CSE 490 GZ
Assignment 2
Due Friday, January 23, 2004

1. Consider a four symbol alphabet $\{a, b, c, d\}$ with probabilities $P(a) = .1, P(b) = .3, P(c) = .5, P(d) = .1$.

   (a) Use the greedy algorithm to construct a Tunstall code whose output codes have length 4. Be sure to reserve at least one output code as an "escape" code for inputs not coded by the Tunstall code.

   (b) Design a fixed code for the strings that require it.

   (c) Compute the average bit rate (in bits per symbol) for your code.

   (d) Compute the first-order entropy for this model. What is the percentage difference of the Tunstall code from entropy.

2. An alternative run length coder is called the γ-code. Recall that a run length coder really codes sequences of integers which are the the number of zeros between the ones in a binary string. In the γ-code, the integer $n \geq 0$ is coded by first writing $n + 1$ in binary, then preceeding it with m 0's where $m + 1$ is the number of bits that were just written. The γ-codes are started in the table:

number	code
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
$\vdots$	$\vdots$

   (a) Explain why the γ-code is uniquely decodable.

   (b) Encode the binary string 0000010000000001000000000001 using the γ-code. This string is first transformed to three integers, then coded using the γ code.

   (c) Give an expression for the length of the γ-code of n as a function of n.

(d) Compare the length of the γ-code with the Golomb code of order n for encoding $n > 0$. Which is shorter and by how much?

3. Suppose we are in the case where we we know the probability of 0 is much more than 1/2, but we don't know it precisely. The question is how to design an adaptive Golomb code that will work well. One way to do this is by doubling the order until the first 1 is found, then use the optimal Golomb code from that point on. This is called the *doubling algorithm*. For example, suppose the input is $0^{12}10^{10}10^{13}1$. The following table show how the coding proceeds:

order	input	output	calculation for next order
1	0	1	$2 = 2 \times 1$
2	00	1	$4 = 2 \times 2$
4	0000	1	$8 = 2 \times 4$
8	000001	0101	$9 = \lceil -1/\log_2(12/13) \rceil$
9	0^9	1	$15 = \lceil -1/\log_2(21/22) \rceil$
15	01	00010	$8 = \lceil -1/\log_2(22/24) \rceil$
8	0^8	1	$11 = \lceil -1/\log_2(30/32) \rceil$
11	000001	01010	$9 = \lceil -1/\log_2(35/38) \rceil$

Note that after coding $0^{12}1$ we have seen exactly one 1 out of 13 input symbols. This is why we switch to order 9. Similarly, after coding $0^{12}10^9$, we have seen exactly one 1 out of 22 total symbols so we switch to order 15, and so on.

(a) Code the string $0^8 10^{15} 10^8 1$ using the doubling algorithm. What is the compression ratio?

(b) Decode the string 111100110000101111 using the doubling algorithm. What is the compression ratio?

4. Consider the model with three symbols $\{a, b, c\}$ with probabilities $P(a) = 1/2$, $P(b) = 1/4$, and $P(c) = 1/4$. Assume an arithmetic coder with the partition of $[0, 1)$ with a first, b second, and c third.

(a) Using the arithmetic coding algorithm to find the interval for the string *babc*. Compute the tag, short code and prefix code for this string.

(b) Using arithmetic coding decode the string 010111 which encodes a string of length 4.