Departamento de Computación, Facultad de Ciencias Exactas y Naturales, UBA

Azar y Autómatas

Clase 7: Selección en secuencias normales. Independencia

Selection

Select a subsequence of symbols of an infinite word. Which selecting functions f guarantee that $f(x)$ is normal when x is normal?

Notice that if a selection procedure is allowed to read the symbol being decided, it would be possible to "select only zeroes", or yield similar schemes that do not preserve normality.

Proefix selection

Prefix-selection considers just the prefix of length $i-1$ to decide whether the symbol at position i is selected.

Let $x = a_1 a_2 \cdots$ be an infinite word over alphabet A.

Let $L \subseteq A^*$ be a set of finite words over alphabet A.

The prefix-selection of x by L is $x \upharpoonright L = a_{i_1} a_{i_2} a_{i_3} \cdots$ where $i_1, i_2, \cdots$ is the enumeration in increasing order of all the integers i such that $a_1 a_2 \cdots a_{i-1} \in L$.

Example:
$x = 010010001000010000010000001000000001\ldots$
$L = (0^*1)^*$,
$x \upharpoonright L = 00000000\ldots$

$\overline{L} = (A^* \setminus L) = (0^*1^*)^*0$
$x \upharpoonright \overline{L} = 101010010001\ldots$

Lemma 1

For any regular set L *of words , the function* $x \mapsto \langle x \upharpoonright L, x \upharpoonright A^* \setminus L \rangle$ *is one-to-one.*

Proof.

Let $y_1 = x \upharpoonright L$ and $y_2 = x \upharpoonright A^* \setminus L$.

By definition, both y_1 and y_2 contain some symbols of x, in the same relative order as they are in x.

It is possible to reconstruct x by interleaving y_1 and y_2 appropriately: For each $i \geq 0$, the symbol at position i of x comes from y_1 if and only if $x[1..i-1]$ is in L.

Thus, there is a unique x such that $y_1 = x \upharpoonright L$ and $y_2 = x \upharpoonright A^* \setminus L$.

□

Selection by finite automata preserves normality

Theorem 1 (Agafonov 1968)

Let $x \in A^\omega$ be normal and let $L \subset A^*$ be regular. Then $x \upharpoonright L$ is normal.

Proof of Theorem 1

Let x be a normal word. Let $L \subset A^*$ be a regular language.

Suppose $x \upharpoonright L$ is not normal. Then, $x \upharpoonright L$ is compressible and we will prove that we can also compress x which contradicts normality of x.

Ingredients:

1. a finite state automaton that accepts $L \subseteq A^*$ (one input)
2. a splitter (one input, two outputs, $x \upharpoonright L$, $x \upharpoonright \overline{L}$)
3. a compressor to compress $x \upharpoonright L$ (one input, one output)
4. another compresor (one input, two outputs) that compresses x, by compressing just $x \upharpoonright L$

Automata with one input and two outputs

A (deterministic) one-input and two-output transducer $\langle Q, A, \delta, I, F\rangle$ is such that the transitions are the form $p \xrightarrow{a|v,w} q$ where a is the symbol read on the input tape and v and w are the words written to the first and the second output tape respectively. We are interested in automatas of this type only when they are are bounded to one.

An infinite word $x = a_1a_2\cdots$ is compressible by a two-output transducer if there is an accepting run

$$q_0 \xrightarrow{a_1|v_1,w_1} q_1 \xrightarrow{a_2|v_2,w_2} q_2 \xrightarrow{a_3|v_3,w_3} \cdots$$

that satisfies

$$\liminf_{n\to\infty} \frac{(|v_1v_2\cdots v_n| + |w_1w_2\cdots w_n|)}{n} < 1.$$

Proof of Theorem 1

Let x be a normal word. Let $L \subset A^*$ be a regular language.

Suppose $x \upharpoonright L$ is not normal. Then, $x \upharpoonright L$ is compressible and we will prove that we can also compress x.

Let $\mathcal{A}$ be a deterministic automaton accepting L.

Let $\mathcal{S}$ be a two-output transducer (splitter) such that on input x outputs $x \upharpoonright L$ and $x \upharpoonright A^* \setminus L$. Each transition that leaves a final state of $\mathcal{A}$, copies its input symbol to the first output tape and each transition that leaves a non-final state of $\mathcal{A}$ copies its input symbol to the second output tape.

Let $\mathcal{C}$ be a one-to-one determinsitic compressor that compresses $x \upharpoonright L$.

