CS 860 ASSIGNMENT #1

JOHN HEARN

2.1.1 (a) Show that $C(0^n|n) \leq c$, where c is a constant independent of n.

(b) Show that $C(\pi_{1:n}|n) \leq c$ where $\pi = 3.1415\ldots$ and c is some constant independent of n.

(c) Show that $C(a_{1:n}|n) \approx n/4$, where a_i is the ith bit in Shakespeare's *Romeo and Juliet*. *Hint:* Use known facts concerning the letter frequencies (entropy) in written English.

(d) What is $C(a_{1:n}|n)$, where a_1 is the ith bit in the expansion of the fine structure constant $a = e^2/\hbar c$, in physics.

2.1.2 Let x be a finite binary string with $C(x) = q$. What is the complexity $C(x^q)$, where x^q denotes concatenation of q copies of x.

2.1.3 Show that there are infinite binary sequences ω such that the length of the shortest program for reference turing machine U to compute the consecutive digits of ω one after another can be significantly shorter than the length of the shortest program to compute an initial n-length segment $\omega_{1:n}$ of ω, for any large enough n.

2.1.5 Below, x, y, and z are arbitrary elements of $\mathbb{N}$. Prove the following:

(a) $C(x|y) \leq C(x) + O(1)$.

(b) $C(x|y) \leq C(x, z|y) + O(1)$.

(c) $C(x|y, z) \leq C(x|y) + O(1)$.

(d) $C(x, x) = C(x) + O(1)$.

(e) $C(x, y|z) = C(y, x|z) + O(1)$.

Date: May 17, 2005.

(f) $C(x|y, z) = C(x|z, y) + O(1)$.

(g) $C(x, y|x, z) = C(y|x, z) + O(1)$.

(h) $C(x|x, z) = C(x|x) + O(1) = O(1)$.

2.1.13 Show that $2C(a, b, c) \leq C(a, b) + C(b, c) + C(c, a) + O(\log n)$.

2.2.5 We say x is an *n-string* if x has length n and $x = n00 \ldots 0$.

(a) Show that there is a constant c such that for all n-strings x we have $C(x|y) \leq c$. (Of course, c depends on the reference Turing machine U used to define C.)

(b) Show there is a constant c such that $C(x|n) \leq c$ for all x in the form of the n-length prefix of $nn \ldots n$.

(c) Let c be as in Item (a). Consider any string x of length n with $C(x|n) \gg c$. Prove that the extension of x to a string $y = x00 \ldots 0$ of length x has complexity $C(y|x) \leq c$. Conclude that there is a constant c such that each sting x, no matter how high its $C(x|l(x))$ complexity, can be extended to a string y with $C(y|l(y)) < c$.

Comments: The $C(x)$ measure contains the information about the *pattern* of 0's and 1's in x and information about the *length* n of x. Since most n's are random, $l(n)$ is mostly about $\log n$. In this case, about $\log n$ bits of the shortest program p for x will be used to account for x's length. For n's that are easy to compute, this is much less. This seems a minor problem at high complexities, but becomes an issue at low complexities, as follows. If the quantities of information related to the *pattern only* is low, say less than $\log n$, for two strings x and y of length n, then distinctions between these quantities for x and y may get blurred in comparison between $C(x)$ and $C(y)$ if the quantity of information related to length n dominates in both. The $C(x|l(x))$ complexity was meant to measure the information content of x apart from its length. However, as the present exercise shows, in that case $l(x)$ may contain already the complete description of x up to a constant number of bits.

2.2.6 (a) Show that there is a constant $d > 0$ such that for every n there are at least $\lfloor 2^n/d \rfloor$ strings of length n with $C(x|n)$, $C(x) \geq n$. *Hint:* There is a constant $c > 0$ such that for every n and every x of length $l(x) \leq n - c$ we have $C(x|n), C(x) > n$ (Theorem 2.1.2). Therefore, there are at most $2^n - 2^{n-c+1}$ programs of length $< n$ available as shortest programs for the strings of length n.

(b) Show that there are constants $c, d' > 0$ such that for every large enough n there are at least $\lfloor 2^n/d' \rfloor$ strings x of length $n-c \leq l(x) \leq n$ with $C(x|n), C(x) > n$. *Hint:* For every n there are equally many strings of length $\leq n$ to be described and potential strings of length $\leq n$ to describe them. Since some programs do not halt (Lemma 1.7.5 on page 34) for every large enough n there exists a string x of length at most n that has $C(x|n), C(x) > n$ (and $C(x|n), C(x) \leq l(x)+c$). Let there be $m \geq 1$ such strings. Given m and n we can enumerate all $2^{n+1}-m-1$ strings x of length $\leq n$ and complexity $C(x|n) \leq n$ by dovetailing the running of all programs of length $\leq n$. The lexicographic first string of length $\leq n$ not in the list satisfies $\log m + O(1) \geq C(x|n) > n$. The unconditional result follows by padding the description of x up to length n.

2.2.8 Prove that for each binary string x of length n there is a y equal to x but for one bit such that $C(y|n) \leq n \log n + O(1)$. *Hint:* the set of binary strings of length n constituting a Hamming code has $2^n/n$ elements and is recursive.

2.3.4 Let ω be an infinite binary string. We call ω *recursive* if there exists a recursive function ϕ such that $\phi(i) = \omega_i$ for all $i > 0$. Show the following:

(a) If ω is recursive, then there is a constant c such that for all n
$C(\omega_{1:n}; n) < c$
$C(\omega_{1:n}|n) < c$
$C(\omega_{1:n}) - C(n) < c$.

Comment: This is easy. The converses also hold but are less easy to show. They follow from items (b), (e), and (f).

(b) For each constant c, there are only finitely many ω such that for all n, $C(\omega_{1:n}; n) \leq c$, and each such ω is recursive.

(c) For each constant c, there are only finitely many ω such that for infinitely many n, $C(\omega_{1:n}; n) \leq c$, and each such ω is recursive.

Comment: Clearly Item (c) implies Item (b).

(d) There exists a constant c such that the set of infinite ω, which satisfies $C(\omega_{1:n}|n) \leq c$ for infinitely many n, has the cardinality of the continuum.

Comment: Conclude that not all such ω are recursive. In particular, the analogue of Item (c) for $C(\omega_{1:n}|n)$ does not hold. Namely, there exist nonrecursive ω for which there exists a constant c such that for infinitely many n we have $C(\omega_{1:n}|n) \leq c$.

Hint: Exhibit a one-to-one coding of subsets of $\mathbb{N}$ into the set of infinite binary strings of which of which infinitely many prefixes are n-strings—in the sense of Example 2.2.5.

(e) For each constant c, there are only finitely many ω such that for all n, $C(\omega_{1:n}|n) \leq c$, and each such ω is recursive.

Comment: Item (e) means that in contrast to the differences between the measures $C(\cdot; l(\cdot))$ and $C(\cdot|l(\cdot))$ exposed by the contrast between Items (c) and (d), Item (b) holds also for $C(\cdot|l(\cdot))$.

(f) For each constant c, there are only finitely many ω with $C(\omega_{1:n}) \leq C(n) + c$ for all n, and each such ω is recursive.

(g) For each constant c, there are only finitely many ω with $C(\omega_{1:n}) \leq l(n) + c$ for all n, and each such ω is recursive.

Comment: Items (f) and (g) show a complexity gap, because $C(n)$ can be much lower than $l(n)$.

(h) There exist nonrecursive ω for which there exists a constant c such that $C(\omega_{1:n}) \leq C(n) + c$ for infinitely many n.

Hint: Use Item (d).