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

Azar, Algoritmos y Autómatas

Clase 1: Introducción

azar - aléatoire - Zufall - rasgelelik - satunnaisuuden - slumpmässighet - randomness - aleatorietà

Todos tenemos una idea intuitiva acerca de lo que es el azar, típicamente relacionada con los “juegos de azar” o con la “suerte”...

En castellano azar y aleatoriedad son sinónimos.
En inglés se dice “random”.

La suerte es loca

¿Creerían que se obtienen echando una moneda para cada posición?

11111111111111111111111111111111111... ✗
¡Son todos unos!

01001000100001000001000000100000001... ✗
¡Esta secuencia tiene un patrón!

00101001010001101110100010010101111...

Azar es imposibilidad de predecir, es falta de patrón.

La suerte es equitativa (a la larga)

Azar es imposibilidad de predecir, es falta de patrón.

Entonces cara y ceca deben ocurrir, a la larga, la misma cantidad de veces

Sino, podríamos aprovecharnos del desvío y bastantes veces podríamos predecir bien.

La suerte es equitativa (a la larga)

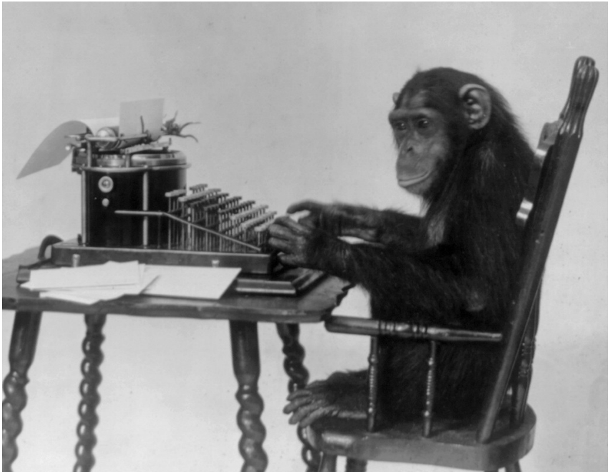
En vez de echar una moneda repartamos cartas.
Si jugamos el suficiente tiempo, y nadie hace trampa, alguna vez me tocarán el ancho de espadas, el ancho de basto y el 7 de espadas.

Un mono y una máquina de escribir

Teorema (Émile Borel 1913)

Si un mono se sienta en una máquina de escribir por siempre jamás escribirá todos los posibles textos, infinitas veces cada uno.

mono escribiendo = secuencia azarosa de símbolos



Émile Borel. La mécanique statique et l'irréversibilité.
Journal de Physique Théorique et Appliquée, 1913, 3 (1), pp.189-196.

> *[. . .] Concevons quon ait dressé un million de singes à frapper au hasard sur les touches d'une machine à écrire et que, sous la surveillance de contremaîtres illettrés, ces singes dactylographes travaillent avec ardeur dix heures par jour avec un million de machines à écrire de types variés. Les contre-maitres illettrés rassembleraient les feuilles noircies et les relieraient en volumes. Et au bout d'un an, ces volumes se trouveraient renfermer la copie exacte des livres de toute nature et de toutes langues conservés dans les plus riches bibliothéques du monde.*

... Burns tenía mil monos con mil máquinas de escribir




Sobre el azar

- ¿Hay una definición matemática de azar?

Sobre el azar

- ¿Hay una definición matemática de azar?
- ¿Podemos dar ejemplos?

Sobre el azar

- ¿Hay una definición matemática de azar?
- ¿Podemos dar ejemplos?
- ¿Hay grados de azar?

Sobre el azar

- ¿Hay una definición matemática de azar?
- ¿Podemos dar ejemplos?
- ¿Hay grados de azar?
- ¿Puede una computadora producir una secuencia puramente al azar?

Sobre el azar

- ¿Hay una definición matemática de azar?
- ¿Podemos dar ejemplos?
- ¿Hay grados de azar?
- ¿Puede una computadora producir una secuencia puramente al azar?
- ¿Podemos garantizar azares independientes?

Hacia una definición matemática de azar

Azar es imposibilidad de predecir. Equivalentemente, azar es imposibilidad de abreviar, imposibilidad de comprimir.

