PF1
Exercise session n° 4 - Codes, encoding and compression

Exercise 1 :

We consider fixed length encoding.

1. Let $X := \{a, b, c\}$ and $\mathcal{P}(X) := \{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. Propose a binary encoding of the set $\mathcal{P}$ such that we can reconstruct the set back.

2. Let $X := \{a, b, c, d\}$. How many possible words of length n are there ? How many words of length $< n$?

3. Let $X := \{0, 1, \ldots, 9, A, B, \ldots, F\}$ and $Y := \{0, 1\}$. What should be the length of a minimal fixed length encoding ? Give the name of a very famous such minimal encoding of X into Y. With this encoding, how is encoded ABC ? Which word is encoded by 00111010 ?

Exercise 2 :

We consider length-varying encoding.

1. Let $X := \{a, b, c, d, e\}$ and $Y := \{0, 1\}$. Let τ be the encoding map defined by :

$$\tau(a) = 0 \quad \tau(b) = 10 \quad \tau(c) = 01 \quad \tau(d) = 110 \quad \tau(e) = 1011.$$

Encode *babc* and *bcac*. What is the problem ?

2. A *prefix* code is a set of words such that no encoding of one letter is the prefix of the encoding of another letter.

a Let $\sigma(a) = 101000, \sigma(b) = 01, \sigma(c) = 1010$. Does σ define a prefix code ?

b Does σ define a suffix code ?

c Any prefix code can be uniquely decoded. Is the converse true ?

3. For the following sets, tel whether it is a code or not, and if so, tell if it is prefix, suffix. Justify your answers.

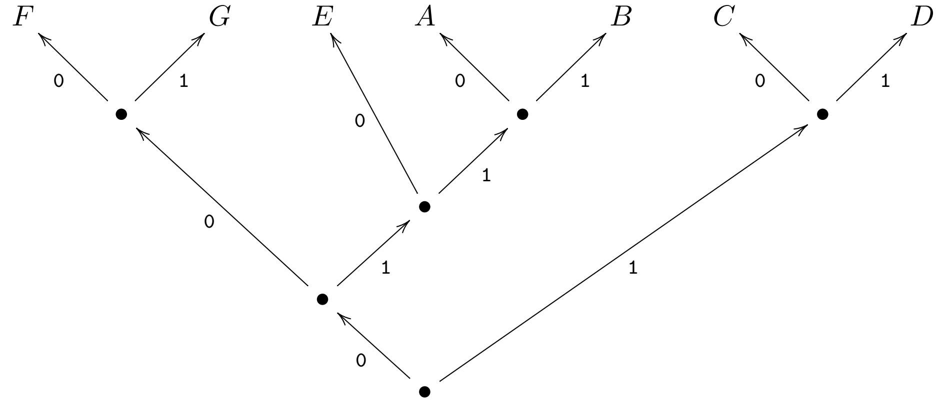
- $E_1 = \{01, 100, 1101, 0111\}$
- $E_2 = \{0, 10\}$
- $E_3 = \{0, 11, 100, 101\}$
- $E_4 = \{001, 100, 101\}$
- $E_5 = \{01, 10, 101\}$
- $E_6 = \{0, 011, 10\}$
- $E_7 = \{0^n 1 \mid n \in \mathbb{N}\}$
- $E_8 = \{01, 101, 110, 1110, 0100\}$
- $E_9 = \{0, 11, 01110, 10101\}$

Exercise 3 :

1. Using the following tree, unencode the two binary words

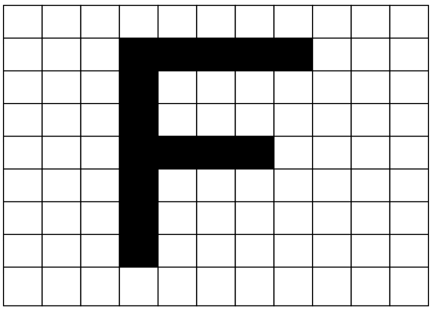
$b_1 = 1001000110010001100101001000110010001100101011011011000011011000$

$b_2 = 101101101100001101100001111100111000001110000111110011100000111000$



2. What is the frequency of each of the seven letters in the word $b = b_1 b_2$? Check that this really is a Huffman tree built on these frequencies. Compute the average length of a word of a code. Compare with a fixed length code.

Exercise 4 :



We consider this black and white representation of the letter **F**. We use 0 to represent the color white and 1 for the color black. In the following, we suppose that the dimensions of the picture are fixed and known, and don't have to be encoded.
RLE (Run-length encoding) replaces each group of 0 (or 1) with it's length. We consider that we always begin with a 0, so the encoding of 0000000111 is $7, 3$ and the encoding of 110000 is $0, 2, 4$.

1. Without encoding, how many bits do we need to represent the whole picture ?

2. Use RLE to represent the picture, reading each line from left to right, independently and from top to bottom. Give the corresponding sequence s.

3. What is the minimum length of a fixed length encoding necessary to encode all the integers in the sequence s ? What is the corresponding size of the picture ? Is it a compression code ? If so, provide the compression rate.

4. From the sequence s, provide the array mapping each value to its number of occurrences.

5. Build the Huffman tree corresponding to the previous array.

6. Encode the sequence s (i.e. the picture) using the Huffman code obtained. What is the sized of the encoded picture ? In this case (RLE + Huffman), is it a compression ? If yes, provide the compression rate.

Exercise 5 :

Prove that for a given frequency list there can correspond several Huffman trees whose structure (forgetting the labels) are different.