Randomness and quantum computation

Computability and the BSS model,
Hiddensee, August 9 , 2016

André Nies
University of Auckland

00100111000101111010101000010101101111011000010111101010
10010101100011111010110001100111111101100000111001111000
00110011011110100011110100011100101011011001011100010110
011001100011110000100110010111011100100101000001110001111
11100100011000101111110100010111110011011100100110011010
00111111011010101101001101010110000011000001001101011100
00111000000000111000110000011101100001001100000000111011
00001000110011000100110100011100110111010101111010100111
11111011001001111101110111110000001010110011101001000100
01100001010000101010110011001100110110001101011010110001
11110010100001110001001100011101110101111100001110101000
01100011001010010010111011011000111101000111111000101111
00111001000100101101000010011110011111101100011111110110
01001001001011010001010000110100010100011100001100000100
11000111110111001000011001011010100111101111010101111111
00000001010011110010000000011011001010011010101101000010

00100100000111111011010101000100010000101101000110000100
01101001100010011000110011000101000101110000000110111000
00111001101000100101001000000100100111000001000100010100
11001111100110001110100000000100000101110111110101001100
01110110001001110011011001000100101000101001010000010000
11110011000111000110100000001001101110111101111100101010
00110011011001111001101001110100100001100011011001100000
01010110000101001101101111100100101111100010100001101110
10011111110000100110101011011010110110101010001110000100
10001011110010010000101101101010111011001100010010111100
11111101100011011110100010011000100001011101001101001100
01101111110110101101011000010111111111101011100101101101
11101000000001101011011111101101111011100011100001101011
11111011010110101000100110011111101001011010111010011111
00100100000010001011111000100101100011111111001100100100
10101000011001100101000111101100111001000101101100111101
11

Did you notice any difference between the bit sequences?

- First sequence: 896 quantum random bits
- Second sequence: the initial 896 bits of the binary expansion of $\pi - 3$

http://www.befria.nu/elias/pi/binpi.html

Compressibility

The binary expansion of π-3 can be compressed:

given n, compute the first n bits, using that

$$\pi = \sum_{i=0}^{\infty} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right)$$

The length of this description is:

number of binary digits of n + constant

(Simon Plouffe, 1995, source: Wikipedia)

Defining Randomness

Can we compress a long sequence of random bits?

NO.

For finite objects, incompressibility can be taken as a formal definition of the intuitive concept of randomness.

Random versus patterned objects

We have already seen that random objects can resemble patterned ones. Here's a musical example, courtesy of D. Hirschfeldt.

Music of Changes by John Cage (1951)

has aleatoric elements

Structures for Two Pianos by

Pierre Boulez (1961) is an example of serialism (deterministic music)

Music of Changes (Cage)

It takes random samples from *I Ching*, the “book of changes” (which the Chinese used for divination).

Serialism (Messiaen/Boulez)

Based on a 12-tone series, which determines all the other musical elements

Compressibility and information content

Objects can have low information content for two reasons:

- Highly compressible
- Highly random

A sequence of 896 zeros is highly compressible, and has no information besides the length.

A sequence of 896 quantum random bits is incompressible, and has no information besides the length.

Bennett depth

Charles Bennett (1988) introduced the notion of depth to gauge the amount of useful information in an object.

To be deep means: the longer a running time you allow, the more patterns can be discerned.

With more time, the object can be compressed further.

This fails for both random, and trivial sequences.

Examples of large depth:

Some paintings. Some Shakespeare plays. DNA

Random and structured parts: Green & Tao

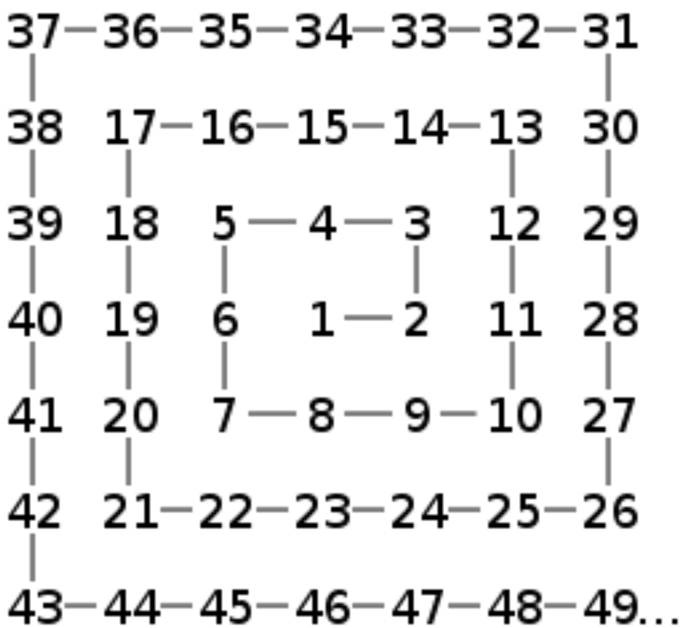
The dichotomy of random versus structured is prominent in the work of Green and Tao.

