Automatic sequences as context-free systems

Joost Winter (jww Marcello Bonsangue and Jan Rutten)

Centrum Wiskunde & Informatica, Leiden University

October 9, 2012

Overview

This talk will be about:

1. Automata as coalgebras (or systems of behavioural differential equations), with an emphasis on context-free systems.
2. Automatic sequences, which are given by finite automata, but can also be seen as context-free streams.

In parcticular, a construction of context-free systems out of 2-automata will be given.

Formal power series...

Given a finite set A called the *alphabet* and a semiring S, a *formal power series* (in noncommuting variables) on A with coefficients in S is a function

$$A^* \to S.$$

Given some S and A, we let $S\langle\langle A \rangle\rangle$ denote the set of all such power series.

...formal languages, and streams

- When S is the Boolean semiring $\mathbb{B}$ (with $1 + 1 = 1$), formal power series on A with coefficients in S correspond to formal languages over the alphabet A.
- When A is a singleton set, formal power series on A with coefficients on S correspond to *streams* over S.

Polynomials

- A *polynomial* over a set X and with coefficients in a semiring S is a formal power series $s \in S\langle\langle X \rangle\rangle$, such that for only finitely many $w \in X^*$, $s(w) \neq 0$.
- We let $S\langle X \rangle$ denote the polynomials from $S\langle\langle X \rangle\rangle$.
- It is easy to see that this definition corresponds to finite linear combinations of words over X.

Automata as coalgebras

An *automaton* over an alphabet A and with output in a set S consists of

- a set Q of states, together with
- a mapping $f = (o, \delta) : Q \to S \times Q^A$.

We write q_a for $\delta(q)(a)$, and call this the *a-derivative* of q.
When (in the case of streams) the alphabet is a singleton set $\{\mathcal{X}\}$, we usually write x' instead of $x_{\mathcal{X}}$.

Word derivatives

We can extend the notion of derivatives from alphabet symbols to words inductively:

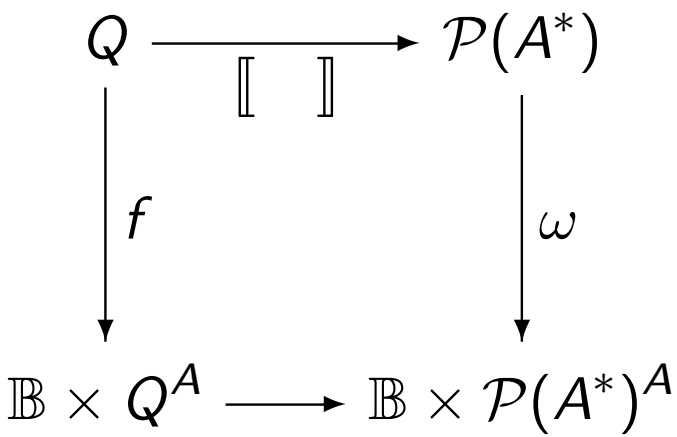
$$\begin{aligned} q_1 &= q \\ q_{a \cdot w} &= (q_a)_w. \end{aligned}$$

For streams: $q^{(n)}$ instead of $q_{\mathcal{X}^n}$.

Stage 1a: a final coalgebra diagram for languages

Every automaton on the semiring $\mathbb{B}$ yields a unique morphism to the *final coalgebra* of all languages, characterized by the equations

$$o(L) = \textbf{if } 1 \in L \textbf{ then } 1 \textbf{ else } 0 \qquad L_a = \{w \mid aw \in L\}.$$



