Normal numbers and automatic complexity

alexander.shen@lirmm.fr, www.lirmm.fr/~ashen

LIRMM CNRS & University of Montpellier

Normal numbers

Normal numbers

- $.00100111010111\ldots$

Normal numbers

- $.00100111010111\ldots$
- $\#_0(n) =$ number of 0 among the first n bits

Normal numbers

- .00100111010111...
- $\#_0(n) =$ number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$

Normal numbers

- $.00100111010111\ldots$
- $\#_0(n) =$ number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$
- $\#_{00}(n) =$ number of occurences of 00 in the first n positions

Normal numbers

- $.00100111010111\ldots$
- $\#_0(n) =$ number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$
- $\#_{00}(n) =$ number of occurences of 00 in the first n positions
- $\#_{00}(n) + \#_{01}(n) + \#_{10}(n) + \#_{11}(n) = n$

Normal numbers

- .00100111010111...
- $\#_0(n)$ = number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$
- $\#_{00}(n)$ = number of occurences of 00 in the first n positions
- $\#_{00}(n) + \#_{01}(n) + \#_{10}(n) + \#_{11}(n) = n$
- normal: $\#_{00}(n)/n \to 1/4$ and the same for all other bit strings

Normal numbers

- .00100111010111...
- $\#_0(n)$ = number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$
- $\#_{00}(n)$ = number of occurences of 00 in the first n positions
- $\#_{00}(n) + \#_{01}(n) + \#_{10}(n) + \#_{11}(n) = n$
- normal: $\#_{00}(n)/n \to 1/4$ and the same for all other bit strings
- Borel introduced as expected property of random numbers

Normal numbers

- $.00100111010111\ldots$
- $\#_0(n)$ = number of 0 among the first n bits
- simply normal: $\#_0(n)/n \to 1/2$, $\#_1(n) \to 1/2$
- $\#_{00}(n)$ = number of occurences of 00 in the first n positions
- $\#_{00}(n) + \#_{01}(n) + \#_{10}(n) + \#_{11}(n) = n$
- normal: $\#_{00}(n)/n \to 1/4$ and the same for all other bit strings
- Borel introduced as expected property of random numbers
- Another approach: cut the sequence into k-bit blocks and count the number of blocks of each type (aligned occurrences)

Basic questions about normal numbers

Basic questions about normal numbers

- Do normal number exist?

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal?

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base?

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base? (Yes)

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base? (Yes)
- Do absolutely normal numbers (all bases) exist?

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base? (Yes)
- Do absolutely normal numbers (all bases) exist? (Yes, almost all)

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base? (Yes)
- Do absolutely normal numbers (all bases) exist? (Yes, almost all)
- Are two approaches equivalent?

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: $0\,1\,10\,11\,100\,101\,110\,111\,1000\,\ldots$
- Does it depend on the base? (Yes)
- Do absolutely normal numbers (all bases) exist? (Yes, almost all)
- Are two approaches equivalent? (Yes)

Basic questions about normal numbers

- Do normal number exist? (Yes, almost all numbers are random)
- Examples? $e, \pi, \sqrt{2}$ normal? (May be yes, nobody knows)
- Champernowne number: 0 1 10 11 100 101 110 111 1000 ...
- Does it depend on the base? (Yes)
- Do absolutely normal numbers (all bases) exist? (Yes, almost all)
- Are two approaches equivalent? (Yes)
- Wall's theorem: α is normal, n integer $\Rightarrow n\alpha, \alpha/n$ are normal

Randomness as incompressibility

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea
- Kolmogorov, Levin, Chaitin,...: randomness = incompressibility

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea
- Kolmogorov, Levin, Chaitin,...: randomness = incompressibility
- 000...000 not random: short description: "million zeros"

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea
- Kolmogorov, Levin, Chaitin, ...: randomness = incompressibility
- 000...000 not random: short description: "million zeros"
- What is "description"? Different answers possible

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea
- Kolmogorov, Levin, Chaitin,...: randomness = incompressibility
- $000\ldots000$ not random: short description: "million zeros"
- What is "description"? Different answers possible
- Normality = weak randomness

Randomness as incompressibility

- Individual random sequences: plausible as outcomes of coin tossing
- (Classical) probability theory: no idea
- Kolmogorov, Levin, Chaitin,...: randomness = incompressibility
- $000\ldots000$ not random: short description: “million zeros”
- What is “description”? Different answers possible
- Normality = weak randomness
- Limited class of descriptions: finite memory

Compressibility

Compressibility

- Binary relation $D(p,x)$ on strings: "p is a description of x"

Compressibility

- Binary relation $D(p,x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| \colon D(p,x)\}$

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p, x) : p = x$, then $C_D(x) = |x|$

Compressibility

- Binary relation $D(p,x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p,x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p,x) : p = x$, then $C_D(x) = |x|$
- $D(p,x)$: x is p where each bit doubled