Pero . . .

Hacia una definición matemática de azar

Azar es imposibilidad de predecir. Equivalentemente, azar es imposibilidad de abreviar, imposibilidad de comprimir.
Pero . . . ¿Para qué habilidades?

Aquí entran en escena las ciencias de la computación, ya que hay distintos modelos de cómputo.

Modelos de cómputo

Distintos modelos de cómputo tienen distintas capacidad de resolver problemas.

- Autómatas finitos
- Autómatas de pila
- Las computadoras actuales (Máquinas de Turing)

Hacia una definición matemática de azar

Una secuencia es azarosa (para los autómatas de la clase $\mathcal{C}^{\text{‘}}$) cuando, esencialmente, la única forma de describirla (mediante un autómata de la clase $\mathcal{C}^{\text{‘}}$) es nombrando explícitamente cada uno de sus símbolos.

Grados de azar

Azar puro: impredecibilidad/incompresibilidad para máquinas de Turing.

Azar básico: impredecibilidad/incompresibilidad para autómatas finitos.

Hay azares intermedios

¿Puede una computadora producir azar puro?

¿Puede una computadora producir azar puro?

¡No!

¿Puede una computadora producir azar puro?

¡No!

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

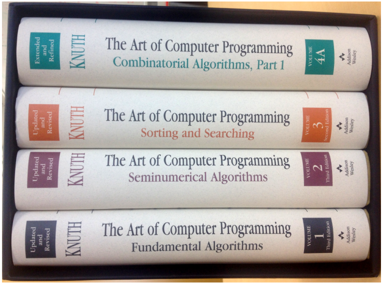
John von Neumann, 1951

¿Puede una computadora producir azar puro?

¡No!

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

John von Neumann, 1951 (cita Knuth, The Art of Computing Programming)



¿Puede una computadora producir azar puro?

Toda secuencia computable es dramáticamente compresible por una máquina de Turing!

¿Puede una computadora producir azar puro?

Toda secuencia computable es dramáticamente compresible por una máquina de Turing! Hay un programa que arroja uno tras otros los símbolos de la secuencia. Luego hay oro programa al que le indico cuántos símblos iniciales de la secuencia imprimir, Entonces:

¿Puede una computadora producir azar puro?

Toda secuencia computable es dramáticamente compresible por una máquina de Turing! Hay un programa que arroja uno tras otros los símbolos de la secuencia. Luego hay oro programa al que le indico cuántos símblos iniciales de la secuencia imprimir, Entonces: Cada segmento inicial de longitud n se puede obtener medianet un program que mide $2 \log n +$constante. Es muy compresible.
El azar puro no es computable.
Sin embargo, una secuencia puede ser no computable y no azarosa.

Towards a definition of randomness

A sequence is random if, essentially, its initial segments can only be described explicitely by a Turing machine. (Chaitin's definition 1975)

Towards a definition of randomness

A sequence is random if, essentially, its initial segments can only be described explicitely by a Turing machine. (Chaitin's definition 1975)

A sequence is normal if, essentially, its initial segments can only be described explicitely by a finite automaton .

(Borel's definition 1909; Schnorr and Stimm 1971; Dai Lathroup Lutz and Mayordomo 2005)

Normality also has an equivalent that does not invlove a machine, it is purely combinatorial. In fact, this is the original formulation given by Borel.

Sequences and real numbers

A base is an integer greater than or equal to 2.

For a real number x in the unit interval, the expansion of x in base b is a sequence $a_1 a_2 a_3 \ldots$ of integers from $\{0, 1, \ldots, b-1\}$ such that

$$x = 0.a_1 a_2 a_3 \ldots$$

where $x = \sum_{k \geq 1} \frac{a_k}{b^k}$, and x does not end with a tail of $b - 1$.

Pure randomness

A sequence is random if its initial segments can only be described explicitely by a Turing machine. That is, its initial segments cannot be compressed with a Turing machine.

Pure randomness

A sequence is random if its initial segments can only be described explicitely by a Turing machine. That is, its initial segments cannot be compressed with a Turing machine.