Let $\mathcal{T}$ be the following deterministic one-to-one transducer that compresses x. Assume input x. It simulates the splitter $\mathcal{S}$ using two buffers of length m large enough, one buffer places $x \upharpoonright L$, the other $x \upharpoonright A^* \setminus L$. Run the compressor $\mathcal{C}$ on the buffer that holds $x \upharpoonright L$.
Claim (proved in Lemma 2): the states that select symbols from x are visited in the run linearly often.
Claim (proved in Lemma 3): x can be compressed and is not normal.
□

Key trivial observation

An infinite word $x = a_1, a_2 \dots$ is normal in alphabet A if, and only if, for any length ℓ, for any $w \in A^\ell$,

$$\begin{aligned}
\lim_{n\to\infty} \frac{\|a_1 \dots a_{n\ell}\|_w}{n} \qquad & = \frac{1}{|A|^\ell} \qquad \text{iff} \\
\forall \varepsilon \exists n_0 \forall n \geq n_0 \qquad & \left| \|a_1 \dots a_{n\ell}\|_w - \frac{n}{|A|^\ell} \right| < \varepsilon n \qquad \text{iff} \\
\forall \varepsilon \exists n_0 \forall n \geq n_0 \qquad & \frac{n}{|A|^\ell} - \varepsilon n < \|a_1 \dots a_{n\ell}\|_w < \frac{n}{|A|^\ell} + \varepsilon n.
\end{aligned}$$

States visited infinitely often are visited linearly often

Lemma 2

Let $x = a_1 a_2 \cdots$ be a normal word and let $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \xrightarrow{a_3} \cdots$ be a run in a deterministic automaton $\langle Q, A, \delta, q_0, F \rangle$. If the state $q \in Q$ is visited infinitely often then

$$\liminf_{n\to\infty} \frac{|\{i : 0 \leq i < n : q_i = q\}|}{n} > 0.$$

Proof of Lemma 2

Sea $\mathcal{A}$ un autómata finito determinístico que acepta x. Para cada estado p y cada palabra w, hay a lo sumo un único estado $p \xrightarrow{w} q$ al que escribimos $p \cdot w$. Sean $q_1, \ldots, q_r$ los estados que ocurren infinitamente en la corrida con entrada x.

Definamos w_0 la palabra tal que $q_1 \cdot w_0 = q_r$. y Definamos las palabras $w_1, .., w_r$ tales que $q_k \cdot w_k = q_1$: Sea $u_{i,j}$ las palabras tales que $q_i \xrightarrow{u_{i,j}} q_j$. Definamos w_k que $w_1 = \lambda$, $w_2 = u_{2,1} w_1$, $w_3 = u_{3,2} u_{2,1} = u_{32} w_2$, $w_4 = u_{43} w_3, \ldots,$ $w_r = u_{r,r-1} w_{r-1}$.

Luego $q_1 \xrightarrow{w_0 w_r} q_1 \cdot w_0 w_r = q_1$.
Llamemos $\ell = |w_0 w_r|$. Como $w_0 w_r$ ocurre en x con frecuencia $|A|^{-\ell}$

$$\liminf_{n\to\infty} \frac{\{i : 0 \leq i < n : q_i = q_1\}}{n} \geq \liminf_{n\to\infty} \frac{\|x[1, n\ell]\|_{w_0 w_r}}{n} \geq |A|^{-\ell}.$$

An extra tape does not help to compress better

Lemma 3

An infinite word is compressible by a bounded-to-one two-output transducer if and only if it is compressible by a bounded-to-one transducer.

Proof of Lemma 3

The "if" part is immediate by not using one of the output tapes.

Suppose that x is compressible by the bounded-to-one two-output transducer $\mathcal{T}_2$. We construct a transducer $\mathcal{T}_1$ with a single output tape which also compresses x.

The idea is to merge the two outputs into the single tape.
Let m be an integer to be fixed later.
$\mathcal{T}_1$ simulates $\mathcal{T}_2$ on the input and uses two buffers of size m to store the outputs made by $\mathcal{T}_2$. Whenever one of the two buffers is full and contains m symbols, its content is copied to the output tape of $\mathcal{T}_1$ with an additional symbol in front of it. This symbol is either 0 or 1, to indicate whether the m following symbols comes from the first or the second buffer.

This trick preserves the bounded-to-one assumption. The additional symbol for each block of size m increases the length of the output by a factor $(m + 1)/m$. For m large enough, the transducer $\mathcal{T}_1$ also compresses x. □

Other forms of selection

Let $x = a_1 a_2 \cdots$ be an infinite word over alphabet A. Let $L \subseteq A^*$ be a set of finite words over A and $X \subseteq A^\omega$ a set of infinite words over A.

