Streams Coalgebraically

Jan Rutten

CWI & RUN & VUA

18 January 2010
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Streams are everywhere

- Psychology: Streams of consciousness
- Philosophy: You never step into the same stream twice
- Poetry: Dream streams and mean streams
- Politics: I had a stream ...
- Computer science, mathematics, engineering: Data flow, Taylor series, signal processing, video streaming, etc. etc.
Overview

- 1. The method of coalgebra
- 2. Defining streams by coinduction
- 3. The Hamming numbers
- 4. Rational streams
1. The method of coalgebra

- *basic*: dynamical systems
- systems with *output*: stream systems
- systems with *output* and *input*: automata
- systems with *output* and *input*: Mealy machines
Dynamical systems

- A dynamical system is: set $X$ plus function $t : X \rightarrow X$
- Notation: $x \rightarrow y \equiv t(x) = y$. Example:

  $\xrightarrow{x} y \xrightarrow{} z \leftarrow p \xrightarrow{} a \rightarrow b$

- Homomorphisms: $X \xrightarrow{t_X} Y \xrightarrow{t_Y} Y$ for reducing systems.
Adding output: stream systems

- A system with output in a set $O$:

$$< o, t > : X \rightarrow O \times X$$

- Notation: $x \xrightarrow{a} y \equiv o(x) = a$ and $t(x) = y$.

- Homomorphisms:
A final system with output: streams

- The set of streams: $O^\omega = \{\sigma \mid \sigma : \mathbb{N} \rightarrow O\}$ is *final*:

$$X \xrightarrow{\exists! f} O^\omega$$

where streams transitions are given by:

$$(\sigma(0), \sigma(1), \sigma(2), \ldots) \xrightarrow{\sigma(0)} (\sigma(1), \sigma(2), \sigma(3), \ldots)$$

and where the *final* homomorphism $f : X \rightarrow O^\omega$ is given by

$$f(x) = (o(x), o(t(x)), o(t^2(x)), \ldots)$$
Streams: minimal systems with output

- **Finality** of the system of streams:

\[
X \xrightarrow{\exists! f} O^\omega
\]

\[
\langle o, t \rangle \xrightarrow{} O \times X \xrightarrow{1 \times f} O \times O^\omega
\]

gives *semantics* in the form of *minimization*:

\[
X_0 \xrightarrow{a} X_1 \xrightarrow{b} X_2 \xrightarrow{a} X_3 \leftrightarrow (ab)^\omega \leftrightarrow (ba)^\omega
\]
Defining streams by coinduction

- *Finality* gives also rise to definitions:

\[
X \xrightarrow{\exists! f} O^\omega
\]

\[
<o,t> \downarrow \downarrow
\]

\[
O \times X \xrightarrow{1 \times f} O \times O^\omega
\]

We say that \( \sigma \in O^\omega \) is defined by *coinduction* if

\[
\exists (X, <o,t>), \exists x \in X : \sigma = f(x)
\]

- Too general, \((X, <o,t>)\) should be finite or *finitary*. To be continued in a minute.
Combining input and output: automata

- Language (of a state, of a regular expression):

\[ X \rightarrow \exists! l \rightarrow 2^A^* \]

\[ 2 \times X^A \rightarrow 2 \times (2^A^*)^A \]

\[ RExp \rightarrow \exists! l \rightarrow 2^A^* \]

\[ 2 \times RExp^A \rightarrow 2 \times (2^A^*)^A \]

- Semantics \( < l(x) > \subseteq 2^A^* \) is minimization of \( < x > \subseteq X \).

