A PROOF OF KOLMOGOROV UNCOMPUTABILITY

Theorem. *In any Turing-complete programming language, there exists no program that decides if the Kolmogorov complexity of a binary string is k, for any natural number k.*

Proof. Fix a Turing-complete programming language L. Let $\mathrm{kol}(s)$ denote the Kolmogorov complexity of binary string s in L. By way of contradiction, suppose P is such a program in L. Choose a natural number k such that $k > \log_2(k) + |P|$. Thus, $|P| < k - \log_2(k)$ [1]. Write a program Q that enumerates over all strings in L in lexicographic order, and on each string x, invokes P to decide if $\mathrm{kol}(x) = k$, outputting x if so (and resuming the enumeration otherwise). Notably, the length of Q is the sum of the length of P, the length of k, and some constant. Thus, $|Q| = |P| + \log_2(k) + c_1$, for some $c_1 \in \mathbb{N}$ [1]. Since Q enumerates over all strings in L in order of increasing length, Q will eventually obtain and output a string x' such that $\mathrm{kol}(x') = k$. This means the Kolmogorov complexity of $\mathrm{kol}(x')$ is at most the length of Q. That is, $(\mathrm{kol}(x') = k) \leq (|Q| = |P| + \log_2(k) + c_1)$. Thus, $k \leq |P| + \log_2(k) + c_1$. This implies $|P| \geq k - \log_2(k) + c_2$, for some $c_2 = -c_1$. This contradicts [1], for a sufficiently large choice of k. $\square$