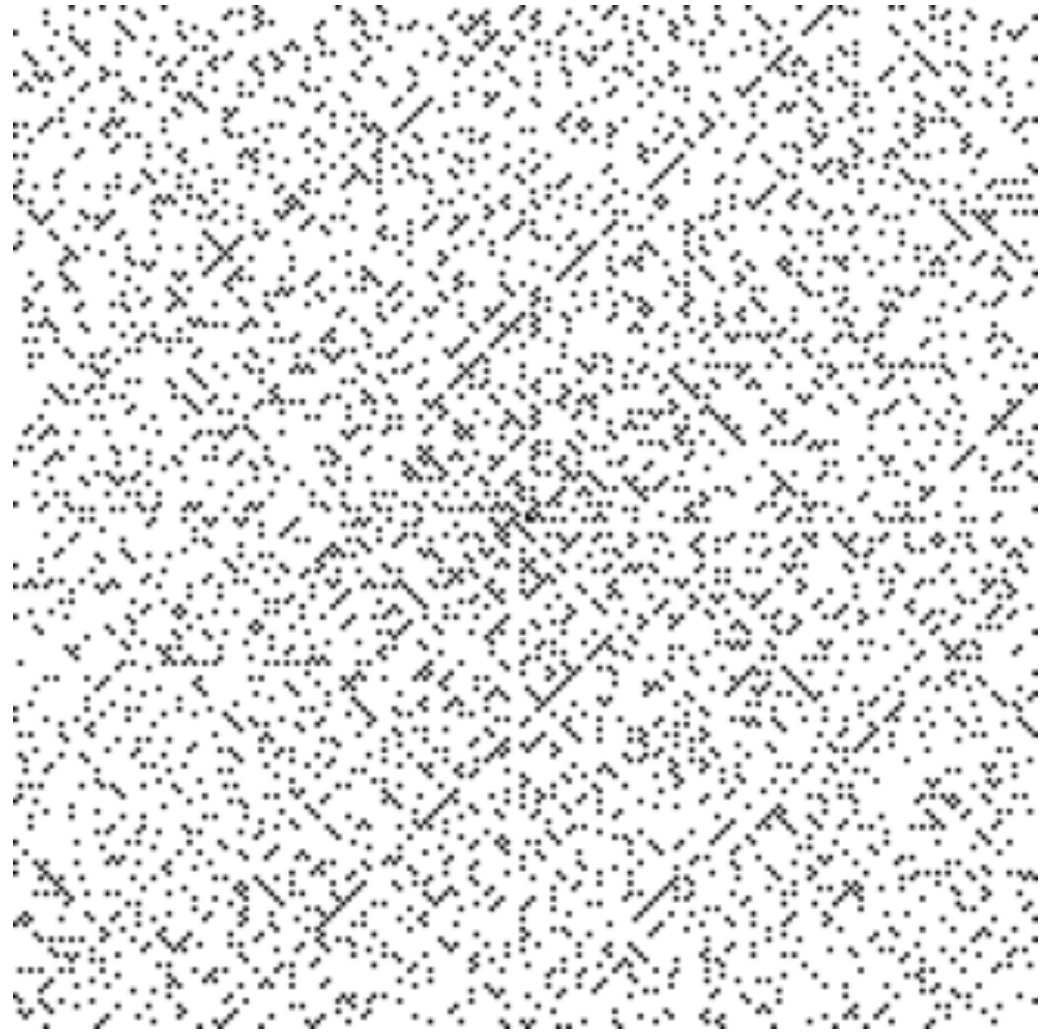
Szemeredi's theorem: every set of natural numbers with positive upper density has arbitrarily long arithmetic progressions.

