tail}\ (f\ (\mathrm{S}\ n)) & =) \quad h\ (\mathrm{S}\ n) & = & ? \\
 & & & \\
 & b_0 : \mathrm{Stream} & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ 0)) & =) \quad \mathrm{head} \quad b_0 & = & ? \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ 0)) & =) \quad \mathrm{tail} \quad\;\; b_0 & = & ? \\
\end{array}
$$

Copattern matching on $\text{tail}\ (f\ (\text{S}\ n)) : \text{Stream}$:

$$
\begin{array}{lllcl}
f : \mathbb{N} \to \text{Stream} \\
\text{head} & & (f\ 0\quad\ ) & = & ? \\
\text{head} & & (f\ (\text{S}\ n)) & = & ? \\
\text{head} & (\text{tail} & (f\ 0\quad\ )) & = & ? \\
\text{tail} & (\text{tail} & (f\ 0\quad\ )) & = & ? \\
\text{head} & (\text{tail} & (f\ (\text{S}\ n))) & = & ? \\
\text{tail} & (\text{tail} & (f\ (\text{S}\ n))) & = & ?
\end{array}
$$

which corresponds to

$$
\begin{array}{llll}
 & f : \mathbb{N} \to \mathrm{Stream} & & \\
 & \mathrm{head} \quad (f\ n) & = & g\ n \\
 & \mathrm{tail} \quad (f\ n) & = & h\ n \\
 & & & \\
 & g : \mathbb{N} \to \mathbb{N} & & \\
(\mathrm{head}\ (f\ 0) & =)\quad g\ 0 & = & ? \\
(\mathrm{head}\ (f\ (\mathrm{S}\ n)) & =)\quad g\ (\mathrm{S}\ n) & = & ? \\
 & & & \\
 & h : \mathbb{N} \to \mathrm{Stream} & & \\
(\mathrm{tail}\ (f\ 0) & =)\quad h\ 0 & = & b_0 \\
(\mathrm{tail}\ (f\ (\mathrm{S}\ n)) & =)\quad h\ (\mathrm{S}\ n) & = & h_0\ n \\
 & & & \\
 & b_0 : \mathrm{Stream} & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ 0)) & =)\quad \mathrm{head} \quad b_0 & = & ? \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ 0)) & =)\quad \mathrm{tail} \quad b_0 & = & ? \\
 & & & \\
 & h_0 : \mathbb{N} \to \mathrm{Stream} & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & =)\quad \mathrm{head} \quad (h_0\ n) & = & ? \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & =)\quad \mathrm{tail} \quad (h_0\ n) & = & ?
\end{array}
$$

Resolving the goals:

$$
\begin{array}{lcl}
f : \mathbb{N} \to \mathrm{Stream} & & \\
\mathrm{head}\ (f\ 0) & = & 0 \\
\mathrm{head}\ (\mathrm{tail}\ (f\ 0)) & = & 0 \\
\mathrm{tail}\ (\mathrm{tail}\ (f\ 0)) & = & f\ N \\
\mathrm{head}\ (f\ (\mathrm{S}\ n)) & = & \mathrm{S}\ n \\
\mathrm{head}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & = & \mathrm{S}\ n \\
\mathrm{tail}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & = & f\ n
\end{array}
$$

which corresponds to

$$
\begin{array}{llllll}
 & & f : \mathbb{N} \to \mathrm{Stream} & & & \\
 & & \mathrm{head} & (f\ n) & = & g\ n \\
 & & \mathrm{tail} & (f\ n) & = & h\ n \\
 & & & & & \\
 & & g : \mathbb{N} \to \mathbb{N} & & & \\
(\mathrm{head}\ (f\ 0) & \Longrightarrow & g\ 0 & & = & 0 \\
(\mathrm{head}\ (f\ (\mathrm{S}\ n)) & \Longrightarrow & g\ (\mathrm{S}\ n) & & = & \mathrm{S}\ n \\
 & & & & & \\
 & & h : \mathbb{N} \to \mathrm{Stream} & & & \\
(\mathrm{tail}\ (f\ 0) & \Longrightarrow & h\ 0 & & = & b_0 \\
(\mathrm{tail}\ (f\ (\mathrm{S}\ n)) & \Longrightarrow & h\ (\mathrm{S}\ n) & & = & h_0\ n \\
 & & & & & \\
 & & b_0 : \mathrm{Stream} & & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ 0)) & \Longrightarrow & \mathrm{head} & b_0 & = & 0 \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ 0)) & \Longrightarrow & \mathrm{tail} & b_0 & = & f\ N \\
 & & & & & \\
 & & h_0 : \mathbb{N} \to \mathrm{Stream} & & & \\
(\mathrm{head}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & \Longrightarrow & \mathrm{head} & (h_0\ n) & = & \mathrm{S}\ n \\
(\mathrm{tail}\ (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & \Longrightarrow & \mathrm{tail} & (h_0\ n) & = & f\ n
\end{array}
$$

