Links between normality and automata

Olivier Carton

IRIF
Université Paris Diderot & CNRS

Based on join works with V. Becher, P. Heiber and E. Orduna
(Universidad de Buenos Aires & CONICET)

Singapore – IMS

Outline

Outline

Normal words

A normal word is an infinite word such that all finite words of the same length occur in it with the same frequency.

If $x \in A^\omega$ and $w \in A^*$, the frequency of w in x is defined by

$$\mathrm{freq}(x, w) = \lim_{N \to \infty} \frac{|x[1..N]|_w}{N}.$$

where $|z|_w$ denotes the number of occurrences of w in z.

A word $x \in A^\omega$ is normal if for each $w \in A^*$:

$$\mathrm{freq}(x, w) = \frac{1}{|A|^{|w|}}$$

where
- $|A|$ is the cardinality of the alphabet A
- $|w|$ is the length of w.

Normal words (continued)

Theorem (Borel, 1909)

The decimal expansion of almost every real number in $[0,1)$ *is a normal word in the alphabet* $\{0,1,...,9\}$.

Nevertheless, not so many examples have been proved normal. Some of them are:

- Champernowne 1933 (natural numbers):

$$12345678910111213141516171819202122232425\cdots$$

- Besicovitch 1935 (squares):

$$149162536496481100121144169196225256289324\cdots$$

- Copeland and Erdős 1946 (primes):

$$235711131719232931374143475359616771737983\cdots$$

Normality as randomness

Normality is the poor mans's randomness. This is the least requirement one can expect from a random sequence.

This is much weaker than Martin-Löf randomness which implies non-computability.

Outline

Selection rules

- If $x = a_1a_2a_3\cdots$ is a normal infinite word, then so is $x' = a_2a_3a_4\cdots$ made of symbols at all positions but the first one.
- If $x = a_1a_2a_3\cdots$ is normal infinite word, then so is $x' = a_2a_4a_6\cdots$ made of symbols at even positions.
- What about selecting symbols at positions 2^n ?
- What about selecting symbols at prime positions ?
- What about selecting symbols following a 1 ?
- What about selecting symbols followed by a 1 ?

Oblivious prefix selection

Let $L \subseteq A^*$ be a set of finite words and $x = a_1 a_2 a_3 \cdots \in A^\omega$.
The prefix selection of x by L is the word $x \upharpoonright L = a_{i_1} a_{i_2} a_{i_3} \cdots$ where $\{i_1 < i_2 < i_3 < \cdots\} = \{i : a_1 a_2 \cdots a_{i-1} \in L\}$.

Example (Symbols following a 1)

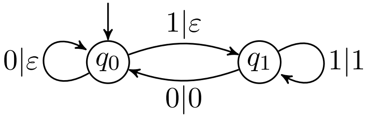
If $L = (0+1)^*1$, then $i_1 - 1, i_2 - 1, i_3 - 1$ are the positions of 1 in x and $x \upharpoonright L$ is made of the symbols following a 1.

Theorem (Agafonov 1968)

Prefix selection by a rational set of finite words preserves normality.

The selection can be realized by a transducer.

Example (Selection of symbols following a 1)



Oblivious suffix selection

Let $X \subseteq A^\omega$ be a set of infinite words and $x = a_1 a_2 a_3 \cdots \in A^\omega$. The suffix selection of x by X is the word $x \upharpoonright X = a_{i_1} a_{i_2} a_{i_3} \cdots$ where $\{i_1 < i_2 < i_3 < \cdots\} = \{i : a_{i+1} a_{i+2} a_{i+3} \cdots \in X\}$.

Example (Symbols followed by a 1)

If $L = 1(0+1)^\omega$, then $i_1 + 1, i_2 + 1, i_3 + 1$ are the positions of 1 in x and $x \upharpoonright X$ is made of the symbols followed by a 1.

Theorem

Suffix selection by a rational set of infinite words preserves normality.

Combining prefix and suffix does not preserve normality in general. Selecting symbols having a 1 just before and just after them does not preserve normality.

