CSEP 590
Assignment 4

Due Thursday, November 1, 2007

1. In this example we explore Sequitur on some pathological strings.

   (a) Use Sequitur to find the grammar for a^4, a^8, a^{16}.

   (b) Generalize to give a grammar for a^n for n a power of 2.

   (c) Assuming a two letter alphabet compute the compression ratio for Sequitur, as a function of n, for strings of the form a^n where n is a power of 2. Use the encoding describe on slide 44 of the lectures.

2. In this example we explore LZ77 on pathological strings. For this problem we assume LZ77 (solution B), that is, the search buffer has size the length of the string and the look-ahead buffer has size 0.

   (a) Use LZ77 (solution B) to find the sequence of tuples for a^4, a^8, a^{16}.

   (b) Generalize to give the sequence of tuples for a^n for n a power of 2.

   (c) Assuming a two letter alphabet compute the compression ratio for LZ77 (solution B), as a function of n, for strings of the form a^n where n is a power of 2. Use the simple fixed length code for this.

3. In some situations a data file has the property of having a relatively small "working set". This means that the current symbol most often comes from a fairly small set of symbols. For example, consider the string x of symbols in he alphabet $\{a, b, c, d, e, f\}$:

   $$x = \texttt{abccaabbbcabddcbcbddceddeddeccdeefeffddefdfdeeff}$$

   which tends to have a working set of about size 3.

   In the move-to-front algorithms we first give an initial index to each symbol as follows:

0	1	2	3	4	5
a	b	c	d	e	f

   Suppose symbol x with index i is encountered in the input stream. The index i is output. Then the index of x becomes 0 and all the symbols indexed $< i$ have their index increased by 1. For example the input y

   $$y = \texttt{bbbfbb}$$

   has output 100510 because after the first `b` is input the indexing becomes

0	1	2	3	4	5
b	a	c	d	e	f

and after the `f` is input then the indexing becomes

0	1	2	3	4	5
f	b	a	c	d	e

and after the fourth `b` is input the indexing becomes

0	1	2	3	4	5
b	f	a	c	d	e

(a) Compute the empirical entropy of the string x. (The empirical entropy is done using the frequencies of the symbols found in the string.)

(b) Compute the empirical entropy of the string output in the move-to-front algorithm executed on x.

(c) In move-to-front compression both the encoder and decoder know the initial indexing and the output of the move-to-front algorithm is losslessly encoded, say with arithmetic coding. Give one example of a data set that might be amenable to move-to-front compression, and explain why it is so. English text is not an example.