CECS 419-519, Writing Assignment 4, Due 8:00 am, February 23rd, 2024, Dr. Ebert

Directions

Make sure name is on all pages. Order pages (front and back) so that solutions are presented in their original numerical order. **Please no staples or folding of corners (your papers won't get lost). A paper clip is OK** Show all necessary work and substantiate all claims. Avoid plagiarism.

Problems

1. Prove that there is a constant c such that, for all $x, y \in \Sigma^*$, $0 \leq K(x|y) \leq K(x) + c$.

   Solution. Let P be a program that is capable of outputting x without the help of any input and consider the following program Q and for which $|P| = K(x)$.

   Input y.

   Return the result of simulating program P.

   Then the size of Q equals the number of bits needed to encode P plus a constant c number of bits to encode the Q's instructions. Therefore,

$$0 \leq K(x|y) \leq |P| + c = K(x) + c.$$

2. Prove that

$$\sum_{|x|=n} 2^{-2K(x|n)}$$

   is bounded by some positive constant C that is independent of n. Hint: use the ideas from Proposition 3.1.

   Solution. As part of the proof of Proposition 3.1 it was observed that, for $i = 0, 1, \ldots, n-1$, there are at most 2^i words x of length n for which $K(x|n) = i$. This observation led to a lower bound for $E[K(x|n)]$ and, in this instance, has the effect of providing an *upper bound* for $2^{-2K(x|n)}$ (why?). Thus,

$$\sum_{|x|=n} 2^{-2K(x|n)} \leq \sum_{i=0}^{n-1} 2^i \cdot 2^{-2i} + 2^{-2n} = \sum_{i=0}^{n-1} \frac{1}{2^i} + 2^{-2n} = 2(1 - \frac{1}{2^n}) + 2^{-2n} < 2$$

   and so is bounded above by $C = 2$ which does not depend on n.

3. A binary word x is said to be **Kolmogorov random** iff $K(x|n) \geq n$, where $n = |x|$. Prove that, if n is sufficiently large and x has $n/4$ 1's and $3n/4$ 0's, then x cannot be Kolmogorov random. Hint: consider Huffman coding!

Solution. Suppose x is Kolmogorov random and has $n/4$ 1's and $3n/4$ 0's, where n is even. Then $K(x|n) \geq n$. Consider the Huffman binary extension code from Example 4.4 of the Information Theory lecture. We may use this code to encode each consecutive pair of bits of x. Moreover, as n increases, the average length per pair equals 1.6875 (this actually requires proof, since there might be a way to arrange the bits so that, for infinitely many n, the average length stays some constant value above 1.675. But for the sake of this exercise, we'll assume this cannot happen). Now consider a program P which stores the $n/2$ codewords $c_1, c_2, \ldots, c_{\frac{n}{2}}$ that encode x.

Input n.

$w = \lambda$.

For $i \in \{1, 2, \ldots, n/2\}$,

$w = w \cdot \text{decode}(c_i)$.

Return w.

Then P outputs x and $|P| = (n/2)(1.675) + c = 0.844n + c$ for some constant $c > 0$. Thus, $K(x|n) < n$, a contradiction.

4. Provide the recursive self-terminating encoding for the number 337.

Solution. $(337)_2 = 101010001$. $|101010001| = 9$ and $(9)_2 = 1001$. $|1001| = 4$ and $(4)_2 = 100$. $|100| = 3$ and $(3)_2 = 11$. Thus, we must concatenate four words and prefix those words with a self-terminating code for 4: 100001. This gives the final encoding

$$100001 \cdot 11 \cdot 100 \cdot 1001 \cdot 101010001.$$

5. Use the `self` programming concept to prove that $K(x)$ is not a URM computable function. Hint: review Theorem 3 and the definition of *recursively enumerable* on pages 17-19.

Solution. Assume $K(x)$ is total computable and consider the following program P's.

For $i \in \mathcal{N}$,

If $K(i) > |\texttt{self}|$, then return i.

Since there are strings with arbitrarily high K-complexity, we know that P returns a value, say x, since $|P| = |\texttt{self}|$ is a constant. Thus, $K(x) > |P|$, but, since P outputs x, it must be true that $K(x) \leq |P|$, a contradiction. Therefore, $K(x)$ is not total computable.