Each of the known proofs proceeds by showing that the set contains a large (pseudo)random subset of a structured set (Tao, 2006 ICM).

Green and Tao (2006) used this idea to get arbitrarily long arithmetic progressions in the primes. E.g. 5, 11, 17, 23, 29

Ulam’s spiral of prime numbers





Souce: Wikipedia

Examples of compression

and of short descriptions

A compression algorithm

Many of you have used compression to save disk space. Usually this compression is based on the DEFLATE algorithm (P. Katz).

Given a long string of symbols:

- First step: create a dictionary of substrings that repeat often. In this way we don't have to write out repeated strings. (Lempel-Ziv 1977)
- Second step: Huffman (1951) encoding. Rare symbols get represented by the longer binary strings, and frequent symbols by the shorter strings.

Genome –compressible how far ?

Fruit fly: 100 million base pairs (Mbp) spread over 8 chromosomes. The "3L arm" chromosome has 24.5 Mbp. Compressible to about 1/8 using gzip.

Human: 3.3 billion base pairs (i.e. about 840 Megabytes when encoding a base pair by two bits). Compressible to about 1.1 Megabyte using DNAzip and now GenomeZip (1200 fold)

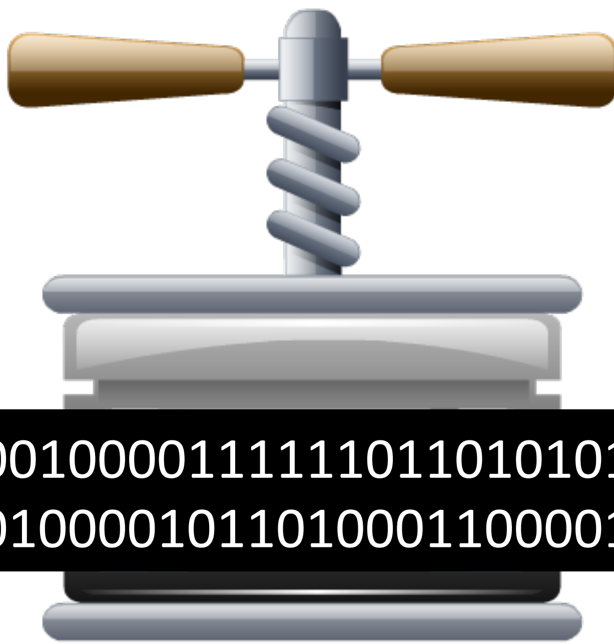
Developed at UC Irvine , 2011. Based on Huffman compression. But uses reference genome (of J. Watson) and only describes the changes.

en.wikipedia.org/wiki/Compression_of_Genomic_Re-Sequencing_Data

It's not surprising that some randomness remains: genome is product of random mutations and selection.

Compression versus description I

Compressing an object: the compressed form is of the same type as the given object. E.g., compress a bit sequence to a shorter one.



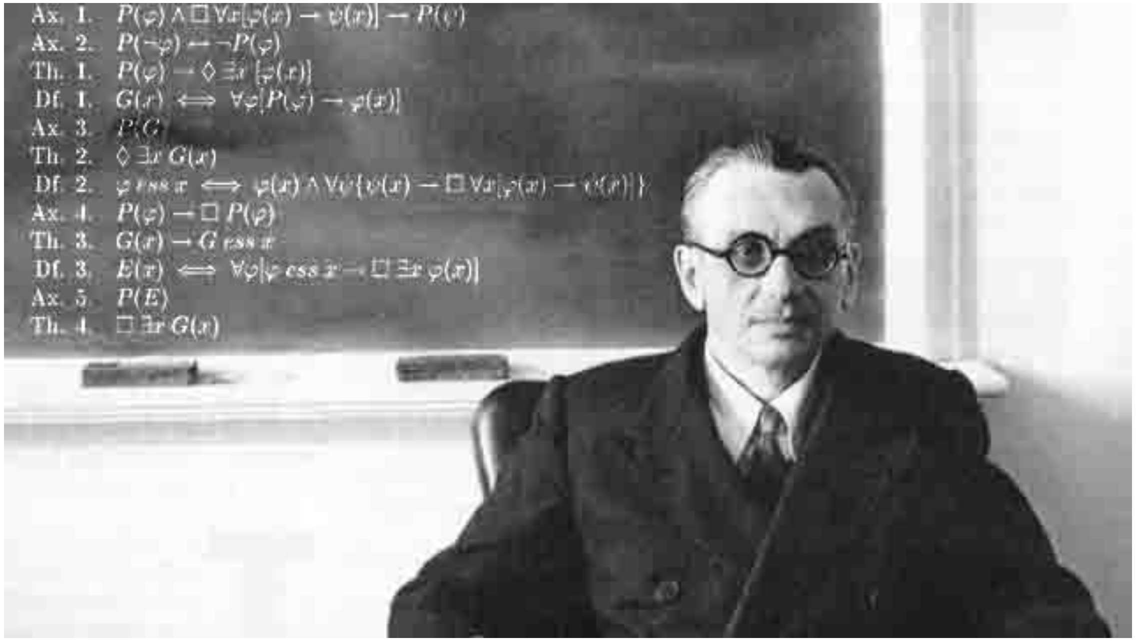
00100100001111101101010101000
10001000010110100011000001000