Prefix-selection looks at just the prefix of length $i-1$ to decide whether the symbol at position i is selected.

Suffix selection looks at just the suffix starting at position $i+1$ to decide whether symbol at position i is selected. The word obtained by suffix-selection of x by X is $x \upharpoonright X = a_{i_1} a_{i_2} a_{i_3} \cdots$ where $i_1, i_2, \cdots$ is the enumeration in increasing order of all the integers i such that $a_{i+1} a_{i+2} a_{i+3} \cdots \in X$.

Two-sided selection looks at the prefix of length $i-1$ and the suffix starting at position $i+1$ to decide the selection of the symbol at position i.

Suffix selection by a regular language preserves normality

Theorem 2 (Becher, Carton and Heiber 2015)

If $x \in A^\omega$ is normal and $X \subset A^\omega$ is regular then $x \upharpoonleft X$ is also normal.

Two sided selection preserves normality?

Theorem 3

The two-sided selection rule "select symbols in between two zeroes" does not preserve normality.

Proof of Theorem 3

Let $x = a_1 a_2 a_3 \cdots$ be a normal infinite word over $\{0, 1\}$ and let y be the result of selecting all symbols between two zeroes, namely $y = a_{p(1)} a_{p(2)} a_{p(3)} \cdots$ where $p(j)$ is the j-th smallest integer in $\{i : a_{i-1} = a_{i+1} = 0\}$.

We show that y is not normal. Let m_n be the length of the shortest prefix of x that contains n instances of 000 or 010,

$$m_n = \min\{m : |\{i : 2 \leq i \leq m-1, a_{i-1} = a_{i+1} = 0\}| = n\}.$$

Let $y = b_1 b_2 b_3 \cdots$ and $k_n = |\{i : 1 \leq i \leq n-1, b_i b_{i+1} = 00\}|$.

Proof of Theorem 3

By definition of m_n and y,

$$\begin{aligned}
k_n &\geq |\{i : 1 \leq i \leq m_n - 3, a_i a_{i+1} a_{i+2} a_{i+3} = 0000\}| \\
&\quad + |\{i : 1 \leq i \leq m_n - 8, a_i a_{i+1} a_{i+2} a_{i+3} a_{i+4} a_{i+5} a_{i+6} a_{i+7} = 00011000\}|. \\
\lim_{n\to\infty} \frac{k_n}{n} &\geq \lim_{n\to\infty} \frac{|\{i : 1 \leq i \leq m_n - 3, a_i a_{i+1} a_{i+2} a_{i+3} = 0000\}|}{n} \\
&\quad + \frac{|\{i : 1 \leq i \leq m_n - 8, a_i a_{i+1} a_{i+2} a_{i+3} a_{i+4} a_{i+5} a_{i+6} a_{i+7} = 00011000\}|}{n} \\
&> \lim_{n\to\infty} \frac{|\{i : 1 \leq i \leq m_n - 3, a_i a_{i+1} a_{i+2} a_{i+3} = 0000\}|}{n} \\
&= \lim_{n\to\infty} \frac{|\{i : 1 \leq i \leq m_n - 3, a_i a_{i+1} a_{i+2} a_{i+3} = 0000\}|}{m_n} \frac{m_n}{n}.
\end{aligned}$$

By definition of normality and the properties of limit,

$$\lim_{n\to\infty} \frac{|\{i : 1 \leq i \leq m_n - 3, a_i a_{i+1} a_{i+2} a_{i+3} = 0000\}|}{m_n} = \frac{1}{2^4} \quad \text{and} \quad \lim_{n\to\infty} \frac{m_n}{n} = 2^2,$$

Then,

$$\lim_{n\to\infty} k_n/n > 2^{-4}\ 2^2 = 1/4,$$

which implies that y is not normal. □

Independence of normal words

When are two normal words independent?

Independence of normal words

First attempt of a definition of independence (it fails):
Two normal words are independent exactly when their join is normal.

Theorem 4 (Becher, Carton and Heiber 2016)

There are two normal words x and y such that x join $y = x$.

Here $x = \mathit{even}(x)$ and $y = \mathit{odd}(x)$, hence they are obviously dependent.

Theorem 5 (Shen 2016)

Let $x_1, x_3, x_5, \ldots$ be uniformly distributed independent symbols from $\{0, 1\}$ and for every odd n, let $x_n = x_{2n} = x_{4n} = \ldots$. Then, with probability 1 the resulting word $x_1 x_2 x_3 \ldots$ is normal.

