CSC310 - Fall 2007
Assignment 1

due on Tuesday, October 9th.

Problem 1 [15p]

Consider the following sets of codewords:

- $C_1 = \{0, 10, 1100, 1110, 1111\}$,
- $C_2 = \{00, 10, 001, 101, 0011, 1011\}$,
- $C_3 = \{00, 10, 001, 101, 0011, 1001\}$,
- $C_4 = \{00, 10, 11, 001, 101, 0011\}$.

For each of them, perform the following analysis.

- State whether it satisfies the Kraft-McMillan inequality.
- Is the code uniquely decodable? If not, give an example of an encoding that has at least 2 possible decodings. If it is, give a *brief* justification why it is.
- If the code is uniquely decodable, is it instantaneously decodable? If not, give an example of a string z which might contain the encoding of at least one symbol, and two different strings y, y' such that the first symbols decoded from zy and zy' are different.

Problem 2 [20p]

Consider a source X with symbols $\{a_1, a_2, a_3\}$ and associated probabilities $\{p_1 = .4, p_2 = .5, p_3 = .1\}$. Compute the entropy of X and of its second extension X^2. Find a Huffman code for each of these sources. Compute the average codeword length per symbol from X.

Suppose we are using a Shannon-Fano code for the n-th extension of this source, X^N. How big should N be so that we are guaranteed that the average codeword length per symbol in X is within 1% of the entropy?

Problem 3 [10p]

Suppose Huffman coding is used on a source with 5 symbols. How many different sets of keyword lengths (up to reordering) can the Huffman algorithm output?

Give an example of a set of probabilities $p_1 \geq p_2 \geq p_3 \geq p_4 \geq p_5$ with $p_5 > 0$ such that there are at least 2 different optimal codes for these probabilities that use different sets of keyword lengths.

Problem 4 [20p]

Consider a guessing game in which player A selects a certain outcome from a set A_X and player B tries to guess which outcome was selected by asking a series of YES/NO questions. There is a connection between a guessing strategy of player B and a code for the symbols in A_X: one can encode a certain outcome $a_i \in A_X$ as the sequence of answers to the questions of player B. For more information on this connection, read section 4.1 in the book.

For example, suppose player A selects a number from $\{1, \ldots, 10\}$ with equal probabilities. Here is an example strategy for player B:

```
is x <= 3?
YES: is x = 1?
     YES: output 1
     NO: is x = 3?
         YES: output 3
         NO: output 2
```

```
NO: is x even or x = 9?
    YES: is x < 7?
         YES: is x = 4?
              YES: output 4
              NO: output 6
         NO: is x = 9?
             YES: output 9
             NO: is x = 10?
                 YES: output 10
                 NO: output 8
    NO: is x < 6?
        YES: output 5
        NO: output 7
```

The corresponding code for $\{1, \ldots, 10\}$ is, in order, $\{11, 100, 101, 0111, 001, 0110, 000, 01000, 0101, 01001\}$. This strategy is not optimal if the outcomes are equally likely.

Consider the following guessing game. Player A rolls two fair 6-sided dice, and player B tries to determine the sum X of the dice. Map this game into a coding problem by giving a source alphabet and source probabilities.

Give a detailed strategy that *minimizes the maximum number of questions ever required.* The strategy should correspond to an instantaneous code, and you should specify the tree along with the questions to be asked at each node, in the way the example above is constructed. Briefly justify why the strategy is the best for this setting.

Give a detailed strategy that *minimizes the expected number of questions required.* Again, give the strategy as a tree, explain how you found it, and justify why it is best for this setting.