= 101011101

Compression versus description II

Describing an object: The description can be of a different type from the given object.

Logician's point of view:
Description is syntax.
Object is semantics.



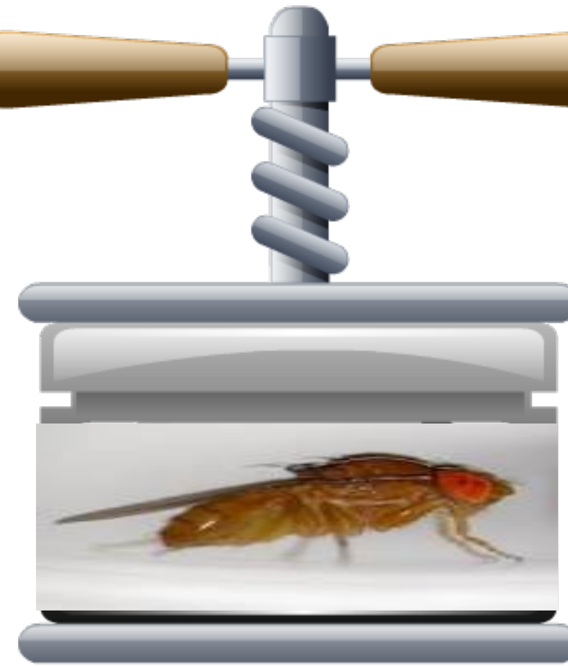
Syntax

Semantics

TAGGGAGAAATATGATCGCGTATGCGA
GAGTAGTGCCAACATATTGTGCTCTTTG
ATTTTTTGGCAACCCAAAATGGTGGCGG
ATGAACGAGATGATAATATATTCAAGTT
GCCGCTAATCAGAAATAAATTCATTGCA
ACGTTAAATACAGCACAATATATGATCG
CGTATGCGAGAGTAGTGCCAACATATTG
TGCTAATGAGTGCCTCTCGTTCTCTGTCT
TATATTACCGCAAACCCAAAAAGACAAT
ACACGACAGAGAGAGAGAGCAGCGGAG
ATATTTAGATTGCCTATTAAATATGATCG
CGTATGCGAGAGTAGTGCCAACATATTC
TGCTCTCTATATAATGACTGCCTCT ...

Initial piece of the "3L arm" chromosome of the fruit fly. http://www.fruitfly.org/sequence

Yes



No

We can only compress symbolic expressions (syntax).
First describe object, then compress the description.

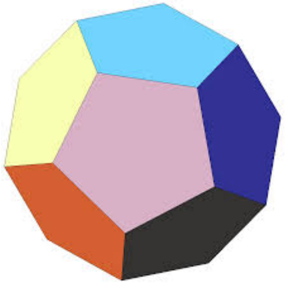
Describing finite mathematical structures

We want short descriptions in logic.

