COMPSCI 501: Formal Language Theory

Lecture 20: Descriptive Complexity

Marius Minea
marius@cs.umass.edu

University of Massachusetts Amherst

8 March 2019

Questions for Today

- What is information?
- Is there an optimal encoding?
- Are there incompressible strings ?
- Can we compute the complexity of a string?

Defining Information Quantity

011011011011011011011

011010011001011010010

String 1 is clearly a repetition, 7 times 011

String 2, less apparent

- Looking for precise, unambiguous description to recreate object
- Short, or shortest one if possible
- Representation rules
  - Consider only objects that are bitstrings
  - Consider only descriptions that are bitstrings

Representations using Turing Machines

- Option 1: no input

1. Construct Turing Machine that that prints string when starting with *blank tape*
2. Encode Turing machine itself

TM will contain some "table" for the string
Not very efficient

- Option 2: some input

Describe string x with TM M and input w
Intuition: w describes part that's inefficient to encode

Represent as $\langle M \rangle w$ (will write $\langle M, w \rangle$)

How to separate a concatenation ?

Double bits in representation of $\langle M \rangle$: 001100001100 for 010010
end with 01 (not doubled, can detect)

Defining Information Quantity

Def: The **minimal description** of a binary string x is the shortest string $\langle M, w \rangle$ where M halts on input w with x on tape.
if several, choose lexicographically first

The **descriptive complexity** (Kolmogorov complexity) is the length of the mininmal description: $\mathsf{K}(x) = |d(x)|$

Theorem $\exists c \forall x \,.\, K(x) \leq |x| + c$

The descriptive complexity of a string is at most a constant more than its length
constant does not depend on string

Proof idea: have the input w be the string x itself
M_{id} does nothing: halt, leave input on tape (identity function)
constant c is $|\langle M_{id} \rangle|$

Complexity and String Operations

Doubling a string should not add much to its complexity:

$\forall x \exists c \,.\, \mathsf{K}(xx) \leq \mathsf{K}(x) + c$

Let $d(x) = \langle M_1, w \rangle$. Construct M_2 that:
reads $\langle M_1, w \rangle$, runs M_1 on w, doubles string left on tape.
Then $d(xx) = \langle M_2 \rangle d(x)$. Constant is $|\langle M_2 \rangle|$.

Complexity of concatenation? Sum of complexities? **Not true**

Need to distinguish break point.

Simple idea: double-encode first string, separate (01)

$\exists c \forall x, y \,.\, \mathsf{K}(xy) \leq 2\mathsf{K}(x) + \mathsf{K}(y) + c$

Concatenation: Can we do better?

Could encode length $|d(x)|$ as binary integer and prepend.
length is doubled to be distinguishable.

$2 \log \mathsf{K}(x) + \mathsf{K}(x) + \mathsf{K}(y) + c$

Even better? Do the same length-encoding with the length:

$2 \log \log \mathsf{K}(x) + \log \mathsf{K}(x) + \mathsf{K}(x) + \mathsf{K}(y) + c$, etc.

Cannot do $\mathsf{K}(x) + \mathsf{K}(y) + c$

Optimality of Definition

Could a different definition achieve smaller complexity?
Not in an algoritic way.

A specific description method: *description language* $p : \Sigma^* \to \Sigma^*$
p: **computable function**

Minimal description $d_p(x)$: first string s with $p(s) = x$
(Think: p = programming language, s = shortest program)

Theorem: For any description language p there exists a constant c (depending only on p), so $\forall x \mathsf{K}(x) \leq \mathsf{K}_p(x) + c$

(Choice of language varies complexity only by constant amount)

Proof: p computable $\Rightarrow$ Turing machine M_p
Encoding is $\langle M_p \rangle d_p(x)$ (prepend interpreter for p)

Incompressible Strings

Def.: A string x is c-**compressible** if $\mathsf{K}(x) \leq |x| - c$.

Not c-commpressible: **incompressible by** c

incompressible = incompressible by 1.

Incompressible strings exist

Amazingly simple:

Number of strings shorter than n is $2^0 + 2^1 + \ldots 2^{n-1} < 2^n$
$\Rightarrow$ at least one n-bit string is incompressible!