Outline

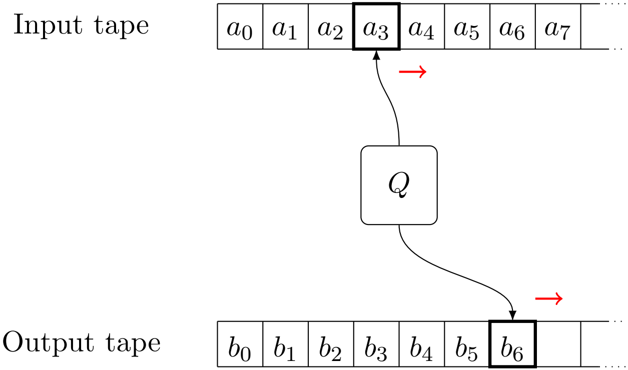
Normality

Selection

Compressibility

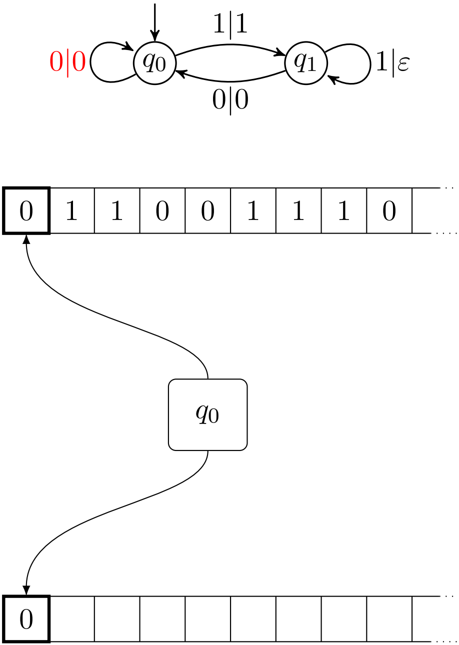
Weighted automata and frequencies

Transducers

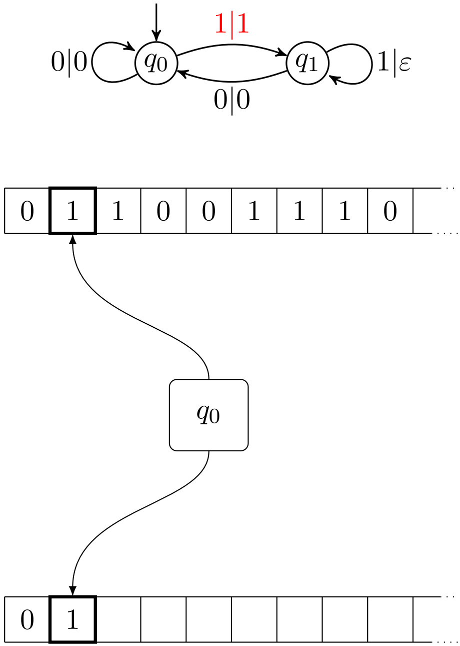


Transitions $p \xrightarrow{a|v} q$ for $a \in A$, $v \in B^*$.