Step 2: Reduction to Primitive (Co)recursion

- This can now easily be reduced to full (co)recursion.
- In this example we can reduce it to primitive (co)recursion:
- First all functions which are defined by copattern matching on Stream can be defined simultaneously:

$$
\begin{array}{lll}
f : \mathbb{N} \to \mathrm{Stream} & & \\
\mathrm{head} \quad (f\ n) & = & g\ n \\
\mathrm{tail} \quad (f\ n) & = & h\ n
\end{array}
$$

$$
\begin{array}{lll}
b_0 : \mathrm{Stream} & & \\
\mathrm{head} \quad b_0 & = & 0 \\
\mathrm{tail} \quad b_0 & = & f\ N
\end{array}
$$

$$
\begin{array}{lll}
h_0 : \mathbb{N} \to \mathrm{Stream} & & \\
\mathrm{head} \quad (h_0\ n) & = & \mathrm{S}\ n \\
\mathrm{tail} \quad (h_0\ n) & = & f\ n
\end{array}
$$

$$
\begin{array}{lll}
g : \mathbb{N} \to \mathbb{N} & & \\
g\ 0 & = & 0 \\
g\ (\mathrm{S}\ n) & = & \mathrm{S}\ n
\end{array}
$$

$$
\begin{array}{lll}
h : \mathbb{N} \to \mathrm{Stream} & & \\
h\ 0 & = & b_0 \\
h\ (\mathrm{S}\ n) & = & h_0\ n
\end{array}
$$

Reduction to Primitive (Co)recursion

- Now these functions can be defined as one function:

$$
\begin{aligned}
&f : \mathbb{N} \to \mathrm{Stream} \\
&f\ n && = && (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n) \\
\\
&(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b_0}} + \underline{\mathrm{h_0}}(\mathbb{N})) \to \mathrm{Stream} \\
&\mathrm{head}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) && = && g\ n \\
&\mathrm{head}\quad ((f + b_0 + h_0)\ \underline{\mathrm{b_0}}) && = && 0 \\
&\mathrm{head}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{h_0}}\ n)) && = && \mathrm{S}\ n \\
&\mathrm{tail}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) && = && h\ n \\
&\mathrm{tail}\quad ((f + b_0 + h_0)\ \underline{\mathrm{b_0}}) && = && (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ N) \\
&\mathrm{tail}\quad ((f + b_0 + h_0)\ (\underline{\mathrm{h_0}}\ n)) && = && (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n) \\
\\
&g : \mathbb{N} \to \mathbb{N} \\
&g\ 0 && = && 0 \\
&g\ (\mathrm{S}\ n) && = && \mathrm{S}\ n \\
\\
&h : \mathbb{N} \to \mathrm{Stream} \\
&h\ 0 && = && (f + b_0 + h_0)\ (\underline{\mathrm{b_0}}) \\
&h\ (\mathrm{S}\ n) && = && (f + b_0 + h_0)\ (\underline{\mathrm{h_0}}\ n)
\end{aligned}
$$

