6 Randomness

6.1 Randomness and Complexity

Intuitively, a random sequence is one that has the same properties as a sequence of coin flips. But this definition leaves the question, what *are* these properties? Kolmogorov resolved these problems with a new definition of random sequences: those with no description noticeably shorter than their full length. See survey and history in [Kolmogorov, V.A.Uspenskii 87, Li, Vitanyi 19].

Kolmogorov Complexity $K_A(x|y)$ of the string x given y is the length of the shortest program p which lets algorithm A transform y into x: $\min\{(\|p\|) : A(p,y) = x\}$. There exists a Universal Algorithm U such that, $K_U(x) \leq K_A(x) + O(1)$, for every algorithm A. This constant $O(1)$ is bounded by the length of the program U needs to simulate A. We abbreviate $K_U(x|y)$ as $K(x|y)$, or $K(x)$ for empty y.

An example: For $A : x \mapsto x$, $K_A(x) = \|x\|$, so $K(x) < K_A(x) + O(1) < \|x\| + O(1)$.

Can we compute $K(x)$ by trying all programs $p, \|p\| < \|x\| + O(1)$ to find the shortest one generating x? This does not work because some programs diverge, and the halting problem is unsolvable.
In fact, no algorithm can compute K or even any its lower bounds except $O(1)$.

Consider the Berry Paradox expressed in the phrase: "The smallest integer which cannot be uniquely and clearly defined by an English phrase of less than two hundred characters." There are $< 128^{200}$ English phrases of < 200 characters. So there must be integers not expressible by such phrases and the smallest one among them. But isn't it described by the above phrase?

A similar argument proves that K is not computable. Suppose an algorithm $L(x) \neq O(1)$ computes a lower bound for $K(x)$. We can use it to compute $f(n)$ that finds x with $n < L(x) \leq K(x)$, but $K(x) < K_f(x) + O(1)$ and $K_f(f(n)) \leq \|n\|$, so $n < K(f(n)) < \|n\| + O(1) = \log O(n) \ll n$: a contradiction.
So, K and its non-constant lower bounds are not computable.

An important application of Kolmogorov Complexity measures the Mutual Information:
$I(x : y) = K(x) + K(y) - K(x, y)$. It has many uses which we cannot consider here.

Deficiency of Randomness

Some upper bounds of $K(x)$ are close in some important cases. One such case is of x generated at random. Define its ***rarity*** for uniform on $\{0,1\}^n$ distribution as $d(x) = n - K(x|n) \geq -O(1)$.

What is the probability of $d(x) > i$, for uniformly random n-bit x ? There are 2^n strings x of length n. If $d(x) > i$, then $K(x|n) < n - i$. There are $< 2^{n-i}$ programs of such length, generating $< 2^{n-i}$ strings. So, the probability of such strings is $< 2^{n-i}/2^n = 2^{-i}$ (regardless of n)! Even for $n = 1,000,000$, the probability of $d(x) > 300$ is absolutely negligible (provided x was indeed generated by fair coin flips).

Small rarity implies all other enumerable properties of random strings. Indeed, let such property "$x \notin P$" have a negligible probability and S_n be the number of n-bit strings violating P, so $s_n = \log(S_n)$ is small. To generate x, we need only the algorithm enumerating S_n and the s_n-bit position of x in that enumeration. Then the rarity $d(x) > n - (s_n + O(1))$ is large. Each x violating P will thus also violate the "small rarity" requirement. In particular, the small rarity implies unpredictability of bits of random strings: A short algorithm with high prediction rate would assure large $d(x)$. However, the randomness can only be refuted, cannot be confirmed: we saw, K and its lower bounds are not computable.

Rectification of Distributions. We rarely have a source of randomness with precisely known distribution. But there are very efficient ways to convert "roughly" random sources into perfect ones. Assume, we have such a sequence with weird unknown distribution. We only know that its long enough (m bits) segments have min-entropy $> k + i$, i.e. probability $< 1/2^{k+i}$, given all previous bits. (Without such m we would not know a segment needed to extract even one not fully predictable bit.) No relation is required between n, m, i, k, but useful are small m, i, k and huge $n = o(2^k/i)$. We can fold X into an $n \times m$ matrix. We also need a small $m \times i$ matrix Z, independent of X and **really** uniformly random (or random Toeplitz, i.e. with restriction $Z_{a+1,b+1} = Z_{a,b}$). Then the $n \times i$ product XZ has uniform with accuracy $O(\sqrt{ni/2^k})$ distribution. This follows from [Goldreich, Levin 89], which uses earlier ideas of U. and V. Vazirani.