Stage 1b: a final coalgebra diagram for power series

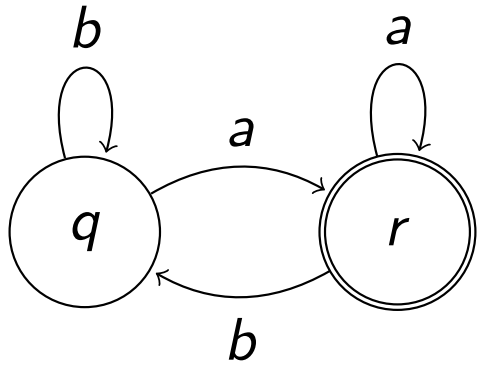
In the case of an arbitrary semiring S, the final coalgebra consists the set $S\langle\langle A\rangle\rangle$. Given a series s, we now have:

$$o(s) = s(1) \qquad s_a(w) = s(aw)$$

$$\begin{array}{ccc} Q & \xrightarrow{\quad [\![\ ]\!] \quad} & S\langle\langle A\rangle\rangle \\ \Big\downarrow{\scriptstyle f} & & \Big\downarrow{\scriptstyle \omega} \\ S \times Q^A & \longrightarrow & S \times S\langle\langle A\rangle\rangle^A \end{array}$$

Stage 1: behavioural differential equations

The automaton



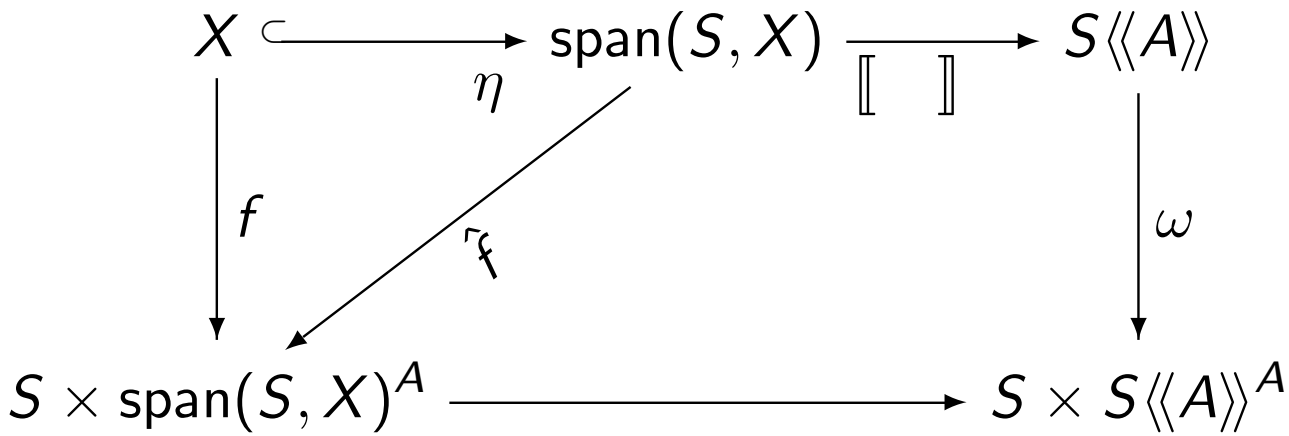
corresponds to the system of *behavioural differental equations*

$$o(q) = 0 \quad q_a = r \quad q_b = q \quad o(r) = 1 \quad r_a = r \quad r_b = q.$$

Right hand side of each derivative: just a state.

Stage 2: rational, nondeterministic and weighted systems

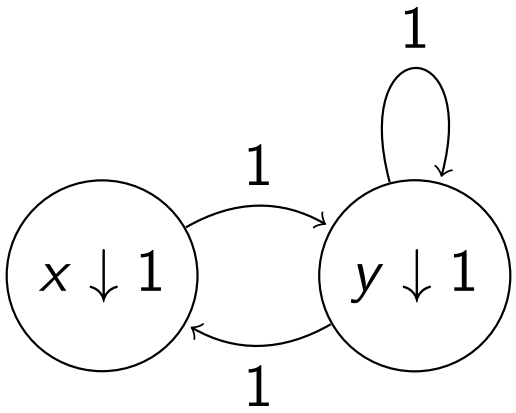
In the case of weighted systems, an extra step of *determinization* takes place:



Here $\hat{f}$ is the unique linear mapping making the left triangle of the diagram commute.