Unfolding of the Pattern Matchings

- g can be defined by primitive recursion.
- The call of h has result always of the form $(f + b_0 + h_0)(n)$.
  So we can replace the recursive call $h\ n$ by $(f + b_0 + h_0)(h'\ n)$.
- However, since primitive corecursion allows as well escaping we replace it by a recursive call

  $$(\mathrm{id} + (f + b_0 + h_0))(h'\ n)$$

  with

  $$h' : \mathbb{N} \to \underline{\mathrm{return}}(\mathrm{Stream}) + (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N}))$$
- In general one would need of course continue
  - nested pattern matching needs to be replaced by simultaneous primitive recursion,

$$f : \mathbb{N} \to \mathrm{Stream}$$
$$f\ n \quad = \quad (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)$$

$$(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream}$$
$$\begin{array}{lllll}
\mathrm{head} & ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & g\ n \\
\mathrm{head} & ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & 0 \\
\mathrm{head} & ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & \mathrm{S}\ n \\
\mathrm{tail} & ((f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)) & = & (\mathrm{id} + (f + b_0 + h_0))\ (h'\ n) \\
\mathrm{tail} & ((f + b_0 + h_0)\ \underline{\mathrm{b}_0}) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ N) \\
\mathrm{tail} & ((f + b_0 + h_0)\ (\underline{\mathrm{h}_0}\ n)) & = & (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)
\end{array}$$

$$g : \mathbb{N} \to \mathbb{N}$$
$$g = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ 0\ (\lambda n, ih.\mathrm{S}\ n)$$

$$h' : \mathbb{N} \to (\underline{\mathrm{return}}(\mathrm{Stream}) + (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})))$$
$$\begin{array}{lll}
h'\ 0 & = & \underline{\mathrm{b}_0} \\
h'\ (\mathrm{S}\ n) & = & \underline{\mathrm{h}_0}\ n
\end{array}$$

Unfolding of the Pattern Matchings

- h' can now be defined by primitive recursion.
- $(f + b_0 + h_0)$ can be defined by primitive corecursion.

$$
\begin{aligned}
&f : \mathbb{N} \to \mathrm{Stream} \\
&f\ n \quad = \quad (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)
\end{aligned}
$$

$$
\begin{aligned}
&(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream} \\
&(f + b_0 + h_0) = \\
&\quad \mathrm{coP}_{\mathrm{Stream},(\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ (\lambda x.\mathrm{case}_r(x)\ \mathrm{of} \\
&\qquad\qquad\qquad (\underline{\mathrm{f}}\ n) \longrightarrow g\ n \\
&\qquad\qquad\qquad (\underline{\mathrm{b}_0}) \longrightarrow 0 \\
&\qquad\qquad\qquad (\underline{\mathrm{h}_0}\ n) \longrightarrow \mathrm{S}\ n) \\
&\qquad\qquad (\lambda x.\mathrm{case}_r(x)\ \mathrm{of} \\
&\qquad\qquad\qquad (\underline{\mathrm{f}}\ n) \longrightarrow h'\ n \\
&\qquad\qquad\qquad (\underline{\mathrm{b}_0}) \longrightarrow \underline{\mathrm{f}}\ N \\
&\qquad\qquad\qquad (\underline{\mathrm{h}_0}\ n) \longrightarrow \underline{\mathrm{f}}\ n)
\end{aligned}
$$

$$
\begin{aligned}
&g : \mathbb{N} \to \mathbb{N} \\
&g = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ 0\ (\lambda n, ih.\mathrm{S}\ n)
\end{aligned}
$$

$$
\begin{aligned}
&h' : \mathbb{N} \to (\underline{\mathrm{return}}(\mathbb{N}) + \underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \\
&h' = \mathrm{P}_{\mathbb{N},(\underline{\mathrm{return}}(\mathbb{N})+\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ \underline{\mathrm{b}_0}\ (\lambda n, ih.\underline{\mathrm{h}_0}\ n)
\end{aligned}
$$

Reduction to Primitive (Co)Recursion

- The case distinction can be trivially replaced by the case distinction operator.