Independence of normal numbers

Two normal words are independent exactly when one does not help to compress the other.

A deterministic finite transducer with 2 input tapes and 1 output tape is a tuple $\mathcal{A} = \langle Q, A, \delta, q_0 \rangle$, where

- Q is the finite state set,
- A is the alphabet,
- $\delta : Q \times (A \cup \{\lambda\}) \times (A \cup \{\lambda\}) \to A^* \times Q$ is the transition function where a transition is written $p \xrightarrow{\alpha,\beta|\gamma} q$,
- q_0 is the initial state.

A run with inputs x and y is a sequence of consecutive transitions

$$q_0 \xrightarrow{\alpha_1,\beta_1|\gamma_1} q_1 \xrightarrow{\alpha_2,\beta_2|\gamma_2} q_2 \cdots$$

We write $\mathcal{A}(x, y) = \gamma_1\gamma_2\gamma_3 \cdots$.
We say $\mathcal{A}$ is one-to-one if for each y fixed, $x \to \mathcal{A}(x, y)$ is one-to-one.

Independence of normal numbers

Let $\mathcal{A}$ be a finite transducer with two input tapes, deterministic and one-to-one. Suppose inputs x and y and the run in $\mathcal{A}$

$$q_0 \xrightarrow{\alpha_1,\beta_2|\gamma_1} q_1 \xrightarrow{\alpha_2,\beta_2|\gamma_2} q_2 \xrightarrow{\alpha_3,\beta_3|\gamma_3} q_3 \cdots$$

where $x = \alpha_1\alpha_2\ldots$ and $y = \beta_1\beta_2\ldots$

The conditional compression ratio of x with respect to y in $\mathcal{A}$ is

$$\rho_{\mathcal{A}}(x/y) = \liminf_{n\to\infty} \frac{|\gamma_1\ldots\gamma_n|}{|\alpha_1\ldots\alpha_n|}.$$

Notice that the number of symbols read from y, namely $|\beta_1\ldots\beta_n|$, is not taken into account in the value of $\rho_{\mathcal{A}}(x/y)$.

The conditional compression ratio of x given y, $\rho(x/y)$, is the infimum of $\rho_{\mathcal{A}}(x/y)$ for all $\mathcal{A}$ deterministic one-to-one.

Independence of normal numbers

Two words x and y are independent if their compression ratios are not 0 and y does not help to compress x and x does not help to compress y,

$$\rho(x) = \rho(x/y) > 0 \text{ and } \rho(y) = \rho(y/x) > 0.$$

Independence of normal numbers

Theorem 6 (Becher and Carton 2016)

The set $\{(x, y) : x \text{ and } y \text{ are independent}\}$ *has Lebesgue measure* 1.

Lemma 4

The set of words that are compressible with the help of a given normal word has Lebesgue measure 0.

Independence of normal numbers

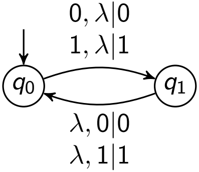
A shuffler $\mathcal{S} = \langle Q, A, \delta, q_0 \rangle$ is a finite transducer with two input tapes and one output tape. The transition function is $\delta : Q \times A \cup \{\lambda\} \times A \cup \{\lambda\} \to Q \times A$, transitions have the form

$$p \xrightarrow{a,\lambda|a} q \quad \text{or} \quad p \xrightarrow{\lambda,a|a} q.$$

For each state q, all incoming transitions have the same type.
Whether the next digit is taken from the first or the second input word only depends the current state.

Independence of normal numbers

Example of a Shuffler that computes the join



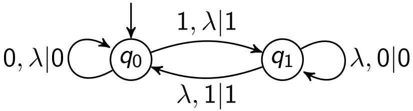
$x = 0011010001\ldots$

$y = 0100011000\ldots$

x join $y = 00011010001101000010\ldots$

Independence of normal numbers

Example of another shuffler. It alternates (possibly empty) blocks of 0s followed by a 1, from each input word.



Input words $\begin{cases} x = 001\;1\;01\;0001\ldots \\ y = 01\;0001\;1\;0001\ldots \end{cases}$

Output word $z = 001011000101100010001\ldots$

Independence of normal numbers

Theorem 7 (Alvarez, Becher and Carton 2016)

Two normal words are independent if and only if every shuffling is normal.

Theorem 8 (Alvarez, Becher and Carton 2016)

There is an algorithm that computes two normal independent words.

Independence of normal numbers

Problem 9

Give combinatorial characterization of finite-state independence.

Problem 10

Construct a normal word that is independent of Champernowne.