Descriptive complexity

Today we will take a brief look at Information Theory, using data compression as our motivating example. The goal of a "lossless compression algorithm" is to map strings to shorter strings (which in practical terms are then cheaper to store, faster to send over the internet, etc) in such a way that the original string can be recovered later.

Throughout, assume some fixed Σ and encoding scheme $\langle \cdot \rangle$.

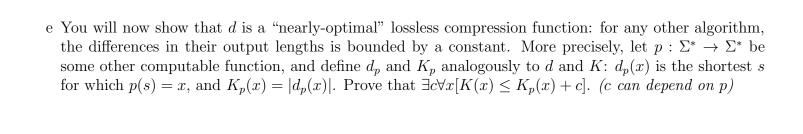
a First you will prove some fundamental limitations of lossless compression: prove that for any one-to-one function $f: \Sigma^* \to \Sigma^*$, for every n there is an "incompressible" string of length n, i.e. an x such that $|f(x)| \ge |x|$. Similarly, argue that any useful lossless compression algorithm will inevitably also map some strings to *longer* strings.

b (from Sipser Definition 6.23) A **description** of a string x is a string $\langle M, w \rangle$ where TM M on input w halts with x on its tape. The minimal description of x, written d(x), is the shortest such string. (More precisely, the first such string in shortlex order.) The **descriptive complexity** (also known as the Kolmogorov complexity) of x, written K(x), is K(x) = |d(x)|.

Provide a short description of the string $(01)^{2^{1000}}$. (It does not have to be provably minimal.)

c In part (a) we saw that any useful compression algorithm must make some strings longer. You will now prove that our "minimal descriptions" are never *much* longer than the original string: prove that $\exists c \forall x [K(x) \leq |x| + c]$.

d Intuitively, the string xx should only be slightly harder to describe than the string x. Prove that $\exists c \forall x [K(xx) \leq K(x) + c]$



f In part (e) we showed one way that d is "nearly-optimal". Come up with a few reasons why we might not actually use it everywhere for real-world compression. (There's one key reason, but try to find others as well. Don't look at part (g) until you're feeling finished with this part.)

g (extra challenge problem) Prove that K(x) is not a computable function. (Hint: suppose towards contradiction it is computable, and then use that to create a short program that outputs some string with extremely high descriptive complexity.)