$$f : \mathbb{N} \to \mathrm{Stream}$$
$$f\ n \;=\; (f + b_0 + h_0)\ (\underline{\mathrm{f}}\ n)$$

$$(f + b_0 + h_0) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N})) \to \mathrm{Stream}$$
$$\begin{aligned}(f + b_0 + h_0) = \\ \quad \mathrm{coP}_{\mathrm{Stream},(\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ & (\mathrm{case}_{(\underline{\mathrm{f}}(\mathbb{N})+(\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})))} \\ & \quad g \\ & \quad (\mathrm{case}_{\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})}\ (\lambda\_.0)\ \mathrm{S}))) \\ & (\mathrm{case}_{(\underline{\mathrm{f}}(\mathbb{N})+(\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})))} \\ & \quad h' \\ & \quad (\mathrm{case}_{\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N})}\ (\lambda\_.\underline{\mathrm{f}}\ N)\ \underline{\mathrm{f}}))\end{aligned}$$

$$g : \mathbb{N} \to \mathbb{N}$$
$$g = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ 0\ (\lambda n, ih.\mathrm{S}\ n)$$

$$h' : \mathbb{N} \to (\underline{\mathrm{return}}(\mathbb{N}) + \underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{b}_0} + \underline{\mathrm{h}_0}(\mathbb{N}))$$
$$h' = \mathrm{P}_{\mathbb{N},(\underline{\mathrm{return}}(\mathbb{N})+\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{b}_0}+\underline{\mathrm{h}_0}(\mathbb{N}))}\ \underline{\mathrm{b}_0}\ (\lambda n, ih.\underline{\mathrm{h}_0}\ n)$$

Fibonacci Numbers

Efficient Haskell version adapted to our codata notation:

$$\begin{array}{l} \mathrm{codata\ Stream} : \mathrm{Set\ where} \\ \quad \mathrm{cons} : \mathbb{N} \to \mathrm{Stream} \to \mathrm{Stream} \end{array}$$

$$\begin{array}{l} \mathrm{tail} : \mathrm{Stream} \to \mathrm{Stream} \\ \mathrm{tail}\ (\mathrm{cons}\ n\ l) = l \end{array}$$

$$\begin{array}{l} \mathrm{addStream} : \mathrm{Stream} \to \mathrm{Stream} \to \mathrm{Stream} \\ \mathrm{addStream}\ (\mathrm{cons}\ n\ l)\ (\mathrm{cons}\ n'\ l') = \mathrm{cons}\ (n + n')\ (\mathrm{addStream}\ l\ l') \end{array}$$

$$\begin{array}{l} \mathrm{fib} : \mathrm{Stream} \\ \mathrm{fib} = \mathrm{cons}\ 1\ (\mathrm{cons}\ 1\ (\mathrm{addStream\ fib}\ (\mathrm{tail\ fib}))) \end{array}$$

Requires lazy evaluation.

Fibonacci Numbers using Coalgebras

$$
\begin{array}{l}
\mathrm{coalg}\ \mathrm{Stream} : \mathrm{Set}\ \ \mathrm{where} \\
\quad \begin{array}{lcl}
\mathrm{head} & : & \mathrm{Stream} \to \mathbb{N} \\
\mathrm{tail} & : & \mathrm{Stream} \to \mathrm{Stream}
\end{array}
\end{array}
$$

$$
\begin{array}{l}
\mathrm{addStream} : \mathrm{Stream} \to \mathrm{Stream} \to \mathrm{Stream} \\
\begin{array}{lcl}
\mathrm{head}\ (\mathrm{addStream}\ l\ l') & = & \mathrm{head}\ l + \mathrm{head}\ l' \\
\mathrm{tail}\ \ \ (\mathrm{addStream}\ l\ l') & = & \mathrm{addStream}\ (\mathrm{tail}\ l)\ (\mathrm{tail}\ l')
\end{array}
\end{array}
$$

$$
\begin{array}{l}
\mathrm{fib} : \mathrm{Stream} \\
\begin{array}{lcl}
\mathrm{head}\ \mathrm{fib} & = & 1 \\
\mathrm{head}\ (\mathrm{tail}\ \mathrm{fib}) & = & 1 \\
\mathrm{tail}\ \ \ (\mathrm{tail}\ \mathrm{fib}) & = & \mathrm{addStream}\ \mathrm{fib}\ (\mathrm{tail}\ \mathrm{fib})
\end{array}
\end{array}
$$

No laziness required. Requires full corecursion (but terminates).