Formally, a sequence is random if its initial segments have almost maximal program-size complexity .

Kolmogorov / program-size complexity

Some long strings can be described using fewer symbols than their length; this is used in data compression .

For example, string consisting of 2^n many a's can be encoded as $\log n$ many symbols plus a constant:

input n
i=0;
while (i<2^n) {print a; i=i+1;}

Kolmogorov / program-size complexity

Definition (Kolmogorov 1965)

Fix a universal Turing machine U. The Kolmogorov complexity of a string s is the length of the shortest input in U that outputs s.

Kolmogorov / program-size complexity

Definition (Kolmogorov 1965)

Fix a universal Turing machine U. The Kolmogorov complexity of a string s is the length of the shortest input in U that outputs s.

For every string s, its Kolmogorov complexity is less than $|s| + constant$.

Kolmogorov / program-size complexity

Definition (Kolmogorov 1965)

Fix a universal Turing machine U. The Kolmogorov complexity of a string s is the length of the shortest input in U that outputs s.

For every string s, its Kolmogorov complexity is less than $|s| + constant$.

Definition (Chaitin 1975)

Fix a universal Turing machine U with prefix-free domain .
The program-size complexity of a string s, $K(s)$, is the length of the shortest input in U that outputs s.

For every string s, $K(s) \leq |s| + 2\log|s| + constant$.

The definition of randomness

Definition (Chaitin 1975)

A sequence $a_1a_2a_3\ldots$ is random if $\exists c\ \forall n\ K(a_1a_2\ldots a_n) > n - c$.

The definition applies immediately to real numbers (one-to-one correspondence between reals and their expansions in any given base).

How do we know that the definition is right?

How do we know that the definition is right?

The definition of randomness was accepted when two different formulations were shown to be equivalent.

This is similar to what happenned with the notion of algorithm in 1930s with Church-Turing thesis.

An equivalent definition of randomness

Definition (Martin-Löf 1965, tests of non-randomness)

A sequence is Martin-Löf random if it passes all computably definable tests of non-randomness. Since there is a universal tests, it suffices that to consider just this universal Martin-Löf test.

An equivalent definition of randomness

Definition (Martin-Löf 1965, tests of non-randomness)

A sequence is Martin-Löf random if it passes all computably definable tests of non-randomness. Since there is a universal tests, it suffices that to consider just this universal Martin-Löf test.

Technically, a sequence is Martin-Löf random if it belongs to no computably definable null set. Since there is a universal computably definable null set, it suffices to consider this one.

An equivalent definition of randomness

Definition (Martin-Löf 1965, tests of non-randomness)

A sequence is Martin-Löf random if it passes all computably definable tests of non-randomness. Since there is a universal tests, it suffices that to consider just this universal Martin-Löf test.

Technically, a sequence is Martin-Löf random if it belongs to no computably definable null set. Since there is a universal computably definable null set, it suffices to consider this one.

Theorem (Schnorr 1975)

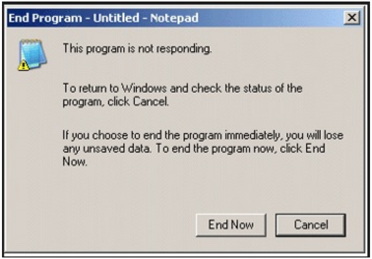
A sequence is random for Chaitin's definition if and only if it does not belong to the universal Martin-Löf null set.

Examples of random sequences

Have you ever experienced that your computer locked up (froze)?

Examples of random sequences

Have you ever experienced that your computer locked up (froze)?



Ω-numbers

Theorem (Chaitin 1975)

The probability that a universal Turing machine with prefix-free domain halts, $\Omega = \sum_{U(p)\textit{halts}} 2^{-|p|}$ *is random.*

Similiarly, probabilities of other computer behaviours called Ω numbers (Becher,Chaitin 2001,2003; Becher,Grigorieff 2005,2009, Becher,Figueira,Grigorieff,Miller 2006; Barmpalias 2016)

Normal numbers, the most basic form of randomness

Definition (Borel, 1909)

A real number x is simply normal to base b if, in the expansion of x in base b, each digit occurs with limiting frequency equal to $1/b$.