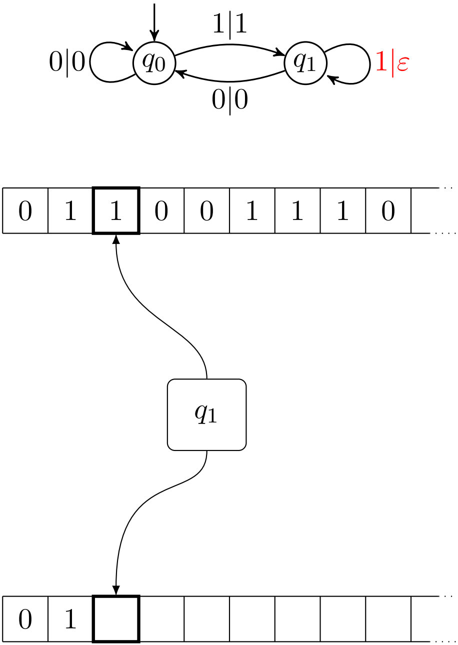
Example



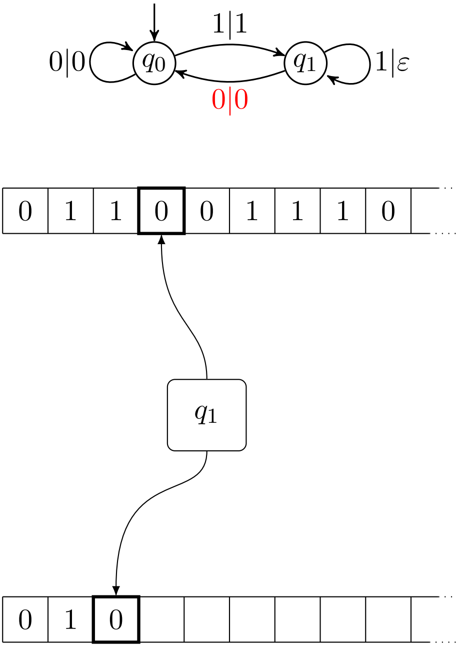
Example



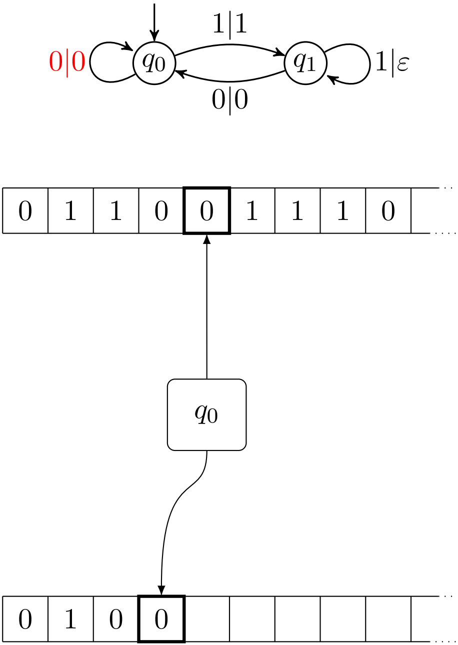
Example



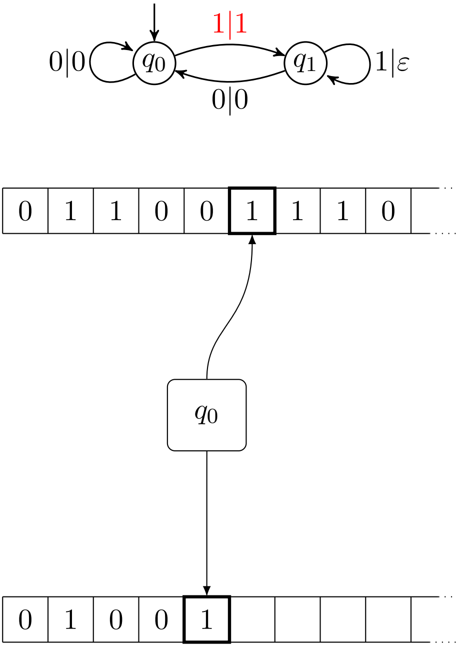
Example



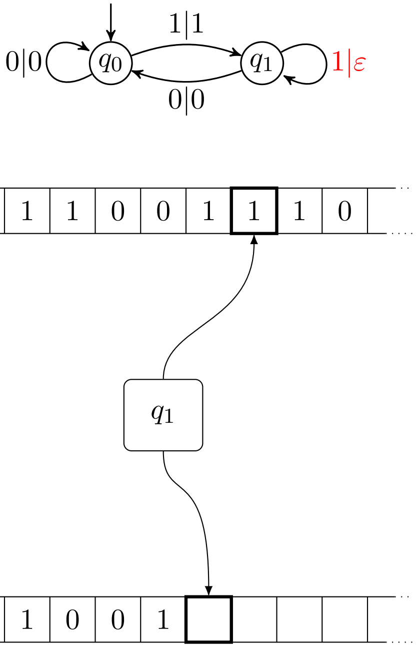
Example



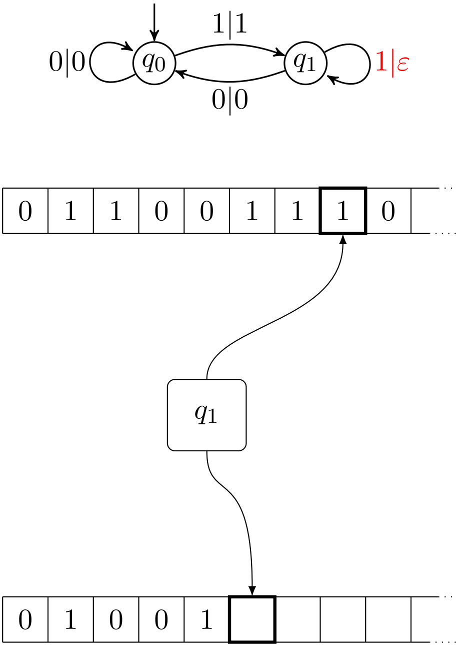
Example



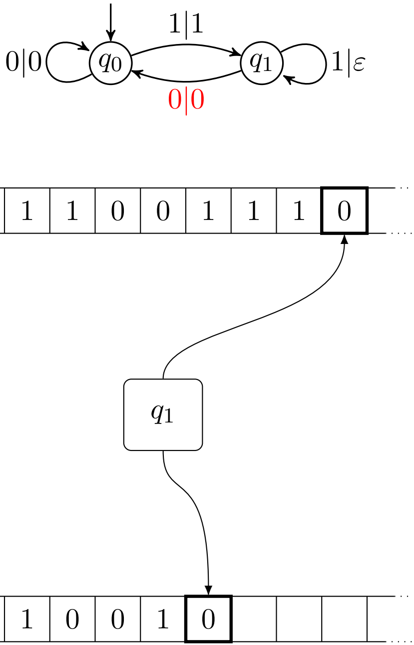
Example



Example



Example



Characterization of normal words

An infinite word $x = a_1a_2a_3\cdots$ is *compressible* by a transducer if there is an accepting run $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$ satisfying

$$\liminf_{n\to\infty} \frac{|v_1v_2\cdots v_n| \log |B|}{|a_1a_2\cdots a_n| \log |A|} < 1.$$

Theorem (Schnorr, Stimm and others)

An infinite word is normal if and only if it cannot be compressed by deterministic one-to-one transducers.

Similar to the characterization of Martin-Löf randomness by non-compressibility by prefix Turing machines.