Multiple Constructors in Algebras and Coalgebras

- Having more than one constructor in algebras correspond to disjoint union:

$$\begin{array}{l} \mathrm{data}\ \mathbb{N} : \mathrm{Set}\ \ \mathrm{where} \\ \quad \begin{array}{lcl} 0 & : & \mathbb{N} \\ \mathrm{S} & : & \mathbb{N} \to \mathbb{N} \end{array} \end{array}$$

corresponds to

$$\begin{array}{l} \mathrm{data}\ \mathbb{N} : \mathrm{Set}\ \ \mathrm{where} \\ \quad \begin{array}{lcl} \mathrm{intro} & : & (1 + \mathbb{N}) \to \mathbb{N} \end{array} \end{array}$$

Multiple Constructors in Algebras and Coalgebras

- Dual of disjoint union is products, and therefore multiple destructors correspond to product:

$$
\begin{array}{l}
\text{coalg Stream} : \text{Set where} \\
\quad \begin{array}{lcl}
\text{head} & : & \text{Stream} \to \mathbb{N} \\
\text{tail} & : & \text{Stream} \to \text{Stream}
\end{array}
\end{array}
$$

  corresponds to

$$
\begin{array}{l}
\text{coalg Stream} : \text{Set where} \\
\quad \begin{array}{lcl}
\text{case} & : & \text{Stream} \to (\mathbb{N} \times \text{Stream})
\end{array}
\end{array}
$$

Codata Types Correspond to Disjoint Union

- Consider

$$
\begin{array}{l}
\text{codata coList} : \text{Set where} \\
\quad \begin{array}{lll} \text{nil} & : & \text{coList} \\ \text{cons} & : & \mathbb{N} \to \text{coList} \to \text{coList} \end{array}
\end{array}
$$

- Cannot be simulated by using several destructors.

Simulating Codata Types by Simultaneous Algebras/Coalgebras

- Represent Codata as follows

$$
\begin{array}{l}
\text{mutual} \\
\quad \mathrm{coalg\ coList : Set\ where} \\
\quad\quad \mathrm{unfold : coList} \to \mathrm{coListShape} \\
\\
\quad \mathrm{data\ coListShape : Set\ where} \\
\quad\quad \begin{array}{lcl} \mathrm{nil} & : & \mathrm{coListShape} \\ \mathrm{cons} & : & \mathbb{N} \to \mathrm{coList} \to \mathrm{coListShape} \end{array}
\end{array}
$$

Definition of Append

$$\begin{array}{l}\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\ \text{append}\ l\ l' = ?\end{array}$$

Definition of Append

$$\begin{array}{l}\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\ \text{append}\ \mathit{l}\ \mathit{l}' = ?\end{array}$$

We copattern match on $\text{append}\ \mathit{l}\ \mathit{l}' : \text{coList}$:

$$\begin{array}{l}\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\ \text{unfold}\ (\text{append}\ \mathit{l}\ \mathit{l}') = ?\end{array}$$

Definition of Append

$$
\begin{array}{l}
\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\
\text{unfold}\ (\text{append}\ l\ l') = ?
\end{array}
$$

We cannot pattern match on l.
But we can do so on $(\text{unfold}\ l)$:

$$
\begin{array}{l}
\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\
\text{unfold}\ (\text{append}\ l\ l') = \\
\quad \text{case}\ (\text{unfold}\ l)\ \text{of} \\
\qquad \begin{array}{lcl}
\text{nil} & \to & ? \\
(\text{cons}\ n\ l) & \to & ?
\end{array}
\end{array}
$$

Definition of Append

$$
\begin{array}{l}
\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\
\text{unfold}\ (\text{append}\ l\ l') = \\
\quad \text{case}\ (\text{unfold}\ l)\ \text{of} \\
\qquad \begin{array}{lcl}
\text{nil} & \to & ? \\
(\text{cons}\ n\ l) & \to & ?
\end{array}
\end{array}
$$

We resolve the goals:

$$
\begin{array}{l}
\text{append} : \text{coList} \to \text{coList} \to \text{coList} \\
\text{unfold}\ (\text{append}\ l\ l') = \\
\quad \text{case}\ (\text{unfold}\ l)\ \text{of} \\
\qquad \begin{array}{lcl}
\text{nil} & \to & \text{unfold}\ l' \\
(\text{cons}\ n\ l) & \to & \text{cons}\ n\ (\text{append}\ l\ l')
\end{array}
\end{array}
$$