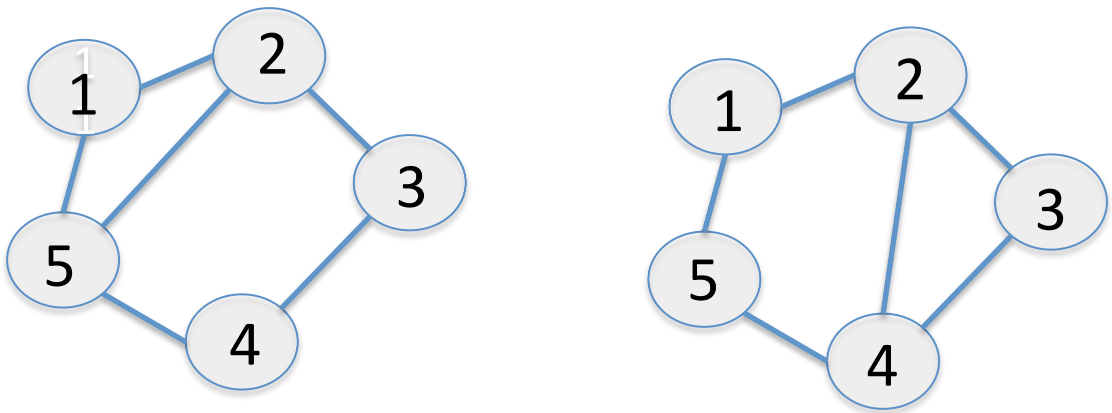
We consider two types of structures:

- Graphs are binary relationships between elements.

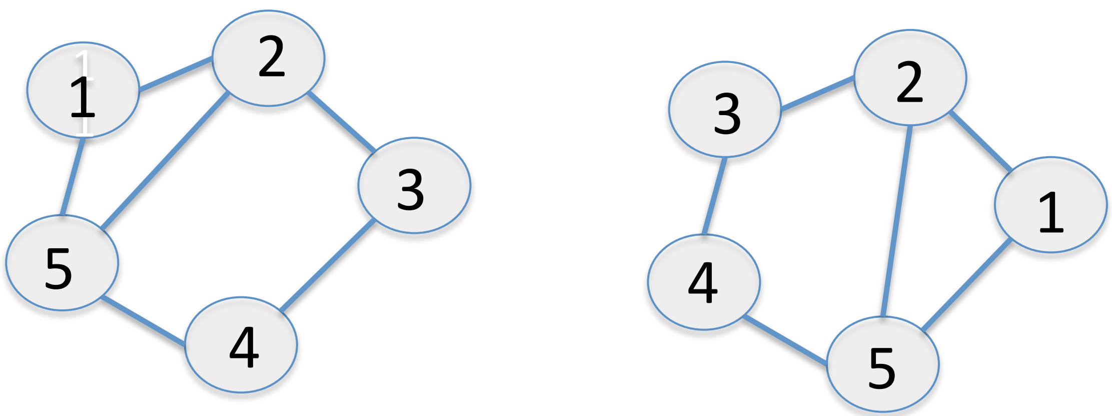
- Groups are symmetries of a set of elements. E.g. the 120 movements that fix the dodecahedron. (Group $A_5 \times Z_2$.)



Re-labeling of graphs

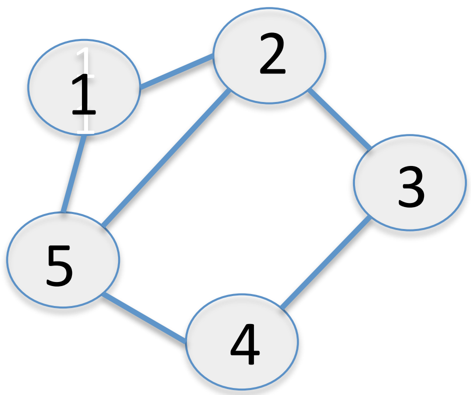


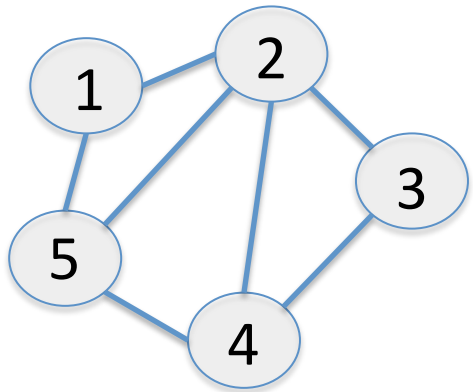
When vertices are labeled, the two graphs are different.



They can be identified after re-labeling second graph

Many non-isomorphic graphs





There are "lots" of graphs on *n* vertices that remain different even when one can re-label. This implies:

For each *n*, there is a graph on *n* vertices such that each binary description of a relabeling has length at least $\varepsilon\ (n^2 - 6 \log n)$.

The naïve description of a graph has length $n^2/2$.