$$\liminf_{n\to\infty} \mathcal{H}(x[1..n]) - n > -\infty$$

where $\mathcal{H}$ is the prefix Kolmogorov complexity.

Ingredients

Shannon (1958)

- frequency of u different from $b^{-|u|}$ implies non maximum entropy
- non-maximum entropy implies compressibility

Huffman (1952)

- simple greedy implementation of Shannon's general idea
- implementation by a finite state tranducer

Robust characterization

Transducers can be replaced by

- Non-deterministic but functional one-to-one transducers
- Transducers with one counter
- Two-way transducers

	det	non-det	non-rt
finite-state	N	N	N
1 counter	N	N	N
≥ 2 counters	N	N	T
1 stack	?	C	C
1 stack + 1 counter	C	C	T

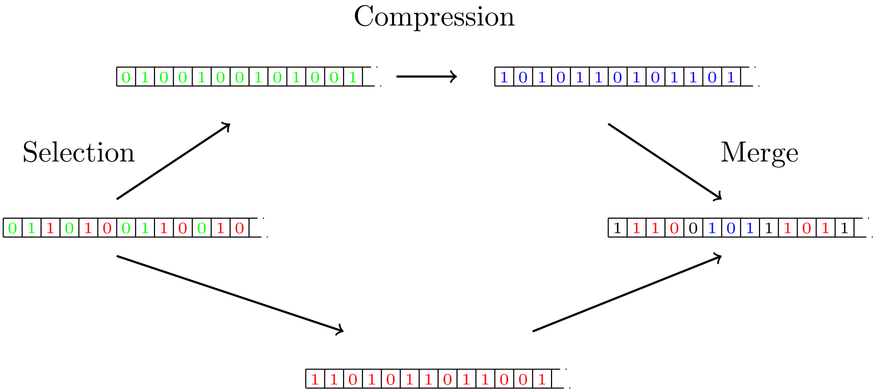
where

N means *cannot compress normal words*

C means *can compress some normal word*

T means *is Turing complete* and thus can compress.