- Language equivalence = coalgebraic bisimilarity

- Proofs of language equivalence: by coinduction.
Combining input and output: Mealy machines

- The set $\Gamma$ of all causal functions

\[ g : A^\omega \to B^\omega \]

is a final Mealy machine:

\[
\begin{array}{c}
\exists! h \\
B \times X \xrightarrow{A} (B \times \Gamma)^A
\end{array}
\]

- Cf. second lecture.
Systems coalgebraically: summary

- Unified perspective on many types of systems:
  
  \[
  \begin{array}{c}
  X \\
  \downarrow \\
  X \\
  \downarrow \\
  X \\
  \downarrow \\
  X \\
  \downarrow \\
  X \\
  \downarrow \\
  X \\
  \downarrow \\
  O \times X \\
  \downarrow \\
  2 \times X^A \\
  \downarrow \\
  (B \times X)^A \\
  \end{array}
  \]

- Homomorphisms for comparing and reducing systems.

- Final systems give semantics by minimization and definitions by coinduction.

- Also proofs by coinduction = bisimulation proof method.
2. Defining streams by coinduction

- We say that $\sigma \in O^\omega$ is defined by \textit{coinduction} if

\[
\begin{array}{c}
X \xrightarrow{f} O^\omega \\
\langle o, t \rangle \downarrow \\
O \times X \xrightarrow{1 \times f} O \times O^\omega \\
\end{array}
\]

\[\exists (X, \langle o, t \rangle), \exists x \in X : \sigma = f(x)\]

- If $X$ is finite then $\sigma = f(x)$ is \textit{eventually periodic}.
Next we present two examples where \( X \) is infinite but *finitary*:

\[
X \xrightarrow{f} O^\omega \\
\begin{array}{c}
\langle o, t \rangle \\
\downarrow \\
O \times X \\
\downarrow \\
\downarrow \\
O \times O^\omega
\end{array}
\]

- Hamming numbers (here \( X \) will be an inductively defined term algebra)
- Rational streams (here \( X \) will be a finite dimensional vector space)
The Hamming numbers

All natural numbers, in increasing order, that have no other prime factors than 2 and 3:

\[ \gamma = (2^0 3^0, 2^1 3^0, 2^0 3^1, 2^2 3^0, 2^1 3^1, 2^3 3^0, 2^0 3^2, 2^2 3^1, \ldots) \]
\[ = (1, 2, 3, 4, 6, 8, 9, 12, \ldots) \]

We define \( \gamma \) by a stream differential equation

\[ \gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1 \]

(where \( \gamma' = \text{tail}(\gamma) = (\gamma(1), \gamma(2), \gamma(3), \ldots) \)).

Our goal is to prove that this stream differential equation has a (unique) solution.
The Hamming numbers

All natural numbers, in increasing order, that have no other prime factors than 2 and 3:

\[ \gamma = (2^03^0, 2^13^0, 2^03^1, 2^23^0, 2^13^1, 2^33^0, 2^03^2, 2^23^1, \ldots) \]
\[ = (1, 2, 3, 4, 6, 8, 9, 12, \ldots) \]

We define \( \gamma \) by a *stream differential equation*

\[ \gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1 \]

(where \( \gamma' = \text{tail}(\gamma) = (\gamma(1), \gamma(2), \gamma(3), \ldots) \)).

Our goal is to prove that this stream differential equation has a (unique) solution.
The Hamming numbers

All natural numbers, in increasing order, that have no other prime factors than 2 and 3:

\[ \gamma = (2^0 3^0, 2^1 3^0, 2^0 3^1, 2^2 3^0, 2^1 3^1, 2^3 3^0, 2^0 3^2, 2^2 3^1, \ldots) \]
\[ = (1, 2, 3, 4, 6, 8, 9, 12, \ldots) \]

We define \( \gamma \) by a stream differential equation

\[ \gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1 \]

(where \( \gamma' = tail(\gamma) = (\gamma(1), \gamma(2), \gamma(3), \ldots) \)).

