Notes on Kolmogorov Complexity

Let $\langle M \rangle$ be a representation of a Turing machine M as a bit string, and for a bit string x let $|x|$ be its length. We say that the pair $(< M >, y)$ where M is a Turing machine and y is a bit string *represents* the bit string x if M on input y outputs x.

We define the Kolmogorov complexity $K(x)$ of a bit string x as the smallest k such that there exists a representation $(\langle M \rangle, y)$ of x such that $|(\langle M \rangle, y)| \leq k$.[1]

The Kolmogorov complexity of a string x can be see as the length of the shortest "self-extracting" compressed encoding of x, where we may think of y as an encoding of x and M as the appropriate decompression algorithm. Under this perspective, up to additive terms, the Kolmogorov complexity of a string is the length of the best possible computable compression.

Indeed, consider any computable compression algorithm C and let D be its computable inverse such that $D(C(x)) = x$ for every bit string x. Then we have that, for every x, $|C(x)| \geq K(x) - O(1)$, because for every x we can use the representation $(\langle D \rangle, C(x))$ whose length is just $|C(x)| + O(1)$.

Even though the Kolmogorov complexity of a bit string is the length of the "ultimate" compression that is applicable to the string, there still are strings whose Kolmogorov complexity is no smaller than the length of the string itself.

Fact 1 *For every n there is a string $x \in \{0,1\}^n$ such that $K(x) \geq n$.*

Proof: There are 2^n strings in $\{0,1\}^n$, but the cardinality of the set

$$\{x : K(x) \leq n-1\}$$

is at most the cardinality of the set of corresponding minimal representations, which is at most the cardinality of the set of all binary strings of length $\leq n-1$ which is

$$1 + 2 + 4 + \cdots + 2^{n-1} = 2^n - 1$$

□

In fact, if we pick at random an n bit string, there is an extremely high probability that we pick a string whose Kolmogorov complexity is very high.

Fact 2 *For every n and every c, the probability that a random n-bit string x has Kolmogorov complexity $\geq n - c$ is more than $1 - \frac{1}{2^c}$.*

Proof: We show that the probability of picking a string of Kolmogorov complexity $\leq n - c - 1$ is less than $1/2^c$. Define

$$S := \{x : K(x) \leq n - c - 1\}$$

[1] In order to properly define the length of pair of bit strings $(\langle M \rangle, y)$, we need to specify how the pair itself is represented. As in the book, we assume that $\langle M \rangle$ is encoded in a "self-delimiting" way, by representing each zero as 00 and each one as 11, and then using 01 as a marker of the end of the representation of $\langle M \rangle$; we then concatenate y to this self-delimiting representation of M. In this representation, if $|\langle M \rangle| = m$ and $|y| = n$, then $(\langle M \rangle, y)$ has length $2m + 2 + n$. There are more succint "self delimiting" encoding, but the one we just described suffices for the purposes of these notes.

Then

$$\begin{aligned} |S| &\leq |\{(\langle M\rangle, y) : |(\langle M\rangle, y)| \leq n-c-1\}| \\ &\leq |\{z : |z| \leq n-c-1\}| \\ &= 1+2+\cdots 2^{n-c-1} \\ &= 2^{n-c}-1 \end{aligned}$$

And the probability that a random n-bit string belongs to S is

$$\frac{|S|}{2^n} \leq \frac{2^{n-c}-1}{2^n} = \frac{1}{2^c} - \frac{1}{2^n} < \frac{1}{2^c}$$

□

For example, if we pick a random 128-bit string, the string has Kolmogorov complexity at least 108, except with probability less than $1/1,000,000$.

Let $R := \{x : K(x) \geq |x|\}$ be the set of incompressible strings. As we proved above, R is an infinite set that contains at least one string of each possible length.

Theorem 3 *R is not decidable.*

Proof: Suppose M is a Turing machine that decides R. Then we construct a Turing machine M' that on input the number n outputs the lexicographically first string in $\{0,1\}^n$ which belongs to R. Let $\langle n\rangle$ denote the number n written in binary, using $\lceil \log_2 n \rceil$ bits, and consider the strings of the form $s_n := (\langle M'\rangle, \langle n\rangle)$.

On the one hand, s_n, being the output of M' on input n, must be an n-bit string in R, and so

$$K(s_n) \geq n$$

on the other hand, $(\langle M'\rangle, \langle n\rangle)$ is a representation of s_n of length $\log n + c$ for some constant c, so must have

$$n \leq \log n + c$$

which is false for sufficiently large values of n. □

Notice that from the above theorem we can deduce a new proof that the Halting problem is undecidable, by showing that if the Halting problem were decidable then R would also be decidable.

We can also deduce a more constructive proof of the existence of unprovable statement in mathematics.

Fix a formalization of mathematics in which, similarly to the treatment we followed in the notes on Gödel's theorem, we assume that

- For every binary string x and integer k, we can construct a statement $S_{x,k}$ equivalent to "$K(x) \geq k$."
- If there is a valid proof P of a statement S, then S is true.
- It is decidable whether a given P is a valid proof of statement S.

Lemma 4 *For every formalization of mathematics as described above, there is a threshold value t, such that all statements of the form $S_{x,k}$ with $k > t$ are unprovable.*

PROOF: Consider the following algorithm:

- Input: k
- $m := 1$
- while (true)
  - For all strings x of length at most m
    - For all strings P of length at most m
      - If P is a valid proof of $S_{x,k}$, output x and halt
  - $m := m + 1$

Let M be the Turing machine that implements the above algorithm. If k is such that there is a string for which $S_{x,k}$ is provable, then M will find such a string and output it. In such a case we have

$$k \leq M(k) \leq \log k + c \tag{1}$$

for some constant c, because the output of M, by construction, has Kolmogorov complexity at least k, but the pair $(\langle M \rangle, \langle k \rangle)$ is a representation of it of length $\log k + O(1)$.

The inequalities in (1) can hold only for a finite number of values k. □

Remarkably, the above theorem gives us a way to easily generate statements that, with high probability, are true and unprovable. Once we fix the formalization of mathematics that we are working with, the algorithm in the proof of the above theorem is well defined, and we can explicitely compute how many bits c it takes to write it down. Then we can find a k such that $k - \log k > c$. Now no statement of the form "$K(x) \geq k$ is provable. Let us pick at random a string x of $k + 20$ bits: with probability at least 99.9999% the string x is such that $K(x) \geq k$, but such a statement is unprovable in our formalization of mathematics.