Which? Can we tell? Not really.

Incompressibility and Randomness

Corollary: At least $2^n - 2^{n-c+1} + 1$ strings of length n are incompressible by c

Or: probability of picking a n-bit string with complexity $\geq n - c$ is more than $1 - \frac{1}{2^c}$

Incompressible strings have usual properties of random strings:
about equal numbers of ones and zeroes
longest run of 0s has length approx. $\log n$, etc.

Most Strings are Close to Incompressible

Theorem:
Let f be a computable property that holds for almost all strings. Then for any $b > 0$, the property is false only for finitely many strings incompressible by b.

holds for almost almost all strings = fraction of strings of length n for which f is false goes to 0 as $n \to \infty$.

Proof: Enumerate strings on which s fails, in string order:
On input i, find and output i^{th} string x where $f(x)$ is false.

This gives a short description: $\langle M, i_x \rangle$. Let $c = |\langle M \rangle|$.

Now consider $b > 0$ and length n so at most $\frac{1}{2^{b+c+1}}$ strings fail f.

Since we have $< 2^{n+1}$ strings of length $\leq n$, all indices are $< 2^{n+1}/2^{b+c+1} = 2^{n-b-c}$.
Their length is $\leq n - b - c$, so with $\langle M \rangle$, still $\leq n - b$.

So $\mathsf{K}(x) \leq n - b$: every sufficiently long string that fails f is compressible by b, so only finitely many are incompressible by b

Incompressible Strings are Undecidable

Let $U = \{x \mid \mathsf{K}(x) \geq |x|\}$ be the set of incompressible strings.

Assume we have a TM that decides U.
We know U has at least one string of each length n.

We use it to construct a TM M that on input n outputs the first n-bit string s_n from U.

By definition, $\mathsf{K}(s_n) \geq n$. But s_n can be represented by $\langle M, n \rangle$, where $|\langle M \rangle| = c$ is constant, and n takes $\log n$ bits, so $K(s_n) \leq c + \log n$.

But $n \leq c + \log n$ is true only for finitely many n, contradiction.

Nearly Incompressible Strings

Theorem: For some constant b, for every string x, the minimal description $d(x)$ is incompressible by b.

Consider a TM M which double-decodes an input:

On input $\langle R, u \rangle$, where R is a TM:
  Run R on y and reject if output not of the form $\langle S, z \rangle$
  Run S on z and halt with result on tape.

Claim: $b = |\langle M \rangle| + 1$ satisfies the theorem.

Assume we had a b-compressible description $d(x)$, thus $|d(d(x))| \leq |d(x)| - b$. But then $\langle M \rangle d\ d(x)$ is a description of x, with length $\leq (b-1) + |d(x) - b| = |d(x)| - 1$, which contradicts the definition of d as minimal.

Applications: Infinitely Many Primes

Suppose not: just k primes $p_1, p_2, \ldots p_k$

Any number described by exponents: $e_1, e_2, \ldots, e_k$.

Let m be incompressible n-bit number, so $\mathsf{K}(m) \geq n$.

Exponents give a short description: each $e_i \leq \log m$.

So $|d(e_i)| \leq \log \log m$ and
$|d((e_1, \ldots, e_k))| \leq 2k \log \log m \leq 2k \log(n+1)$, so
$\mathsf{K}(m) \leq 2k \log(n+1) + c$.

For large enough n, this cannot be $\geq n$, contradiction.

Enumerating Incompressible Strings

Theorem: Any enumerable subset of incompressible strings is finite.

Proof: Take $A = \{x \mid \mathsf{K}(x) \geq |x|\}$.

Assume it had an infinite enumerable subset $B \subseteq A$.

Define $h(n) =$ first enumerated string with length $\geq n$.

Then h is computable, and by definition of A,
$\mathsf{K}(h(n)) \geq |h(n)| \geq n$.

But at the same time, $h(n)$ is described by n, so
$\mathsf{K}(h(n)) \leq \mathsf{K}(n) + c \leq \log n + c$, contradiction,
since $n > \log n + c$ for large n.