Algorithms and Applications (15-499)
Assignment #1b — Due Date: 11 Feb 03

Problem 1: PPM (20pt)

A. Suppose an encoder using PPM is currently at a context ac and the current state of the dictionary is given by figure 1. Calculate the amount of information, in bits, required to encode the following letter a. Assume that escape count is given by the number of different characters for each context, and exclusion is **not** used. Use $k = 2$.

Context	Counts	Context	Counts	Context	Counts
empty	$a = 4$	a	$b = 2$	ab	$c = 1$
	$b = 2$		$c = 2$	ac	$a = 1$
	$c = 4$				$c = 1$
		b	$c = 1$	bc	$c = 1$
		c	$a = 2$	ca	$a = 1$
					$b = 1$
			$b = 2$	cb	$a = 2$

Figure 1: Dictionary

B. Calculate the number of bits required to encode d for the same context and dictionary as in the previous question. Also list the changes made to the dictionary during this step. Assume the alphabet has 26 characters.

C. How does the answer to part (b) change if we use exclusion?

D. Recompute the answer to part (a) and (b) using $k = 1$ instead of $k = 2$, and without using exclusion.

Problem 2: Lempel-Ziv (20pt)
For this question, assume that Gamma codes are used to encode all integers.

A. How many bits does the LZ77 algorithm use to encode the string $\mathtt{a}^n$ if the window size and the lookahead buffer is unbounded? Assume that the probability of character a is given by $p_{\mathtt{a}}$.

B. How many bits does LZW take to encode the same string with an unbounded dictionary?

Problem 3: Burrows Wheeler (40pt)

A. Let $H_0(s)$ denote the entropy of a string s based on single character probabilities. This is known as the *zeroth order entropy* of s.

Given a string s, let W^k be the set of all distinct k character substrings of s. For $w \in W^k$, let $succ(w)$ denote the string formed by each character that follows an occurrence of w in s.[1] Calculate the number of bits required to encode s, using contexts of length k. Express the answer in terms of $H_0(succ(w))$, for $w \in W^k$. This is known as the kth order conditional entropy $H_k(s)$. Assume that the cost of encoding character probabilities is negligible.

B. Let s' be the output of the Burrows Wheeler algorithm on string s. Show that for any $w \in W^k$ (defined as above), $succ(w)$ (or a permutation of it) is a substring of s'.

The following questions give insight into why it is important to apply Move-To-Front to the output of BW before encoding it. Assume that the cost of encoding character probabilities is negligible.

C. What is the output of the BW transform (before MTF) on the string `aa(ba)`^{20}`bb`? Roughly how many bits does Arithmetic Encoding take to encode this output? Give the answer to within an accuracy of ± 2 bits.

D. Compute the number of bits taken by Arithmetic Encoding on the BW output in part (c) after being processed using MTF.

E. Let $s' = s_1 s_2 \cdots s_t$. [Manzini'99] proves that the number of bits in the Move-To-Front encoding of s' is roughly at most $c \sum_i |s_i| H_0(s_i)$, where c is some constant. Use this to prove that the number of bits taken to encode the output of BW with MTF on a string s is at most $cH_k(s)$ for any value of k.

[1]For example, if $s =$`mississippi`, $succ$(`ss`) = `ii` and $succ$(`si`) = `sp`.

Problem 4: Wavelets (20pt)

A. Consider the interger range from 1 to $n = 2^k$. Write down the following functions as a sum of the n Haar Wavelets ϕ_{ij} scaled to fit that range.

$$f(x) = x$$

$$f(x) = \begin{cases} 1 & x \leq 3/8n \\ 0 & \text{otherwise} \end{cases}$$

B. For each step in the Haar transform the "low-pass" term (the average) and the "high-pass" them (the difference) each depend on two values. In particular:

$$\begin{aligned} h_i &= (a_{2i} - a_{2i+1})/2 \\ l_i &= (a_{2i} + a_{2i+1})/2 \end{aligned}$$

The (5-tap, 3-tap) integer wavelet transform used by the revesible version of JPEG2000 uses the following functions to calculate the low and high pass terms.

$$\begin{aligned} h_i &= a_{2i+1} - \left\lfloor \frac{a_{2i} + a_{2i+2}}{2} \right\rfloor \\ l_i &= a_{2i} + \left\lfloor \frac{h_{i-1} + h_i + 2}{4} \right\rfloor \end{aligned}$$

It is called a (5-tap, 3-tap) since the low-pass term depends on five values and the high-pass term depends on three values. In practice it works much better than the Haar transform since it has broader coverage. Show that this is reversible by giving the function for generating the a_i from the h_i and l_i.