Unfolding Nested Patterns and Copatterns

Anton Setzer

(Swansea, UK)

Shonan Meeting on Coinduction
7 October 2013

Codata types and Decidable Equality

Pattern and Copattern Matching

Reduction of Mixed Pattern/Copattern Matching to Operators

Conclusion

Theorem Regarding Undecidabilty of Equality

Theorem

Assume the following:

- *There exists a subset* $\mathrm{Stream} \subseteq \mathbb{N}$,[1]
- *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:

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

[1] Thanks to somebody in the audience (M. Hofmann?) pointed out during the talk that Stream needs not to be decidable.

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

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

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

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.

Related Work (Added after the Talk)

- During the talk a related article by Conor McBride was discussed
  - Let's see how things unfold: Reconciling the infinite with the intensional. Proceedings of CALCO'09, LNCS, 2009, 113 – 126.
- While this paper contains the idea we believe that we state a more precise theorem and provide a more formal proof.
- We were not able to reduce the result directly to the undecidability of the Turing Halting problem as suggested in that paper.

Coalgebras defined by Elimination Rules

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

Copattern matching:

$$
\begin{array}{lcl}
g : A \to \text{Stream} & & \\
\text{head}\ (g\ a) & = & f_0\ a \\
\text{tail}\ \ (g\ a) & = & g\ (f_1\ a) \\
\text{or} & & \\
\text{tail}\ \ (g\ a) & = & f_2\ a
\end{array}
$$

Patterns and Copatterns

- We can define now functions by patterns and copatterns.
- Example define stream:
  $f\ n =$
  $n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$

Patterns and Copatterns

$$f\ n = n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$$

$$\begin{array}{lcl} f & : & \mathbb{N} \to \mathrm{Stream} \\ f & = & ? \end{array}$$

Patterns and Copatterns

$f\ n = n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$

$$\begin{array}{l} f : \mathbb{N} \to \mathrm{Stream} \\ f \quad = \quad ? \end{array}$$

Copattern matching on $f : \mathbb{N} \to \mathrm{Stream}$:

$$\begin{array}{l} f : \mathbb{N} \to \mathrm{Stream} \\ f\ n \quad = \quad ? \end{array}$$

Patterns and Copatterns

$f\ n = n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$

$$\begin{array}{l} f : \mathbb{N} \to \mathrm{Stream} \\ f\ n \quad = \quad ? \end{array}$$

Copattern matching on $f\ n : \mathrm{Stream}$:

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

Patterns and Copatterns

$f\ n = n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$

$$\begin{array}{lcl} f : \mathbb{N} \rightarrow \mathrm{Stream} & & \\ f\ n & = & ? \end{array}$$

Solve first case, copattern match on second case:

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

Patterns and Copatterns

$f\ n = n, n, n{-}1, n{-}1, \ldots 0, 0, N, N, N{-}1, N{-}1, \ldots 0, 0, N, N, N{-}1, N{-}1,$

$$
\begin{array}{lcl}
f : \mathbb{N} \to \mathrm{Stream} & & \\
f\ n & = & ?
\end{array}
$$

Solve second line, pattern match on n

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

Patterns and Copatterns

$f\ n = n, n, n-1, n-1, \ldots 0, 0, N, N, N-1, N-1, \ldots 0, 0, N, N, N-1, N-1,$

$$\begin{array}{lcl} f : \mathbb{N} \to \mathrm{Stream} & & \\ f\ n & = & ? \end{array}$$

Solve remaining cases

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

Operators for Primitive (Co)Recursion

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

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

Operators for full/primitive (co)recursion

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

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

Consider Example from above

$$
\begin{array}{llcl}
\multicolumn{4}{l}{f : \mathbb{N} \rightarrow \mathrm{Stream}} \\
\mathrm{head} & (f\ n) & = & n \\
\mathrm{head} & (\mathrm{tail}\ (f\ n)) & = & n \\
\mathrm{tail} & (\mathrm{tail}\ (f\ 0)) & = & f\ N \\
\mathrm{tail} & (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & = & f\ n
\end{array}
$$

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.

Step 1: Unnesting of Nested (Co)Pattern Matching

We follow the steps in the pattern matching:

We start with

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

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

$$
\begin{array}{lrcl}
\multicolumn{4}{l}{f : \mathbb{N} \to \mathrm{Stream}} \\
\mathrm{head} & (f\ n) & = & n \\
\mathrm{head} & (\mathrm{tail}\ (f\ n)) & = & n \\
\mathrm{tail} & (\mathrm{tail}\ (f\ n)) & = & ?
\end{array}
$$

corresponds to

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

Pattern matching on $\mathrm{tail}\ (\mathrm{tail}\ (f\ n))$:

$$\begin{array}{llcl}
\multicolumn{4}{l}{f : \mathbb{N} \to \mathrm{Stream}} \\
\mathrm{head} & (f\ n) & = & n \\
\mathrm{head} & (\mathrm{tail}\ (f\ n)) & = & n \\
\mathrm{tail} & (\mathrm{tail}\ (f\ 0)) & = & f\ N \\
\mathrm{tail} & (\mathrm{tail}\ (f\ (\mathrm{S}\ n))) & = & f\ n
\end{array}$$

corresponds to

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

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 combine f, g into one function $f + g$.

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

Unfolding of the Pattern Matchings

- The call of k has result always of the form $(f+g)(\ fbf\ \ n))$. So we can replace the recursive call $k\ n$ by $(f+g)(\underline{\mathrm{f}}\ (k'\ n))$.

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

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

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

Unfolding of the Pattern Matchings

- $(f + g)$ can be defined by primitive corecursion.
- k' can be defined by primitive recursion.

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

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

$$k' : \mathbb{N} \to \mathbb{N}$$
$$k' = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ n\ (\lambda n, ih.n)$$

Reduction to Primitive (Co)Recursion

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

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

$$(f + g) : (\underline{\mathrm{f}}(\mathbb{N}) + \underline{\mathrm{g}}(\mathbb{N})) \to \mathrm{Stream}$$
$$\begin{aligned}(f + g) = \\ &\mathrm{coP}_{\mathrm{Stream},\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{g}}(\mathbb{N})}\ (\mathrm{case}_{\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{g}}(\mathbb{N})}\ \mathrm{id}\ \mathrm{id}) \\ &\qquad (\mathrm{case}_{\underline{\mathrm{f}}(\mathbb{N})+\underline{\mathrm{g}}(\mathbb{N})}\ \underline{\mathrm{g}}\ (\underline{\mathrm{f}} \circ k'))\end{aligned}$$

$$k' : \mathbb{N} \to \mathbb{N}$$
$$k' = \mathrm{P}_{\mathbb{N},\mathbb{N}}\ n\ (\lambda n, ih.n)$$

Conclusion

- Codata types make the assumption

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

  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.