Normal numbers, the most basic form of randomness

Definition (Borel, 1909)

A real number x is simply normal to base b if, in the expansion of x in base b, each digit occurs with limiting frequency equal to $1/b$.

A real number x is normal to base b if, for every positive integer k, every block of k digits (starting at any position) occurs in the expansion of x in base b with limiting frequency $1/b^k$.

Normal numbers, the most basic form of randomness

Definition (Borel, 1909)

A real number x is simply normal to base b if, in the expansion of x in base b, each digit occurs with limiting frequency equal to $1/b$.

A real number x is normal to base b if, for every positive integer k, every block of k digits (starting at any position) occurs in the expansion of x in base b with limiting frequency $1/b^k$.

A real number x is absolutely normal if x is normal to every base.

Not normal

$0.01\ 002\ 0003\ 00004\ 000005\ 0000006\ 00000007\ 000000008\ldots$
is not simply normal to base 10.

Not normal

$0.01\ 002\ 0003\ 00004\ 000005\ 0000006\ 00000007\ 000000008\ldots$
is not simply normal to base 10.

$0.0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ldots$
is simply normal to base 10, but not simply normal to base 100.

Not normal

$0.01\ 002\ 0003\ 00004\ 000005\ 0000006\ 00000007\ 000000008\ldots$
is not simply normal to base 10.

$0.0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ldots$
is simply normal to base 10, but not simply normal to base 100.

The numbers is the middle third Cantor set are not simply normal to base 3 (their expansions lack the digit 1).

Not normal

$0.01\ 002\ 0003\ 00004\ 000005\ 0000006\ 00000007\ 000000008\ldots$
is not simply normal to base 10.

$0.0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ldots$
is simply normal to base 10, but not simply normal to base 100.

The numbers is the middle third Cantor set are not simply normal to base 3 (their expansions lack the digit 1).

The rational numbers are not normal to any base.

Not normal

$0.01\ 002\ 0003\ 00004\ 000005\ 0000006\ 00000007\ 000000008\ldots$
is not simply normal to base 10.

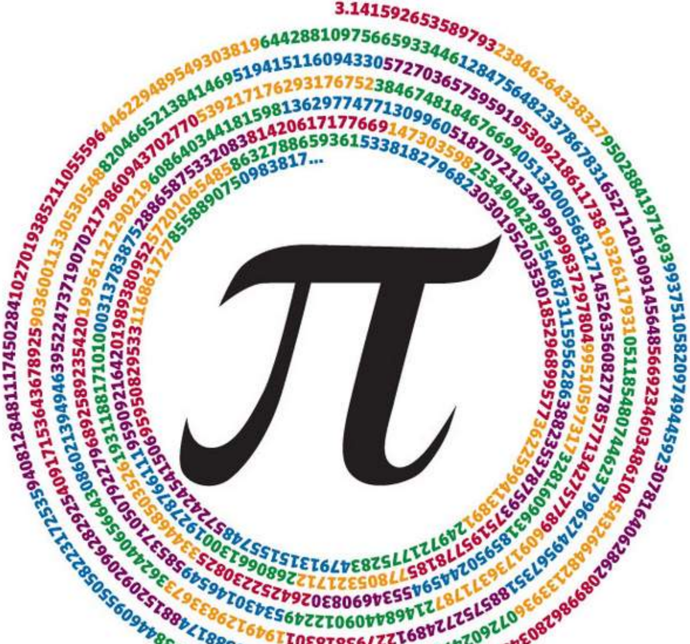
$0.0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ \ 0123456789\ldots$
is simply normal to base 10, but not simply normal to base 100.

The numbers is the middle third Cantor set are not simply normal to base 3 (their expansions lack the digit 1).

The rational numbers are not normal to any base.

Liouville's constant $\sum_{n\geq 1} 10^{-n!}$ is not normal to base 10.

Es π simplemente normal?



Examples of normal numbers?

Theorem (Borel 1909)

Almost all real numbers are absolutely normal.

Problem (Borel 1909)

Give one example of an absolutely normal number.

Examples of normal numbers?

Theorem (Borel 1909)

Almost all real numbers are absolutely normal.

