INFORMATION THEORY
Master of Logic, Master AI, Master CS, University of Amsterdam, 2018
TEACHER: Christian Schaffner, TAs: Yfke Dulek, Esteban Landerreche, Kyah Smaal

# Practice problem set 3

This week's exercises deal with source codes and data compression. You do not have to hand in these exercises, they are for practicing only. During the work session, start with solving the exercise you will be moderating. Work out a full solution on paper/computer and get it approved by the teacher. Also think about the following questions: What is the point of the exercise? What kind of problems will students encounter when solving this problem? What kind of questions could you ask the presenter on Friday? Problems marked with a ★ are generally a bit harder. If you have questions about any of the exercises, please post them in the discussion forum on Canvas, and try to help each other. We will also keep an eye on the forum.

## Problem 1: Prefix-free arithmetic codes

**(a)** What are the names of the binary intervals $[\frac{6}{8}, \frac{7}{8})$ and $[\frac{7}{16}, \frac{8}{16})$?

**(b)** What are the binary intervals with the names 0110 and 011?

**(c)** Prove that if the name of a binary interval $I$ is the prefix of the name of another binary interval $J$, it must be that $J \subset I$.

**(d)** Use (c) to prove that for any source, the resulting arithmetic code $AC^{pf}$ is indeed prefix-free.

## Problem 2: Comparison of arithmetic codes

**(a)** Given $X$ with $\mathcal{X} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and $P_X(\mathtt{a}) = P_X(\mathtt{b}) = 1/3$, $P_X(\mathtt{c}) = P_X(\mathtt{d}) = 1/6$. Construct the standard arithmetic code $AC$ as well as the prefix-free arithmetic code $AC^{pf}$ for this source. How do the average codeword lengths compare?

**(b)** Recall the proof that $\ell_{AC}(P_X) \leq H(X)+1$. Adapt the proof to show that for the prefix-free procedure, the average codeword length $\ell_{AC^{pf}}(P_X)$ is upper bounded by $H(X)+2$ for any source.

## Problem 3: An optimal code

Let $X$ be a random variable which takes on values in the finite set $\mathcal{X}$.

**(a)** Show that if there exists an $n \in \mathbb{N}_{>0}$ such that for all $x \in \mathcal{X}$, $P_X(x) = \frac{1}{2^n}$, then there exists a uniquely decodable source code whose expected length equals the entropy.

**(b)** Show that if for all $x \in \mathcal{X}$, there exists an $n \in \mathbb{N}_{>0}$ such that $P_X(x) = \frac{1}{2^n}$, then there exists a uniquely decodable source code whose expected length equals the entropy.

## ★ Problem 4: Unique decodability

Construct a binary symbol code with a finite number of codewords that is uniquely decodable, but for which there exists an *infinite* binary string that can be decoded in more than one way.

## Problem 5: 5

Prove the following statement. If $|\mathcal{X}| > 1$, then the word lengths of a binary Huffman code for the source $P_X$ must satisfy Kraft's inequality with equality, i.e. $\sum_i 2^{-\ell_i} = 1$.

## ★ Problem 6: Optimal codeword lengths

Although the codeword lengths of an optimal variable length code are complicated functions of the source probabilities, it can be said that less probable symbols are encoded into longer codewords. Suppose that the message probabilities are given in decreasing order $p_1 > p_2 \geq \cdots \geq p_m$.

**(a)** Prove that for any binary Huffman code, if the most probable message symbol has probability $p_1 > 2/5$, then that symbol must be assigned a codeword of length 1.

**(b)** Prove that for any binary Huffman code, if the most probable message symbol has probability $p_1 < 1/3$, then that symbol must be assigned a codeword of length $\geq 2$.