Our goal is to prove that this stream differential equation has a (unique) solution.
The stream differential equation

$$\gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1$$

where the ordered merge $\parallel : \mathbb{N}^\omega \times \mathbb{N}^\omega \rightarrow \mathbb{N}^\omega$ is given by

$$(\sigma \parallel \tau)' = \begin{cases} 
\sigma' \parallel \tau & \text{if } \sigma(0) < \tau(0) \\
\sigma' \parallel \tau' & \text{if } \sigma(0) = \tau(0) \\
\sigma \parallel \tau' & \text{if } \sigma(0) > \tau(0)
\end{cases}$$

$$(\sigma \parallel \tau)(0) = \begin{cases} 
\sigma(0) & \text{if } \sigma(0) < \tau(0) \\
\tau(0) & \text{if } \sigma(0) \geq \tau(0)
\end{cases}$$

and where $2 \times \sigma$ (and similarly $3 \times \sigma$) satisfies

$$(2 \times \sigma)' = 2 \times (\sigma') \quad (2 \times \sigma)(0) = 2 \cdot \sigma(0)$$
Solving the differential equation

\[ \gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1 \]

Assuming the solution exists, we compute the first few derivatives of \( \gamma \):

\[
\begin{align*}
\gamma^{(1)} &= (2 \times \gamma) \parallel (3 \times \gamma) \\
\gamma^{(2)} &= (2 \times ((2 \times \gamma) \parallel (3 \times \gamma))) \parallel (3 \times \gamma) \\
\gamma^{(3)} &= (2 \times ((2 \times \gamma) \parallel (3 \times \gamma))) \parallel (3 \times ((2 \times \gamma) \parallel (3 \times \gamma)))
\end{align*}
\]

The idea: define \textit{syntactic} terms for all possible such expressions:

\[
\text{Term} \ni t ::= c \mid \sigma (\sigma \in \mathbb{N}^\omega) \mid 2\text{times}(t) \mid 3\text{times}(t) \mid \text{merge}(t_1, t_2)
\]
The syntactic stream system of terms

\[ \text{Term} \ni t ::= c \mid \sigma (\sigma \in \mathbb{N}^\omega) \mid 2\text{times}(t) \mid 3\text{times}(t) \mid \text{merge}(t_1, t_2) \]

Next we turn the set \text{Term} into a stream system

\[ \text{Term} \xrightarrow{<o,n>} \mathbb{N} \times \text{Term} \]

by defining functions \( o : \text{Term} \rightarrow \mathbb{N} \) and \( n : \text{Term} \rightarrow \text{Term} \) by \textit{induction} on the structure of terms, following the stream diff. eqn’s. For instance,

\[
n(\text{merge}(t_1, t_2)) = \begin{cases} 
\text{merge}(n(t_1), t_2) & \text{if } o(t_1) < o(t_2) \\
\text{merge}(n(t_1), n(t_2)) & \text{if } o(t_1) = o(t_2) \\
\text{merge}(t_1, n(t_2)) & \text{if } o(t_1) > o(t_2)
\end{cases}
\]
The solution

• By finality, \[ \text{Term} \xrightarrow{\exists! f} \mathbb{N}^\omega \]
  \[
  \begin{array}{c}
  \langle o, n \rangle \\
  \downarrow \\
  \mathbb{N} \times \text{Term} \\
  \downarrow \\
  \mathbb{N} \times \mathbb{N}^\omega
  \end{array}
  \xrightarrow{1 \times f}
  \]

• Using \( f \), we define

\[
\gamma = f(c) \\
\sigma \parallel \tau = f(\text{merge}(\sigma, \tau))
\]

(and similarly for \( 2 \times \sigma \) and \( 3 \times \sigma \)).