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p, x) : p = x$, then $C_D(x) = |x|$
- $D(p, x)$: x is p where each bit doubled
- then $C_D(x)$: $|x|/2$ for good x (with doubled bits), and $+\infty$ for bad

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p|: D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p, x) : p = x$, then $C_D(x) = |x|$
- $D(p, x)$: x is p where each bit doubled
- then $C_D(x)$: $|x|/2$ for good x (with doubled bits), and $+\infty$ for bad
- restriction: D is function: every description describes only one object

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p, x) : p = x$, then $C_D(x) = |x|$
- $D(p, x)$: x is p where each bit doubled
- then $C_D(x)$: $|x|/2$ for good x (with doubled bits), and $+\infty$ for bad
- restriction: D is function: every description describes only one object
- technical: weaker restriction: $O(1)$-valued function; only $O(1)$ objects for each description

Compressibility

- Binary relation $D(p, x)$ on strings: "p is a description of x"
- $C_D(x) = \min\{|p| : D(p, x)\}$
- trivial: Λ is a description of everything, $C_D(x) = 0$
- $D(p, x) : p = x$, then $C_D(x) = |x|$
- $D(p, x)$: x is p where each bit doubled
- then $C_D(x)$: $|x|/2$ for good x (with doubled bits), and $+\infty$ for bad
- restriction: D is function: every description describes only one object
- technical: weaker restriction: $O(1)$-valued function; only $O(1)$ objects for each description
- only finite-memory (automatic) relations allowed as D

Finite automata

Finite automata

- Idea: $D(p,x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory

Finite automata

- Idea: $D(p, x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic

Finite automata

- Idea: $D(p,x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u,v), (u,ε), (ε,u), $(\varepsilon,\varepsilon)$

Finite automata

- Idea: $D(p,x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u,v), (u,ε), (ε,u), $(\varepsilon,\varepsilon)$
- path $\Rightarrow$ pair of strings

Finite automata

- Idea: $D(p, x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u, v), (u, ε), (ε, u), $(\varepsilon, \varepsilon)$
- path $\Rightarrow$ pair of strings
- $D =$ the set of all pairs that can be read along paths

Finite automata

- Idea: $D(p,x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u,v), (u,ε), (ε,u), $(\varepsilon,\varepsilon)$
- path $\Rightarrow$ pair of strings
- $D =$ the set of all pairs that can be read along paths
- multiplication and division by an integer constant are automatic relations

Finite automata

- Idea: $D(p, x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u, v), (u, ε), (ε, u), $(\varepsilon, \varepsilon)$
- path $\Rightarrow$ pair of strings
- D = the set of all pairs that can be read along paths
- multiplication and division by an integer constant are automatic relations
- union of two automatic relations is automatic

Finite automata

- Idea: $D(p, x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u, v), (u, ε), (ε, u), $(\varepsilon, \varepsilon)$
- path $\Rightarrow$ pair of strings
- $D =$ the set of all pairs that can be read along paths
- multiplication and division by an integer constant are automatic relations
- union of two automatic relations is automatic
- composition of two automatic relations is automatic

Finite automata

- Idea: $D(p, x)$ is automatic if it can be checked reading p and x bit by bit, with finite memory
- our examples (trivial, equality, doubled bits) are all automatic
- Formal definition: graph; edges labeled by (u, v), (u, ε), (ε, u), $(\varepsilon, \varepsilon)$
- path $\Rightarrow$ pair of strings
- $D =$ the set of all pairs that can be read along paths
- multiplication and division by an integer constant are automatic relations
- union of two automatic relations is automatic
- composition of two automatic relations is automatic
- warning: no initial and final states

Optimal decompressors

Theorem (Becher, Heiber)

A sequence $x_1x_2x_3\ldots$ is normal $\Leftrightarrow$

$$\liminf C_D(x_1\ldots x_n)/n \geq 1$$

for every automatic $O(1)$-valued relation $D(p,x)$

Optimal decompressors

Theorem (Becher, Heiber)

A sequence $x_1x_2x_3\ldots$ is normal $\Leftrightarrow$

$$\liminf C_D(x_1\ldots x_n)/n \geq 1$$

for every automatic $O(1)$-valued relation $D(p,x)$

A counterpart in algorithmic information theory:
A sequence $x_1x_2x_3\ldots$ has effective dimension 1 $\Leftrightarrow$

$$\lim C_D(x_1\ldots x_n)/n \geq 1$$

for every computably enumerable $O(1)$-valued relation $D(p,x)$

Optimal decompressors

Theorem (Becher, Heiber)