A rational stream

With $\mathbb{N}$ as underlying semiring, the weighted automaton

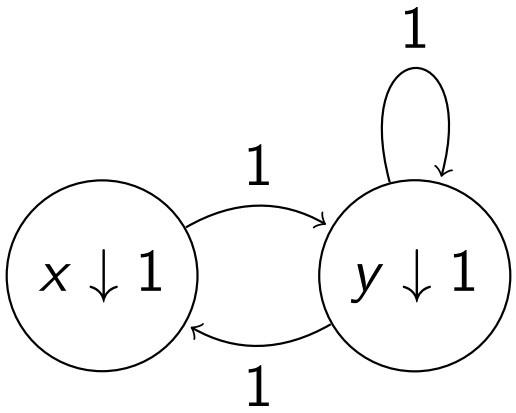


corresponds to the *linear* system of equations

$$o(x) = 1 \qquad x' = y$$
$$o(y) = 1 \quad y' = y + x$$

A rational stream

With $\mathbb{N}$ as underlying semiring, the weighted automaton



corresponds to the *linear* system of equations

$$o(x) = 1 \qquad x' = y$$
$$o(y) = 1 \quad y' = y + x$$

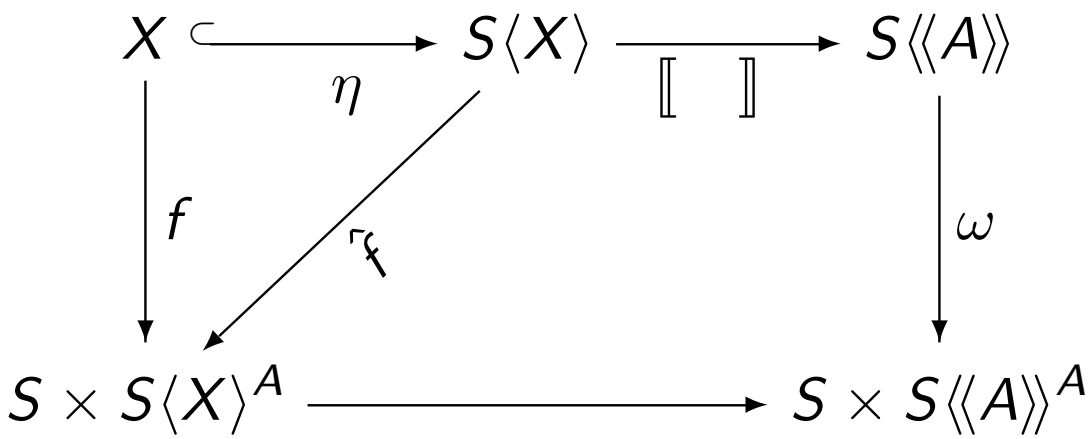
where $[\![x]\!]$ is the Fibonacci sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 \ldots$$

Right hand side of each derivative: linear combination of states.

Stage 3: context-free or polynomial systems

The corresponding construction for context-free languages and the corresponding general class of power series is:



The product rule

Here $\hat{f}$ is the unique extension of f that is linear, and in addition satisfies the equations

$$(st)_a = s_a t + o(s)t_a, \qquad \text{(product rule)}$$

$$o(st) = o(s)o(t), \qquad 1_a = 0, \quad \text{and} \quad o(1) = 1$$

for all $a \in A$ and $s, t \in S\langle X \rangle$.

