39.2 Optimal Compression, Kolmogorov Complexity

Kolmogorov Complexity

We define the **Kolmogorov complexity** of a bitstream `B` to be the shortest bitstream C_B that outputs `B`. Let the *Java-Kolmogorov complexity* $K_J(B)$ be the shortest Java program that generates `B`.

Note that for any bitstream B, $K(B)$ definitely exists. However, finding and proving $K(B)$ might be difficult or even impossible.

Languages and Complexity

An important thing to note is that Kolmogorov complexity is language-independent. To run any program in one language in another, all I have to do is write an interpreter. For example, if I want to run a Python program that is not easily translatable to Java, I could instead just write a Java interpreter to read the text of the Python program and run it. In this case, $K_J(B) \leq K_P(B) + I$, where I is the length of the interpreter (a constant value).

This highlights a very deep fact about Kolmogorov complexity: most bitstreams are fundamentally incompressible no matter which language we choose for our compression algorithm.

Consider a bitstream of 1,000,000 bits. Out of all compression algorithms possible, only 1 in $2^{4999999}$ bitstreams have a change of being compressed by more than 50% (499,999 bits or less).

Uncomputability

Another important fact regarding Kolmogorov complexity is that it is impossible to compute. A proof of this fact is provided here.

Practically, this means that it is impossible to write a "perfect" (optimal) compression algorithm, since we can't even compute the length of the shortest program!

Previous
39.1 Models of Compression

Next
39.3 Space/Time-Bounded Compression

Last updated 1 year ago