TSBK08

Data compression

Exercises

Contents

Informationskodning, ISY, Linköpings universitet, 2016

Problems

1 Information theory

1.1 The random variable X takes values in the alphabet $\mathcal{A} = \{1, 2, 3, 4\}$. The probability function is $p_X(x) = \frac{1}{4},\ \forall x \in \mathcal{A}$.

Calculate $H(X)$.

1.2 The random variable Y takes values in the alphabet $\mathcal{A} = \{1, 2, 3, 4\}$. The probability function is $p_Y(1) = \frac{1}{2},\ p_Y(2) = \frac{1}{4},\ p_Y(3) = p_Y(4) = \frac{1}{8}$.

Calculate H(Y).

1.3 Suppose that X and Y in 1.1 and 1.2 are independent. Consider the random variable (X, Y).

a) Determine $p_{XY}(x, y)$.

b) Calculate $H(X, Y)$

c) Show that $H(X, Y) = H(X) + H(Y)$ when X and Y are independent.

d) Generalization: Show that

$$H(X_1, X_2, \ldots, X_n) = H(X_1) + \ldots + H(X_n)$$

Whenever all the variables are mutually independent.

1.4 The random variable Z takes values in the alphabet $\mathcal{A} = \{1, 2, 3, 4\}$.

a) Give an example of a probability function p_Z that maximizes $H(Z)$. Is p_Z unique?

b) Give an example of a probability function p_Z that minimizes $H(Z)$. Is p_Z unique?

1.5 Let the random variable U take values in the infinite alphabet $\mathcal{A} = \{0, 1, 2, \ldots\}$, with probability function $p_U(u) = q^u(1-q),\quad 0 < q < 1,\quad u \in \mathcal{A}$.

a) Confirm that $\sum_{u=0}^{\infty} p_U(u) = 1$.

b) Calculate $H(U)$

c) Calculate the average value of U

1.6 Show that
$$I(X,Y) = H(X) - H(X \mid Y) \;\geq\; 0$$
Hint: Use the inequality $\ln x \leq x - 1$.

1.7 A binary memoryless source where the two symbols have the probabilities $\{p, 1-p\}$ has the entropy H. Express the entropies for the following sources in H.

a) A memoryless source with four symbols with the probabilities $\{\frac{p}{2}, \frac{p}{2}, \frac{1-p}{2}, \frac{1-p}{2}\}$

b) A memoryless source with three symbols with the probabilities $\{\frac{p}{2}, \frac{p}{2}, 1-p\}$

c) A memoryless source with four symbols with the probabilities $\{p^2, p(1-p), (1-p)p, (1-p)^2\}$.

1.8 Let X be a random variable and f any deterministic function of the alphabet. Show that

a) $H(f(X) \,|\, X) \;=\; 0$

b) $H(X, f(X)) \;=\; H(X)$

c) $H(f(X)) \;\leq\; H(X)$

1.9 Let X and Y be two independent random variables. Show that
$$H(X) \;\leq\; H(X+Y) \;\leq\; H(X,Y)$$

1.10 Given three random variables X, Y and Z, show that
$$H(X \,|\, Z) \;\leq\; H(X \,|\, Y) \;+\; H(Y \,|\, Z)$$
Hint: Start with $H(X \,|\, Y) \;\geq\; H(X \,|\, YZ)$ and use the chain rule.

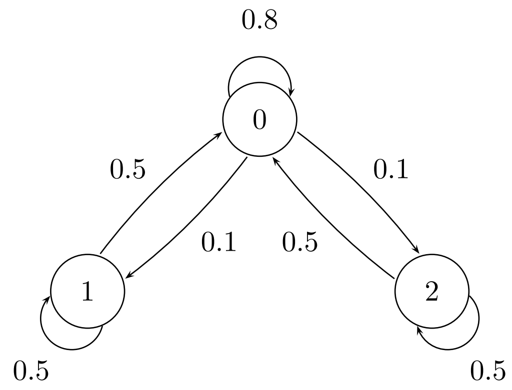
1.11 Show that
$$H(X_1, \ldots, X_{n+1}) \;=\; H(X_1, \ldots, X_n) + H(X_{n+1} \,|\, X_1, \ldots X_n)$$

1.12 A uniformly distributed random variable X takes values in the alphabet $\{0000, 0001, 0010, \ldots, 1011\}$ (the numbers 0 to 11 written as four bit binary numbers).

a) What is the entropy of each bit?

b) What is the entropy of X?

1.13 A Markov source X_i of order 1 with alphabet $\{0, 1, 2\}$ has transition probabilities according to the figure. Calculate the stationary probabilities of the states.