...distributive laws, SOS rules, and systems of equations

The equations from the previous slide can equivalently be seen as:

- A schema of behavioural differential equations.
- A distributive law, specifying how the coalgebraic operations o and δ distribute over the algebraic operations $+$ and $\cdot$.
- SOS rules of the type

$$\frac{s \xrightarrow{a} u \qquad t \xrightarrow{a} v \qquad s \downarrow o}{st \xrightarrow{a} ut + ov}$$

Correspondences

When the underlying semiring is $\mathbb{B}$, such 'context-free systems' are in direct correspondence with:

- CFGs in Greibach normal form.
- Pushdown automata with a single state.

A context-free stream

Again taking $\mathbb{N}$ as underlying semiring, the Catalan numbers

$$1, 1, 2, 5, 14, 42, 132, 429, 1430, \ldots$$

are context-free, and are generated by the following *polynomial* system of equations:

$$o(x) = 1 \quad x' = x \cdot x$$

Right hand side of each derivative: a polynomial.

Automatic sequences (1)

A *q-automaton* is a finite automaton over the alphabet of digits

$$A_q = \{\bar{0}, \ldots \overline{q-1}\}.$$

Words $w \in A_q^*$ represent numbers in base q (with the least significant bit first), and we let the function

$$[\,] : A_q^* \to \mathbb{N}$$

assign to each word of digits the natural number it represents.

Automatic sequences (2)

We call a stream σ over the field $\mathbb{F}_q$ q-automatic, whenever there is a q-automaton (Q, o, δ) and a state $q \in Q$, such that for each word $w \in A_q^*$,

$$o(q_w) = \sigma([w]).$$

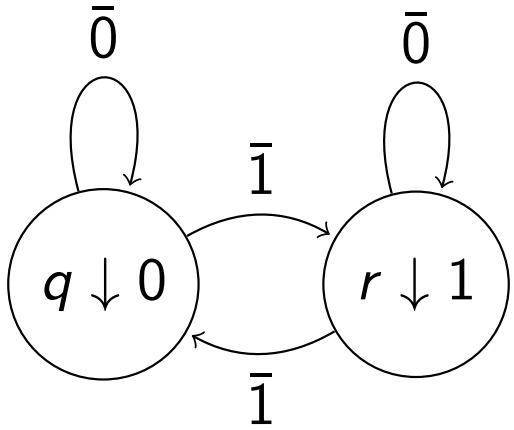
A state q for which this holds for some stream σ is called *consistent*, and we write

$$\mathrm{str}(q) = \sigma.$$

if this is the case. We call an automaton consistent whenever all of its states are.

Example: the Prouhet-Thue-Morse sequence (1)

In the 2-automaton



both states are consistent. Furthermore, $\mathrm{str}(q)$ is equal to the Prouhet-Thue-Morse sequence

$$0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, \ldots$$

Stream functions: even, odd, zip

$$\begin{aligned} \mathsf{even}(\sigma)(n) &= \sigma(2n) \\ \mathsf{odd}(\sigma)(n) &= \sigma(2n+1) \end{aligned}$$

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

Facts about even, odd, and zip

1. $$\sigma = \mathsf{zip}(\mathsf{even}(\sigma), \mathsf{odd}(\sigma)),$$

2. $$o(\mathsf{zip}(\sigma, \tau)) = o(\sigma) \qquad \mathsf{zip}(\sigma, \tau)' = \mathsf{zip}(\tau, \sigma'), \qquad \text{and}$$

3. when the underlying semiring is the finite field $\mathbb{F}_2$:

$$\mathsf{zip}(\sigma, \tau) = \sigma^2 + \mathcal{X}\tau^2.$$

Correspondence between derivatives and even/odd

Lemma

Given a 2-automaton (Q, o, δ) and a consistent $q \in Q$, we have

$$\mathrm{str}(q_{\bar{0}}) = \mathrm{even}(\mathrm{str}(q)) \qquad \text{and} \qquad \mathrm{str}(q_{\bar{1}}) = \mathrm{odd}(\mathrm{str}(q)).$$

Correspondence between derivatives and even/odd

Lemma

Given a 2-automaton (Q, o, δ) and a consistent $q \in Q$, we have

$$\mathrm{str}(q_{\bar{0}}) = \mathrm{even}(\mathrm{str}(q)) \qquad \text{and} \qquad \mathrm{str}(q_{\bar{1}}) = \mathrm{odd}(\mathrm{str}(q)).$$

Proof.