Non-compressibility implies selection



Outline

Normality

Selection

Compressibility

Weighted automata and frequencies

Preservation of normality

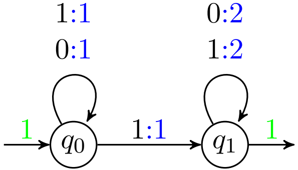
A functional transducer $\mathcal{T}$ is said to preserve normality if for every normal word $x \in A^\omega$, $\mathcal{T}(x)$ is also normal.

Question

Given a deterministic complete transducer $\mathcal{T}$, does $\mathcal{T}$ preserve normality?

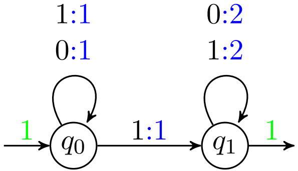
Weighted Automata

A weighted automaton $\mathcal{T}$ is an automaton whose transitions, not only consume a symbol from an input alphabet A, but also have a transition weight in $\mathbb{R}$ and whose states have initial weight and final weight in $\mathbb{R}$.



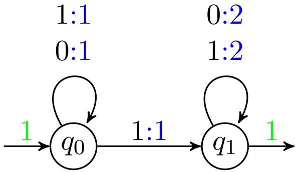
This weighted automaton computes the value of a binary number.

The weight of a run $q_0 \xrightarrow{b_1} q_1 \xrightarrow{b_2} \cdots \xrightarrow{b_n} q_n$ in $\mathcal{A}$ is the product of the weights of its n transitions times the initial weight of q_0 and the final weight of q_n.



$$\text{weight}_{\mathcal{A}}(q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2) = 1 * 1 * 1 * 2 * 1 = 2$$

The weight of a run $q_0 \xrightarrow{b_1} q_1 \xrightarrow{b_2} \cdots \xrightarrow{b_n} q_n$ in $\mathcal{A}$ is the product of the weights of its n transitions times the initial weight of q_0 and the final weight of q_n.



The weight of a word w in $\mathcal{A}$ is given by the sum of weights of all runs labeled with w:

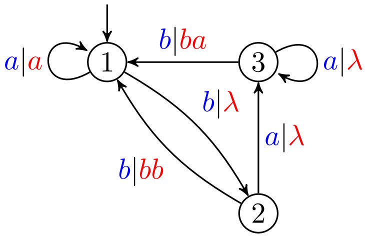
$$\text{weight}_{\mathcal{A}}(w) = \sum_{\gamma \text{ run on } w} \text{weight}_{\mathcal{A}}(\gamma)$$

$$\begin{aligned}\text{weight}_{\mathcal{A}}(110) \;=\; & \text{weight}_{\mathcal{A}}(q_0 \xrightarrow{1} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_1) + \\ & \text{weight}_{\mathcal{A}}(q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_1) \quad = 2 + 4 = 6\end{aligned}$$

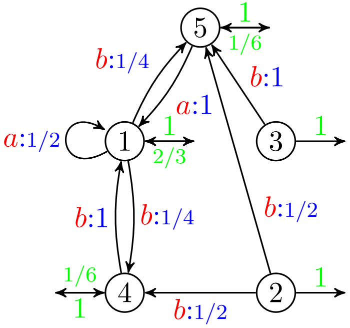
Theorem

For every strongly connected deterministic transducer $\mathcal{T}$ there exists a weighted automaton $\mathcal{A}$ such that for any finite word w and any normal word x, $\mathrm{weight}_{\mathcal{A}}(w)$ is exactly the frequency of w in $\mathcal{T}(x)$.

Example



Transducer $\mathcal{T}$



Weighted Automaton $\mathcal{A}$

Deciding preservation of normality

Proposition

Such a weighted automaton can be computed in cubic time with respect to the size of the transducer.

Theorem

It can decided in cubic time whether a given deterministic transducer does preserve normality (that is sends each normal word to a normal word)

.

Recap of the links between automata and normality

- Selecting with an automaton in an normal word preserves normality.
- Normality is characterized by non-compressibility by finite state machines.
- Frequencies in the output of a deterministic transducer are given by a weighted automaton.

Thank you