Finite groups have short first-order descriptions

Last year N. and Katrin Tent showed the following, starting from some earlier work of N. with summer student Y. Maehara:

Each group of *n* elements has a description in first-order logic of size *constant* • *(log n)*^{3}.

Such a description is invariant under re-labeling of the group elements.

Example of a first-order sentence: $\forall x \exists y [y \cdot y = x]$

Gödel incompleteness (1931)

For each sufficiently strong formal system F, there is an expression that is true but unprovable. It says

"I am not provable in system F".

Paris/Harrington (1977) provided a true mathematical fact that is unprovable in the usual formal system axiomatizing arithmetic (Peano arithmetic).

Their fact is a strengthening of the finite Ramsey theorem.

Chaitin's proof of incompleteness (1969)

For a number n, consider the following true fact:
some string x is not compressible below length n.

If *n* is large compared to the size of a formal system *F*, then the fact cannot be proved in *F*.

For otherwise, "the first string *x* that *F* can prove to be incompressible below length n" yields a description of that string *x* of length log(n) + constant.

Randomness and compression

for infinite objects

What is an infinite object?

E.g. a real number: it has infinite precision.

The real number π has a finite description,

Most real numbers don't have one.

Can we compress an infinite object?

Not really.

But we can try to compress

all of its finite parts.

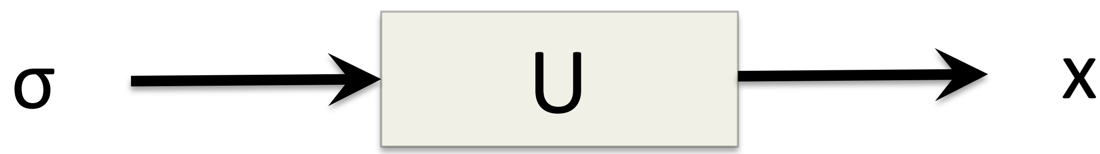
Prefix-free Kolmogorov complexity K(x)

For a finite sequence *x*, let *K(x)* denote the shortest length of a compressed form of *x*

(Solomonoff/Kolmogorov).

We use a *universal de-compressor U*.

K(x) is the length of a shortest σ such that U(σ) = x.



A technical, but important modification: if σ, τ are in the domain of U, then τ does not extend σ.

Random versus trivial

Let Z be an *infinite* bit sequence.

Let *Z*|*n* denote the first *n* bits of *Z*.

- Z is random if for some number d

  $K(Z|n) \geq n\text{-}d$ for each n.

- Z is K-trivial if for some number b,

  $K(Z|n) \leq K(n) + b$ for each n.

An infinite sequences A is Bennett deep if for each computable t, for each c, for a.e. n, $K(A|n) + c \leq K_{t(n)}(A|n)$.

Neither randoms (Bennett, 1988),
nor K-trivials (Moser/Stephan, 2014) are deep.

Far-from-random sequences

Z is K-trivial if for some number b, $K(Z|n) \leq K(n) + b$.

Musical example: Spiegel im Spiegel by Arvo Pärt.

FACT: If we can compute all the bits of Z,
then Z is K-trivial.

Solovay 1975:
some Z is K-trivial but not computable.

This Z looks as far-from-random as possible,
but is still not totally predictable.