Problem (Borel 1909)

Give one example of an absolutely normal number.

Are the usual mathematical constants, such as π, e, or $\sqrt{2}$, absolutely normal? Or at least simply normal to some base?

Examples of normal numbers?

Theorem (Borel 1909)

Almost all real numbers are absolutely normal.

Problem (Borel 1909)

Give one example of an absolutely normal number.

Are the usual mathematical constants, such as π, e, or $\sqrt{2}$, absolutely normal? Or at least simply normal to some base?

Conjecture (Borel 1950)

Irrational algebraic numbers are absolutely normal.

Normal to a given base

Theorem (Champernowne, 1933)

$0.123456789101112131415161718192021\ldots$ *is normal to base* 10.

It is unknown if it is normal to bases that are not powers of 10.

Besicovitch 1935; Copeland and Erdös 1946; Levin 1999;. . . Ugalde 2000; Alvarez, Becher, Ferrari and Yuhjtman 2016.

Absolutely normal

Sierpinski 1917, Lebesgue 1917; Turing 1937; Schmidt 1961; M. Levin 1970; . . . Lutz and Mayordomo 2013; Figueira and Nies 2013, 2020.

Theorem (Becher, Heiber and Slaman, 2013)

There is an algorithm that computes an absolutely normal number with just above quadratic time-complexity.

Absolutely normal

Sierpinski 1917, Lebesgue 1917; Turing 1937; Schmidt 1961; M. Levin 1970; ... Lutz and Mayordomo 2013; Figueira and Nies 2013, 2020.

Theorem (Becher, Heiber and Slaman, 2013)

There is an algorithm that computes an absolutely normal number with just above quadratic time-complexity.

$0.4031290542003809132371428380827059102765116777624189775110896366\ldots$

Normal to some bases and not to others

Theorem (Cassels 1959; Schmidt 1961)

Almost all numbers in the Cantor ternary set are normal to base 2.

Normal to some bases and not to others

Theorem (Cassels 1959; Schmidt 1961)

Almost all numbers in the Cantor ternary set are normal to base 2.

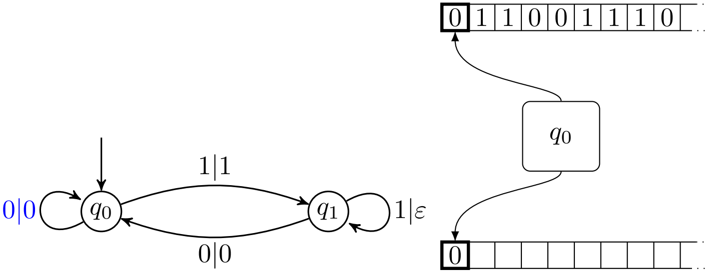
Theorem (Bailey and Borwein 2012)

Stoneham number $\alpha_{2,3} = \sum_{k \geq 1} \frac{1}{3^k \; 2^{3^k}}$ is normal to base 2 but not simply normal to base 6.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

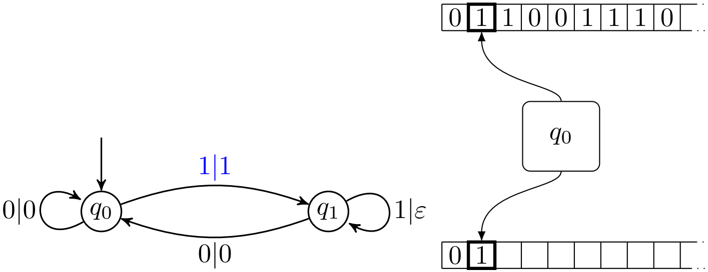
Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.



Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.

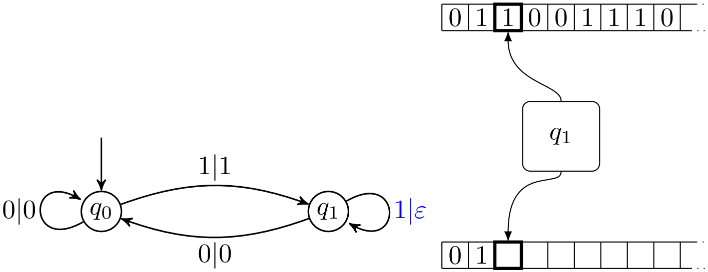


The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.

