Basic Compression Methods

1 Exercise 1 – Huffman

1. The DNA alphabet is {A,C,T,G} with respective probabilities {0.5,0.3,0.15,0.05}. Calculate the entropy and gives Huffman coding.

2. Give the Huffman tree and encoding of "sir sid eastman easily teases sea sick seals".

2 Exercise 2 – MTF Coding

1. Give the Move-to-Front encoding of "the boy on my right is the right boy", proceeding word by word.

2. Decode 0the, 1right, 2of, 2, 3boy, 4is, 5on, 6my, 6.

3 Exercise 3 – LZ??

1. Apply the LZ77 compression algorithm to "sir sid eastman easily teases sea sick seals" with a search buffer of 10 characters, and a look ahead buffer of 5.

2. Apply the LZ78 compression algorithm to "sir sid eastman easily teases sea sick seals". Which structure could be used to make the dictionary search faster?

3. Apply the LZW compression algorithm to "sir sid eastman easily teases sea sick seals". Initialize the dictionary with the ASCII Table.

4. Give the result of the decompression using LZ77 of (0,0,a) (0,0,b) (0,0,r) (3,1,a) (5,1,c) (7,1,d) (1,1,a) (10,4,*).

5. Give the result of the decompression using LZ78 of (0,a) (1,a) (0,b) (3,a) (4,a) (5,a) (4,b).

6. Give the result of the decompression using LZW of 97, 108, 102, 32, 101, 97, 115, 105, 108, 121, 259, 97, 116, 115, 32, 256, 102, 271, 97.

4 Exercise 4 : arithmetic coding

Consider again the DNA alphabet {A,C,T,G} with respective symbols probabilities {0.5,0.3,0.15,0.05}.

Evaluate the arithmetic code for the chain : ACTAGC and propose the decoding process

5 Exercise 5

Provide the arithmetic code for the sentence: BE_A_BEE

6 Exercise 6

In order to avoid the calculus of a to high number arithmetic codes are modified in a way to limit the size of the low and high bound (let say 4) to a certain number of digits and shift to the left the most significant bit and insert a 0 to the least significant bit of low bound and a 9 of the high bound.

Encode the sentence " SWISS_MISS " by using the shifting approach.

7 Exercise 7

Recall : the PPM encoder uses a statistical model of the text in a context way (order 0 is the character level, order 1 counts of groups of 2 characters and so on) and an arithmetic coder for all recognized symbol. We assume the arithmetic coder achieves the entropy. We assume that the 14-symbol string *“assanissimassa”* has been completely input and encoded, so the current order-2 context is *“sa”* (f is the counting and p the probability) :

Order 2			Order 1			Order 0		
Context	f	p	Context	f	p	Symbol	f	p
as→s	2	2/3	a→ s	2	2/5	a	4	4/19
esc	1	1/3	a→ n	1	1/5	s	6	6/19
			esc→	2	2/5	n	1	1/19
ss→a	2	2/5				i	2	2/19
ss→i	1	1/5	s→ s	3	3/9	m	1	1/19
esc	2	2/5	s→ a	2	2/9	esc	5	5/19
			s→ i	1	1/9			
sa→n	1	1/2	esc	3	3/9			
esc	1	1/2						
			n→ i	1	1/2			
an→i	1	1/2	esc	1	1/2			
esc	1	1/2						
			i→ s	1	1/4			
ni→s	1	1/2	i→ m	1	1/4			
esc	1	1/2	esc	2	2/4			
is→s	1	1/2	m→ a	1	1/2			
esc	1	1/2	esc	1	1/2			
si→m	1	1/2						
esc	1	1/2						
im→a	1	1/2						
esc	1	1/2						
ma→s	1	1/2						
esc	1	1/2						

1. Suppose that the **15**th symbol to be input was a *n*. How many bit(s) would it take to encode it?
2. Suppose that the **15**th symbol to be input was a *s*. How many bits would it take to encode it?
3. Suppose that the **15**th symbol to be input was a *d*. How many bits would it take to encode it?
4. Suppose that the **16**th symbol is also a *d*. How many bits would it take to encode this second d?