Computer Science Fundamentals WS 2021/22

Prof. Dr. J. Schmidt; D. Stecher, M.Sc. Exercise 6

Please solve the following exercises at home prior to the tutorial:

Excercise 1

Given is a message source using the alphabet A shown in the table below with symbols $\{x_i\}$ and the respective occurrence probabilities $\{p_i\}$:

x_i	a	b	c	*	=	+
p_i	0.25	0.25	0.20	0.15	0.10	0.05

a) Calculate the entropy of this data source.

b) Create a binary encoding for this example. For word length, use the information content rounded up to the nearest integer. Determine the average word length and redundancy. Draw the corresponding code-tree and try to optimize the result, i.e. reduce the average word length even further.

c) Determine the optimal code for alphabet A using the Huffman algorithm as well as the corresponding average word length and redundancy.

d) Give the Huffman code for the following message:

a*a+b*b=c*c

e) Determine a code using the Fano algorithm as well as the corresponding average word length and redundancy.

The following exercises will be done during the tutorial:

Excercise 2

Given is the alphabet $\{x_i\}$ with corresponding occurrence probabilities $\{p_i\}$ as shown in the table below:

x_i	A	E	I	O	U	Y
p_i	0.105	0.22	0.105	0.04	0.45	0.08

a) Calculate the information contents $I(x_i)$ and the entropy *H*.

b) Create the optimal binary code using the Huffman algorithm.

c) Create a binary code using the Fano algorithm.

d) Give the average word length and redundancy for both b) and c).

e) Encode the alphabet using an optimally short code with constant word length (Block code). What compression factor do we get compared to the Huffman code from b)?

Excercise 3

Given is an alphabet $A = \{x_i\}$ with n symbols and occurrence probabilities p_i. What conditions must be met by n and p_i for the Huffman Code to yield a code with constant word length?