# A final coalgebra for the $k$-regular and $k$-automatic sequences

Joost Winter, Helle Hvid Hansen, Clemens Kupke, Jan Rutten

Centrum Wiskunde & Informatica
Radboud Universiteit Nijmegen
University of Strathclyde

September 21, 2013

# Introduction

- *k*-automatic and *k*-regular sequences: classes defined by Allouche/Shallit
- A sequence $\sigma \in \mathbb{Z}^\omega$ is *k*-automatic if generated by a deterministic automaton with output in $\{0, \ldots, k-1\}$...
- ...where $\sigma(n)$ is output after reading *n* in base *k*.

# Introduction

- *k*-automatic and *k*-regular sequences: classes defined by Allouche/Shallit
- A sequence $\sigma \in \mathbb{Z}^\omega$ is *k*-automatic if generated by a deterministic automaton with output in $\{0, \ldots, k-1\}\ldots$
- ... where $\sigma(n)$ is output after reading *n* in base *k*.
- *k*-regular sequences generalize this:

$$\frac{\text{k-regular}}{\text{k-automatic}} = \frac{\text{weighted automata}}{\text{deterministic automata}}$$

- This talk: connecting *k*-regular sequences to (abstract) coalgebra and (concrete) behavioural differential equations.

# $k$-regular sequences: a definition (for $k = 2$)

We call a sequence (or stream) $\sigma$ 2-regular when there is a finite family of sequences

$$\Sigma = (\sigma_i) \quad i \leq n \in \mathbb{N}$$

with $\sigma_0 = \sigma$, s.t. for all $i \leq n$ the sequences $\mathbf{even}(\sigma_i)$ and $\mathbf{odd}(\sigma_i)$ are linear combinations of sequences from $\Sigma$.
Here **even** and **odd** are defined by

$$\mathbf{even}(\tau)(n) = \tau(2n)$$

and

$$\mathbf{odd}(\tau)(n) = \tau(2n + 1)$$

# Derivative and **zip**

We will reason with the stream derivative from the coinductive stream calculus. Definition:

$$\sigma'(n) = \sigma(n+1)$$

We can define streams and operators coinductively by giving the first element and the derivative, e.g.

$$\begin{aligned} \mathbf{zip}(\sigma,\tau)(0) &= \sigma(0) \\ \mathbf{zip}(\sigma,\tau)' &= \mathbf{zip}(\tau,\sigma') \end{aligned}$$

gives

$$\begin{aligned} \mathbf{zip}(\sigma,\tau)(2k) &= \sigma(k) \\ \mathbf{zip}(\sigma,\tau)(2k+1) &= \tau(k) \end{aligned}$$

and thus

$$\mathbf{zip}(\mathbf{even}(\sigma),\mathbf{odd}(\sigma)) = \sigma$$

# Systems of **zip**-equations

$k$-regular sequences can be seen as solutions to finite systems of equations.

$$\begin{array}{rcl} \tau_1 & = & \mathbf{zip}(\tau_1^e, \tau_1^o) \\ \vdots & & \vdots \\ \tau_n & = & \mathbf{zip}(\tau_n^e, \tau_n^o) \end{array}$$

**Example:** the sequence of numbers whose base 3 representation does not contain the digit '2'

$$0, 1, 3, 4, 9, 10, 12, 13, 27, 28, 30, 31, \ldots$$

is a solution to

$$\begin{array}{rcl} \sigma & = & \mathbf{zip}(3\sigma, 3\sigma + \mathbf{ones}) \\ \mathbf{ones} & = & \mathbf{zip}(\mathbf{ones}, \mathbf{ones}) \end{array}$$

(with $\mathbf{ones}(0) = 1$, $\sigma(0) = 0$)

# Automata as coalgebras

- Automaton (with output in $S$, input in $A$) is coalgebra for the functor $S \times -^A$.
- Semantics $[\![-]\!]$ given by unique morphism into final automaton:

$$\begin{tikzcd} X \arrow[r, dashed, "\exists! [\![-]\!]"'] \arrow[d, "(o,\delta)"'] & S^{A^*} \arrow[d, "(O,\Delta)"] \\ S \times X^A \arrow[r, dashed, "1_S \times [\![-]\!]"'] & S \times (S^{A^*})^A \end{tikzcd}$$

Fact: $[\![x]\!](w) = o(x_w)$

# Streams are an instance of this

If $|A| = 1$, note that $S^{A^*} \cong S^{\mathbb{N}}$ and we get

$$\begin{tikzcd}
X \arrow[r, dashed, "\exists! [\![-]\!]"'] \arrow[d, "(o,\delta)"'] & S^{\mathbb{N}} \arrow[d, "(O,\Delta)"] \\
S \times X \arrow[r, dashed] & S \times (S^{\mathbb{N}})
\end{tikzcd}$$

$$\begin{aligned}
O(\sigma) &= \sigma(0) \\
\Delta(\sigma) &= \sigma'
\end{aligned}$$