The first equality holds because

$$\begin{aligned} \mathrm{str}(q_{\bar{0}})([w]) &= o((q_{\bar{0}})_w) \\ &= o(q_{\bar{0}\cdot w}) \\ &= \mathrm{str}(q)([\bar{0} \cdot w]) \\ &= \mathrm{str}(q)(2 \cdot [w]) \\ &= \mathrm{even}(\mathrm{str}(q))([w]). \end{aligned}$$

Systems of equations from 2-automatic sequences

Given a consistent 2-automaton (Q, o, δ), consider the following system of equations, with the following set of variables

$$X = \{(x, i) \mid x \in Q, i \in \{0, 1\}\}$$

and set, for all $x \in Q$:

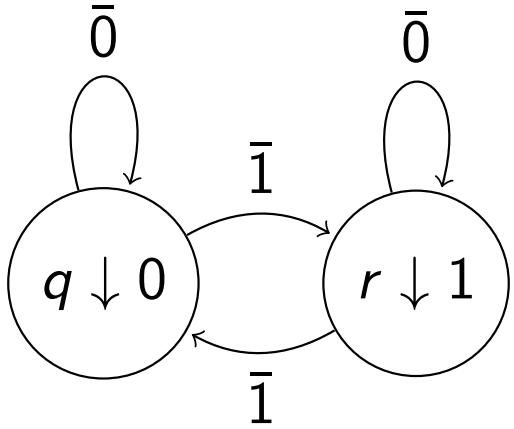
$$\begin{array}{cc} o(x,0) = o(x) & (x,0)' = (x,1) \\ o(x,1) = o(x_{\overline{1}}) & (x,1)' = (x_{\overline{0}},1)^2 + \mathcal{X}(x_{\overline{1}},1)^2 \end{array}$$

Theorem

For all $x \in Q$, $[\![(x,0)]\!] = \mathrm{str}(x)$.

Example: the Prouhet-Thue-Morse sequence (2)

Applying this construction to the earlier example,



we get:

$$
\begin{array}{ll}
o(q,0) = 0 & (q,0)' = (q,1) \\
o(q,1) = 1 & (q,1)' = (q,1)^2 + \mathcal{X}(r,1)^2 \\
o(r,0) = 1 & (r,0)' = (r,1) \\
o(r,1) = 0 & (r,1)' = (r,1)^2 + \mathcal{X}(q,1)^2
\end{array}
$$

Generalizations

This construction can easily be generalized from 2-automata to q-automata. Here q must be of the form p^n for some prime p and natural number n.

- Instead of $\mathrm{zip}(\sigma, \tau)$, we now have $\mathrm{zip}_q(\sigma_0, \ldots, \sigma_{q-1})$.
- Instead of even and odd, we now have $\mathrm{unzip}_{i,q}(\sigma)$ for all $i \in \mathbb{N}$ with $i < q$.
- Instead of $\mathrm{zip}(\sigma, \tau) = \sigma^2 + \mathcal{X}\tau^2$, we now have

$$\mathrm{zip}_q(\sigma_0, \ldots, \sigma_q) = \sum_{i<q} (\mathcal{X}^i \sigma_i^q).$$

Related work

- Work on automatic sequences as regular coalgebras with even/odd-specifications by Kupke/Rutten, including a nice 'relatively final' coalgebra theorem.
- Grabmayer/Endrullis/Hendriks/Klop/Moss also obtained similar results in a paper about automatic sequences and zip-specifiations.
- In 1971, Michel Fliess showed that diagonals of rational series in two commuting variables are *constructively algebraic*. This result is closely related to our result, but via another route.