Kolmogorov Complexity

Suppose I say I tossed a coin 40 times and got:

1010101010101010101010101010101010101010

What do you say?

Suppose I say I tossed a coin 40 times and got:

1010101010101010101010101010101010101010

You don't believe me

Suppose I say I tossed a coin 40 times and got:

11101010101001010100111001010010111100010

What do you say?

Suppose I say I tossed a coin 40 times and got:

1010101010101010101010101010101010101010

You don't believe me

Suppose I say I tossed a coin 40 times and got:

11101010101001010100111001010010111100010

Maybe

Why? What is the probability of the two strings?

Suppose I say I tossed a coin 40 times and got:

Pr[1010101010101010101010101010101010101010] = 2^{-40}

You don't believe me

Suppose I say I tossed a coin 40 times and got:

Pr[11101010101001010100111001010010111100010] = 2^{-40}

Maybe

Why? The two strings have the same probability!

Classical probability theory does not capture intuitive notion of "random"

Observation:

1010101010101010101010101010101010101010

can be programmed as

Classical probability theory does not capture intuitive notion of “random”

Observation:

1010101010101010101010101010101010101010

can be programmed as

“Repeat `10' 20 times”

Program length much shorter than string.
String is compressible

This seems impossible for

11101010101001010100111001010010111100010

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where x,y $\in$ {0,1}* ?

Can't just concatenate bits: ?

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where x,y $\in$ {0,1}* ?

Can't just concatenate bits: you wouldn't know when x ends.

One solution: ?

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where x,y $\in$ {0,1}* ?

Can't just concatenate bits: you wouldn't know when x ends.

One solution: write each bit of x twice, use “01” as delimiter

|(x,y)| = ?

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where $x,y \in \{0,1\}^*$?

Can't just concatenate bits: you wouldn't know when x ends.

One solution: write each bit of x twice, use “01” as delimiter
$|(x,y)| = 2|x| + |y| + 2$

Better: ?

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where x,y $\in$ {0,1}* ?

Can't just concatenate bits: you wouldn't know when x ends.

One solution: write each bit of x twice, use “01” as delimiter
$|(x,y)| = 2|x| + |y| + 2$

Better: write the length of x in binary, then x, then y
$|(x,y)| = ?$

We are going to make a formal definition.

We need to calculate program lengths somewhat precisely

How do you represent a pair (x,y), where $x,y \in \{0,1\}^*$?

Can't just concatenate bits: you wouldn't know when x ends.

One solution: write each bit of x twice, use “01” as delimiter

$|(x,y)| = 2|x| + |y| + 2$

Better: write the length of x in binary, then x, then y

$|(x,y)| = 2 \text{ floor}(\log |x| + 1) + |x| + |y| + 2$

$\leq 2\log |x| + |x| + |y| + 4$

Exercise: do better

Definition:
The Kolmogorov complexity of x, denoted K(x), is the minimum length of a pair (M,y) such that TM M on input y outputs x.

Fact: $\exists$ c : $\forall$ x : $K(x) \leq |x| + c$
Proof:

Definition:
The Kolmogorov complexity of x, denoted K(x), is the minimum length of a pair (M,y) such that
TM M on input y outputs x.

Fact: $\exists$ c : $\forall$ x : $K(x) \leq |x| + c$
Proof: Define M := “On input y, output y.” $|(M,x)| \leq |M| + |x|$

Fact: $\exists$ c : $\forall$ x : $K(xx) \leq K(x) + c \leq |x| + c$
Proof:

Definition:

The Kolmogorov complexity of x, denoted K(x), is the minimum length of a pair (M,y) such that TM M on input y outputs x.

Fact: $\exists$ c : $\forall$ x : K(x) ≤ |x| + c

Proof: Define M := "On input y, output y." |(M,x)| ≤ |M| + |x|

Fact: $\exists$ c : $\forall$ x : K(xx) ≤ K(x) + c ≤ |x| + c

Proof: Let (M,y) be a shortest pair such that M(y) = x.
Consider M' that on input (M,y) runs M(y) to get x, and then makes two copies of x.
So M'((M,y)) = xx, and |(M',(M,y))| ≤ 2|M'| + |(M,y)| ≤ K(x) + c.

Exercise: $\exists$ c $\forall$ x : K(x^2 + 17) ≤ K(x) + c

Fact: $\exists$ c : $\forall$ x,y : $K(xy) \leq 2 \log(K(x)) + K(x) + K(y) + c$

Proof:

Let $M_x(x') = x$ where $|(M_x, x')| = K(x)$

$M_y(y') = y$ where $|(M_y, y')| = K(y)$

Consider M that first runs $M_x(x')$ then $M_y(y')$

$|(M, ((M_x, x'), (M_y, y')))| =$

$= 2|M| + |((M_x, x'), (M_y, y'))|$

$= 2|M| + 2\log(K(x)) + K(x) + K(y)$,

using pairing (.,.) that we discussed

Exercise: $\forall$ c $\exists$ x,y : $K(xy) \geq K(x) + K(y) + c$

Definition: A string x is incompressible if $K(x) \geq |x|$.

Fact:
$\forall$ n there are incompressible strings of length n.

Proof:

Definition: A string x is incompressible if $K(x) \geq |x|$.

Fact:
$\forall$ n there are incompressible strings of length n.

Proof: JUST COUNT
The number of descriptions (M,x) of length < n is at most

$2^0 + 2^1 + 2^2 + \ldots + 2^{n-1} < 2^n$ = number of length-n strings

Exercise:
- The set of incompressible strings is undecidable

Exercise:
- K(x) is not computable