• Finally one shows that, indeed,

\[
\gamma' = (2 \times \gamma) \parallel (3 \times \gamma) \quad \gamma(0) = 1
\]
Discussion

- An elementary and fairly general *syntactic* solution method.
- Many other examples. E.g., Thue-Morse sequence:
  \[
  \sigma'' = \text{zip}(\sigma', \text{inv}(\sigma')) \quad \sigma(0) = 0, \quad \sigma(1) = 1
  \]
  Also similar schemes for regular expressions, formal power series etc.
- *Format* of the right handside of the diff. eqn’s is important.
- Interplay between algebra (terms, induction) and coalgebra (coinduction).
- Cf. $\lambda$-coiteration.
Rational streams

- Ingredients: vector space (here $\mathbb{R}^2$), and two linear maps:

  \[ O : \mathbb{R}^2 \to \mathbb{R}, \quad T : \mathbb{R}^2 \to \mathbb{R}^2 \]

- Finality as before, but everything now lives in $\textbf{Vect}$ (category of vector spaces and linear maps):

\[
\begin{array}{ccc}
\mathbb{R}^2 & \xrightarrow{\exists!F} & \mathbb{R}^\omega \\
\langle O, T \rangle & \downarrow & \\
\mathbb{R} \times \mathbb{R}^2 & \longrightarrow & \mathbb{R} \times \mathbb{R}^\omega
\end{array}
\]

- Note that $\mathbb{R}^2$ is infinite but $O$ and $T$ are finite.
Rational streams

- Ingredients: vector space (here $\mathbb{R}^2$), and two linear maps:

$$O : \mathbb{R}^2 \to \mathbb{R}, \quad T : \mathbb{R}^2 \to \mathbb{R}^2$$

- Finality as before, but everything now lives in $\text{Vect}$ (category of vector spaces and linear maps):

$$\begin{array}{c}
\mathbb{R}^2 \xrightarrow{\exists ! F} \mathbb{R}^\omega \\
\downarrow \downarrow \downarrow \downarrow \downarrow \\
\mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R} \times \mathbb{R}^\omega
\end{array}$$

- Note that $\mathbb{R}^2$ is infinite but $O$ and $T$ are finite.
An example

- Let $O = \begin{pmatrix} 1 & 1 \end{pmatrix} : \mathbb{R}^2 \to \mathbb{R}$ and let

$$T = \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix} : \mathbb{R}^2 \to \mathbb{R}^2$$

- We can now compute $F$ from $O$ and $T$:

$$\begin{array}{c}
\mathbb{R}^2 \rightarrow \exists! F \rightarrow \mathbb{R}^\omega \\
\downarrow \quad \downarrow \quad \downarrow \\
\mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R} \times \mathbb{R}^\omega
\end{array}$$
An example

- We can now compute $F$ from $O$ and $T$:

\[
\begin{align*}
\mathbb{R}^2 \rightarrow \exists ! F \rightarrow \mathbb{R}^\omega \\
\langle O, T \rangle \downarrow \downarrow \\
\mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R} \times \mathbb{R}^\omega
\end{align*}
\]

using *stream calculus*: for all $(r_1, r_2) \in \mathbb{R}^2$,

\[
F(r_1, r_2) = O \circ (1 - (X \times T))^{-1} (r_1, r_2)
\]

\[
= \begin{pmatrix} 1 & 1 \end{pmatrix} \circ
\begin{pmatrix}
\frac{1-2X}{(1-X)^2} & \frac{X}{(1-X)^2} \\
\frac{-X}{(1-X)^2} & \frac{1}{(1-X)^2}
\end{pmatrix}
\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}
\]

\[
= \frac{r_1 + r_2 + X \times (-3r_1 + r_2)}{(1 - X)^2}
\]
An example: conclusions

- We have computed $F$ from $O$ and $T$:

$$\mathbb{R}^2 \vdash \exists ! F \rightarrow \mathbb{R}^\omega$$

$$\downarrow$$

$$\mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R} \times \mathbb{R}^\omega$$

$$F(r_1, r_2) = \frac{(r_1 + r_2) + X \times (-3r_1 + r_2)}{(1 - X)^2}$$

- More generally: a stream is *rational* if it can be obtained by coinduction from a finite dimensional vector space.

- Cf. Regular languages, formal power series.

- Cf. weighted automata.