# Main result (for case $k = 2$)

**Theorem**

*A sequence $\sigma$ is 2-regular if and only if it is the unique solution to a system of stream differential equations*

$$o(x) = k \qquad x' = \mathbf{zip}(x_e, x_o)$$

*for a finite set $X$, where $k \in \mathbb{Z}$, and for each $x \in X$, $x_e$ and $x_o$ are given as a linear combination of elements from $X$.*

(also found by Endrullis/Moss/Silva)

# Main result (for case $k = 2$)

**Theorem**
*A sequence $\sigma$ is 2-regular if and only if it is the unique solution to a system of stream differential equations*

$$o(x) = k \qquad x' = \mathbf{zip}(x_e, x_o)$$

*for a finite set $X$, where $k \in \mathbb{Z}$, and for each $x \in X$, $x_e$ and $x_o$ are given as a linear combination of elements from $X$.*

(also found by Endrullis/Moss/Silva)

Idea: transform flat systems into guarded systems.

or: move from standard base $k$ numeration to bijective base $k$ numeration

Construct a system of stream differential equations from the earlier system:

$$
\begin{aligned}
\sigma' &= \mathbf{zip}(3\sigma + \mathbf{ones}, 3\sigma') \\
\sigma'' &= \mathbf{zip}(3\sigma', 3\sigma' + \mathbf{ones}') \\
\mathbf{ones}' &= \mathbf{zip}(\mathbf{ones}', \mathbf{ones}) \\
\mathbf{ones}'' &= \mathbf{zip}(\mathbf{ones}', \mathbf{ones}')
\end{aligned}
$$

or

$$
\begin{aligned}
w' &= \mathbf{zip}(3w + y, 3x) \\
x' &= \mathbf{zip}(3x, 3x + z) \\
y' &= \mathbf{zip}(y, z) \\
z' &= \mathbf{zip}(z, z)
\end{aligned}
$$

Add output values to specification and you're done!

# A final coalgebra diagram

Semantics can be given by the following diagram (initiality + finality):

$$\begin{tikzcd}
X \arrow[r, hook, "\eta"] \arrow[d, "{(o,d)}"] & S^X \arrow[r, dashed, "{[\![-]\!]}"] \arrow[dl, "{(\hat{o},\hat{d})}"] & S^{\mathbb{N}} \arrow[d, "{(\mathbf{head},\delta)}"] \\
S \times (S^X)^{A_2} \arrow[rr, dashed] & & S \times (S^{\mathbb{N}})^{A_2}
\end{tikzcd}$$

with

$$\begin{aligned}
\delta(\sigma)(1) &= \mathbf{even}(\sigma') \\
\delta(\sigma)(2) &= \mathbf{odd}(\sigma')
\end{aligned}$$

# An isomorphism of final coalgebras

$$\begin{tikzcd}
S^{\mathbb{N}} \arrow[r, equal, "\cong"] \arrow[d, "(\mathbf{head}, \delta)"'] & S\langle\langle A_2 \rangle\rangle \arrow[d, "(O, \Delta)"] \\
S \times (S^{\mathbb{N}})^{A_2} \arrow[r, equal] & S \times S\langle\langle A_2 \rangle\rangle^{A}
\end{tikzcd}$$

Can be proven using the bijective base $k$ numeration between $\mathbb{N}$ and $(A_k)^*$.

Gives correspondence with weighted automata (over any semiring $S$).

# Application: divide and conquer recurrences

On the Online Encyclopedia of Integer Sequences, some formats for divide and conquer recurrences are given. E.g.

$$\begin{aligned} a(2n) &= Ca(n) + Ca(n-1) + P(n) \\ a(2n+1) &= 2Ca(n) + Q(n) \end{aligned}$$

where $P$ and $Q$ are expressible by a rational g.f.

# Application: divide and conquer recurrences

On the Online Encyclopedia of Integer Sequences, some formats for divide and conquer recurrences are given. E.g.

$$\begin{aligned} a(2n) &= Ca(n) + Ca(n-1) + P(n) \\ a(2n+1) &= 2Ca(n) + Q(n) \end{aligned}$$

where $P$ and $Q$ are expressible by a rational g.f.

Q (asked on `oeis.org/somedcgf.html`): 'An open question would be whether all sequences here discussed are 2-regular.'

A: if you replace the condition 'expressible by a rational g.f.' by '2-regular' yes (includes all their examples), otherwise no.

# Generalizations, conclusions and future work

- Everything told here about 2 works for any $k \geq 2$.
- We established a correspondence between rational power series in $k$ (noncomm.) variables and $k$-regular sequences over arbitrary semirings.
- ... allowing us to translate back and forth between recurrences and systems of stream differential equations.

# Generalizations, conclusions and future work

- Everything told here about 2 works for any $k \geq 2$.
- We established a correspondence between rational power series in $k$ (noncomm.) variables and $k$-regular sequences over arbitrary semirings.
- ... allowing us to translate back and forth between recurrences and systems of stream differential equations.
- Future work: how about $k$-algebraic sequences?
- ... further investigate the connections with recurrences.