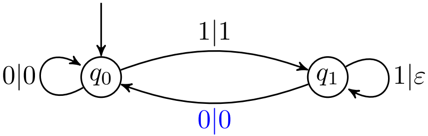


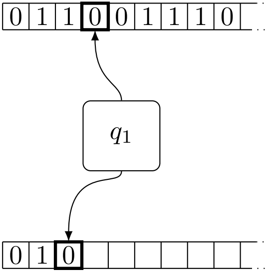
The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.



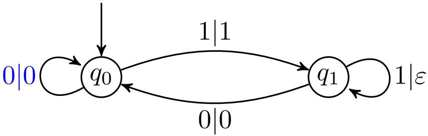


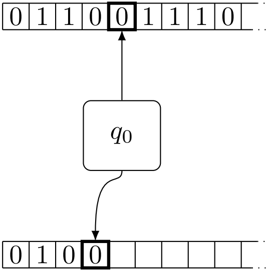
The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.



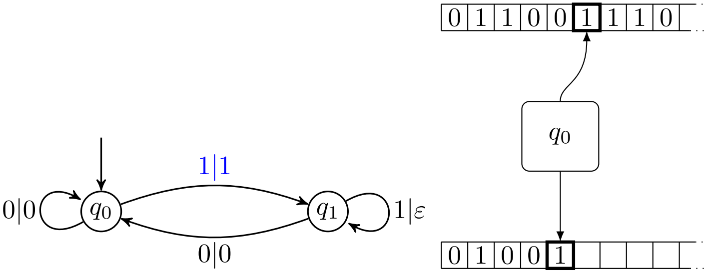


The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.

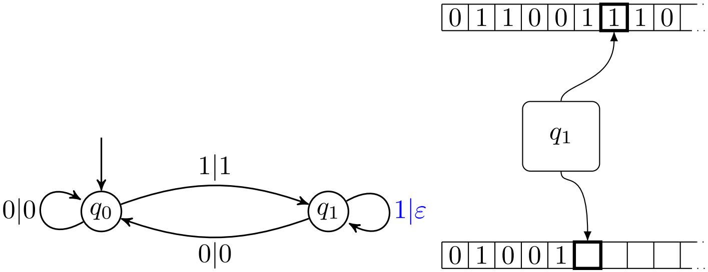


The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.

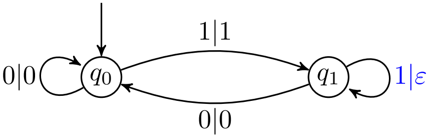


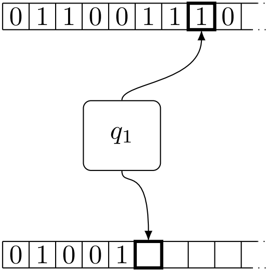
The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.



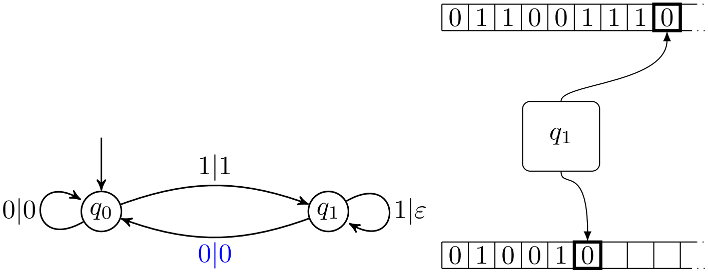


The transducer transforms rows of 1s into a single 1.

Normality and finite automata

A deterministic finite transducer T is defined by $\langle Q, A, \delta, q_0 \rangle$ where A is the alphabet, Q is a finite set of states with q_0 the starting state, and $\delta : Q \times A \to A^* \times Q$ is a transition function.
Every infinite run is accepting (Büchi acceptance condition).

Running T with input $a_1 a_2 a_3 \ldots$ gives $T(a_1 a_2 a_3 \ldots)$.