Far from random= close to computable

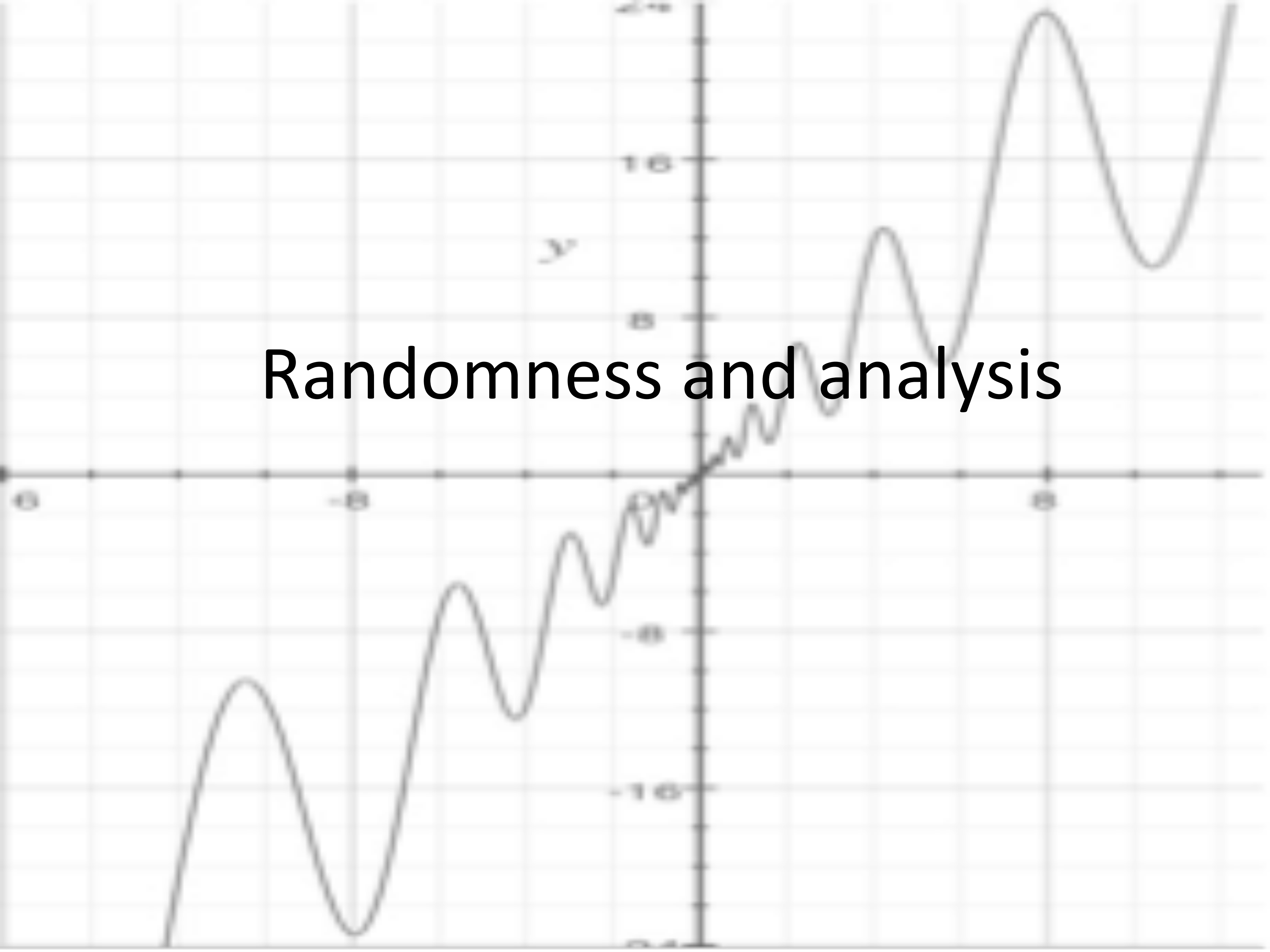
Numerous results suggest that far-from-random means that the computational power is very low.

A bit sequence Z is called low for K if, when using queries to Z as an auxiliary computational device in de-compression, we don't gain more than a constant.

N., 2005:

Z is K-trivial if and only if Z is low for K.

Randomness and analysis



Lebesgue's theorem

Henri Lebesgue (1904) introduced a notion of size for sets of real numbers.
This is used to express that a statement holds with probability one.
His intuition may have been that the statement holds for a "random" real.

Lebesgue, 1904:

Let f be increasing with domain [0,1].

Then $f'(z)$ exists for a real z with probability 1.

Algorithmic forms of Lebesgue's theorem I

We say that a real z is betting-random (Schnorr, 1975) if no effective betting strategy succeeds on its binary expansion.

The strategy always bets on the value of next bit. Success means the capital is unbounded.

(This randomness notion is weaker than the one we have defined in terms of incompressible initial segments.)

Brattka, Miller, N., 2011 (Trans. AMS, 2016):

Let f be increasing and computable.

Then f'(z) exists for any betting random real z.

Algorithmic forms of Lebesgue's theorem II

We say that a real z is polynomial time betting-random if no polynomial time computable betting strategy succeeds on the real.

N., 2014 (Symp. Theoret. Aspects CS):

Let f be increasing and polynomial time computable.

Then f ’(z) exists for any polynomial time betting random real z.

Computability in Physics

Undecidability of the spectral gap (Nature 2015)