A sequence $x_1x_2x_3\ldots$ is normal $\Leftrightarrow$

$$\liminf C_D(x_1\ldots x_n)/n \geq 1$$

for every automatic $O(1)$-valued relation $D(p,x)$

A counterpart in algorithmic information theory:
A sequence $x_1x_2x_3\ldots$ has effective dimension 1 $\Leftrightarrow$

$$\lim C_D(x_1\ldots x_n)/n \geq 1$$

for every computably enumerable $O(1)$-valued relation $D(p,x)$
One can use computable functions as decompressors instead of $O(1)$-relations; dimension 1 is weaker than Martin-Löf randomness

Non-normal sequences are compressible

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller
- H is a lower bound for decodable codes

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller
- H is a lower bound for decodable codes
- $H + 1$ can be achieved

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller
- H is a lower bound for decodable codes
- $H + 1$ can be achieved
- block codes are needed

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller
- H is a lower bound for decodable codes
- $H + 1$ can be achieved
- block codes are needed
- block coding uses finite memory

Non-normal sequences are compressible

- String over $\{a_1, \ldots, a_k\}$, probabilities $p_1, \ldots, p_k$.
- Different probabilities $\Rightarrow$ more efficient coding
- Morse code: frequent letters have shorter codes
- Minimal code length: $H = \sum p_i \log(1/p_i)$ (Shannon entropy)
- $\log k$ for equiprobable letters, otherwise smaller
- H is a lower bound for decodable codes
- $H + 1$ can be achieved
- block codes are needed
- block coding uses finite memory
- Technical: select a subsequence that has limit frequencies; use these frequencies for block coding, use convexity of entropy function

Normal sequences are not compressible

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$

Normal sequences are not compressible

- Normal sequence $x_1 x_2 \ldots$
- Some automatic $O(1)$ relation D

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1X_2\ldots X_M$

Normal sequences are not compressible

- Normal sequence $x_1 x_2 \ldots$
- Some automatic $O(1)$ relation D
- Why $x_1 x_2 \ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1 X_2 \ldots X_M$
- description p can be also split into corresponding blocks

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1X_2\ldots X_M$
- description p can be also split into corresponding blocks
- so $C_D(xy) \geq C_D(x) + C_D(y)$, the main feature of automatic compression

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1X_2\ldots X_M$
- description p can be also split into corresponding blocks
- so $C_D(xy) \geq C_D(x) + C_D(y)$, the main feature of automatic compression
- all k-bit strings appear equally often among $X_1, X_2, \ldots, X_M$

Normal sequences are not compressible

- Normal sequence $x_1 x_2 \ldots$
- Some automatic $O(1)$ relation D
- Why $x_1 x_2 \ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1 X_2 \ldots X_M$
- description p can be also split into corresponding blocks
- so $C_D(xy) \geq C_D(x) + C_D(y)$, the main feature of automatic compression
- all k-bit strings appear equally often among $X_1, X_2, \ldots, X_M$
- most of k-bit strings are incompressible

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1X_2\ldots X_M$
- description p can be also split into corresponding blocks
- so $C_D(xy) \geq C_D(x) + C_D(y)$, the main feature of automatic compression
- all k-bit strings appear equally often among $X_1, X_2, \ldots, X_M$
- most of k-bit strings are incompressible
- so the economy is negligible compared to length

Normal sequences are not compressible

- Normal sequence $x_1x_2\ldots$
- Some automatic $O(1)$ relation D
- Why $x_1x_2\ldots x_N$ is not compressible?
- Split it into k-bit blocks $X_1X_2\ldots X_M$
- description p can be also split into corresponding blocks
- so $C_D(xy)\ge C_D(x)+C_D(y)$, the main feature of automatic compression
- all k-bit strings appear equally often among $X_1,X_2,\ldots,X_M$
- most of k-bit strings are incompressible
- so the economy is negligible compared to length
- almost the same works for non-aligned definition of normality, since the frequencies of compressible blocks are only k times bigger

Hall's theorem

Hall's theorem

- α is normal, *n* integer $\Rightarrow$ $n\alpha$ and α/n are normal

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations
- Composition of automatic relations is automatic

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations
- Composition of automatic relations is automatic
- So if D compresses α, then $D \circ [\times N]$ compresses $N\alpha$.

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations
- Composition of automatic relations is automatic
- So if D compresses α, then $D \circ [\times N]$ compresses $N\alpha$.
- The same for division

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations
- Composition of automatic relations is automatic
- So if D compresses α, then $D \circ [\times N]$ compresses $N\alpha$.
- The same for division
- ...or for adding rational numbers

Details:

Hall's theorem

- α is normal, n integer $\Rightarrow$ $n\alpha$ and α/n are normal
- Multiplication and division by a constant are $O(1)$-valued automatic relations
- Composition of automatic relations is automatic
- So if D compresses α, then $D \circ [\times N]$ compresses $N\alpha$.
- The same for division
- ...or for adding rational numbers

Details: `https://arxiv.org/pdf/1701.09060.pdf`