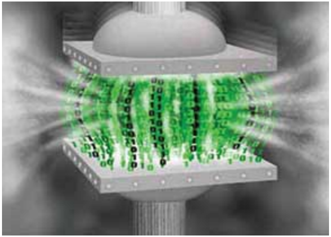
The transducer transforms rows of 1s into a single 1.

Normality and finite automata

Consider transducer $T = \langle Q, A, \delta, q_0 \rangle$. If $\delta(p, a) = \langle v, q \rangle$ write $p \xrightarrow{a|v} q$.

Definition

A sequence $x = a_1 a_2 a_3 \cdots$ is compressible by a finite transducer T if and only if the run in T $q_0 \xrightarrow{a_1|v_1} q_1 \xrightarrow{a_2|v_2} q_2 \xrightarrow{a_3|v_3} q_3 \cdots$ satisfies $\liminf_{n \to \infty} \frac{|v_1 v_2 \cdots v_n|}{n} < 1$.



Recall that the a's are symbols and the v's are words, possibly empty.

Normality and finite automata

Theorem (Schnorr, Stimm 1971; Dai, Lathrop, Lutz, Mayordomo 2004)

A sequence is normal if and only if it is incompressible by every one-to-one finite transducer .

Huffman 1959 calls them lossless compressors. A direct proof in Becher and Heiber, 2012.

Normality and finite automata

Theorem (Schnorr, Stimm 1971; Dai, Lathrop, Lutz, Mayordomo 2004)

A sequence is normal if and only if it is incompressible by every one-to-one finite transducer .

Huffman 1959 calls them lossless compressors. A direct proof in Becher and Heiber, 2012.

Theorem (Becher, Carton, Heiber 2013)

Non-deterministic one-to-one finite transducers, even if augmented with a counter, can not compress normal sequences.

Normality and pushdown automata

Question

Can deterministic pushdown transducers compress normal infinite sequences?

Normality and pushdown automata

Question

Can deterministic pushdown transducers compress normal infinite sequences?

Theorem (Boasson 2012)

Non-deterministic puhdown transducers can compress some normal sequences.

0123456789 9876543210 00 01 02 03 ...98 99 99 98 97...03 02 01 00 000 001 002...

Theorem (Carton and Perifel 2022)

Deterministic puhdown transducers can compress some normal sequences.

Sobre secuencias aleatorias

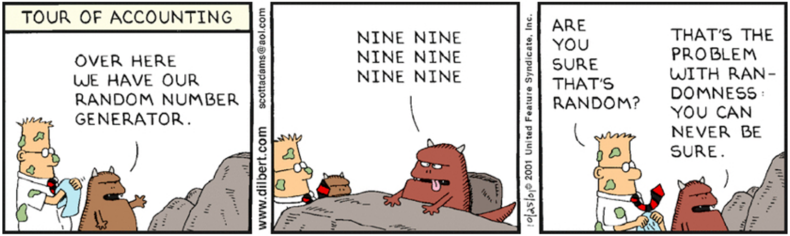
1. ¿Las secuencias puramente aleatorias son normales?
   Sí
2. ¿Las secuencias aleatorias tienen rachas del mismo símbolo?
   Sí, pero no hay mal que dure mil años.
3. ¿Si un número es puramente aleatorio en base 2, lo es en base 10?
   Sí

Randomness ☠ Computers

Random number generators (pseudo randomness)
USA National Institute of Standards and Technology
http://csrc.nist.gov/groups/ST/toolkit/rng/

http://www.random.org/

Test U01 (Pierre L'Ecuyer)
http://simul.iro.umontreal.ca/testu01/tu01.html



Actuales Investigadores
Verónica Becher, UBA CONICET - SINFIN
Martín Mereb, UBA CONICET
Nicolás Alvarez, UNS - SINFIN
Olivier Carton, Université de Paris - SINFIN
Serge Grigorieff, Université de Paris - SINFIN

Actuales Tesis y Becas
Ignacio Mollo Cunningham
Ivo Pajor
Carlos Soto
Gabriel Sac Himelfarb
Tomás Tropea
Agustín Marchionna
Darío Ocles