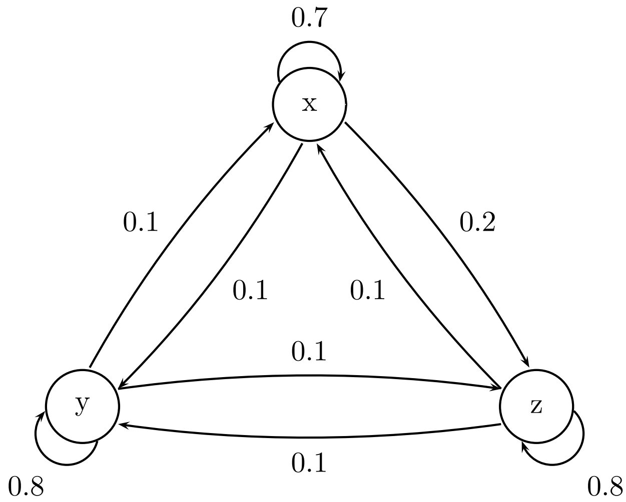
1.14 Consider the Markov source in problem 1.13.

a) Calculate the memoryless entropy $H(X_i)$ of the source.

b) Calculate the block entropy $H(X_i, X_{i+1})$ of the source. Compare this to $H(X_i)+H(X_{i+1}) = 2H(X_i)$.

c) Calculate the entropy rate of the source.

1.15 A Markov source X_n of order 1, with alfabet $\mathcal{A} = \{x, y, z\}$, is given by the state diagram below



Calculate the entropies $H(X_n)$, $H(X_n|X_{n-1})$ and $H(X_n, X_{n+1}, X_{n+2}, X_{n+3}, X_{n+4})$ for the source.

1.16 A second order Markov source X_i with alphabet $\{a, b\}$ has the transition probabilities $p_{X_i|X_{i-1}X_{i-2}}$ below:

$$p(a|aa) = 0.7, \quad p(b|aa) = 0.3, \quad p(a|ba) = 0.4, \quad p(b|ba) = 0.6$$

$$p(a|ab) = 0.9, \quad p(b|ab) = 0.1, \quad p(a|bb) = 0.2, \quad p(b|bb) = 0.8$$

Calculate the entropies $H(X_i)$, $H(X_iX_{i-1})$, $H(X_iX_{i-1}X_{i-2})$, $H(X_i|X_{i-1})$ and $H(X_i|X_{i-1}X_{i-2})$.

1.17 A fax machine works by scanning paper documents line by line. The symbol alphabet is black and white pixels, ie $\mathcal{A} = \{b,\ w\}$. We want to make a random model X_i for typical documents and calculate limits on the data rate when coding the documents.

From a large set of test documents, the following conditional probabilities $p(x_i|x_{i-1}, x_{i-2})$ (note the order) have been estimated.

$$\begin{array}{ll} p(w|w,w) = 0.95 & p(b|w,w) = 0.05 \\ p(w|w,b) = 0.9 & p(b|w,b) = 0.1 \\ p(w|b,w) = 0.2 & p(b|b,w) = 0.8 \\ p(w|b,b) = 0.3 & p(b|b,b) = 0.7 \end{array}$$

a) The given probabilities imply a Markov model of order 2. Draw the state diagram for this Markov model and calculate the stationary probabilities.

b) Calculate the entropies $H(X_i)$, $H(X_i|X_{i-1})$ and $H(X_i|X_{i-1}, X_{i-2})$ for the model.

1.18 For a stationary binary source X_n with alphabet $\mathcal{A} = \{a, b\}$, the probabilties $p(x_n, x_{n-1}, x_{n-2})$ (note the symbol order) for 3-tuples have been estimated from a large amount of data

$$\begin{array}{llll} \text{p(aaa)} = 36/95 & \text{p(aab)} = 9/95 & \text{p(aba)} = 1/95 & \text{p(abb)} = 9/95 \\ \text{p(baa)} = 9/95 & \text{p(bab)} = 1/95 & \text{p(bba)} = 9/95 & \text{p(bbb)} = 21/95 \end{array}$$

a) Is a Markov source of order 1 a good model for the source?

b) What is the best estimate of the entropy rate of the source that we can make?

2 Source coding

2.1 A suggested code for a source with alphabet $\mathcal{A} = \{1, \ldots, 8\}$ has the codeword lengths $l_1 = 2$, $l_2 = 2$, $l_3 = 3$, $l_4 = 4$, $l_5 = 4$, $l_6 = 5$, $l_7 = 5$ and $l_8 = 6$. Is it possible to construct a prefix code with these lengths?

2.2 A memoryless source has the infinite alphabet $\mathcal{A} = \{1, 2, 3, \ldots, \}$ and symbol probabilities $P = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots\}$, ie $p(i) = 2^{-i},\ i \in \mathcal{A}$.

Construct an optimal binary prefix code for the source and calculate the expected data rate R in bits/symbol.

2.3 A memoryless source has the alphabet $\mathcal{A} = \{x, y, z\}$ and the symbol probabilities

$$p(x) = 0.6, \quad p(y) = 0.3, \quad p(z) = 0.1$$

a) What is the lowest average rate that can be achieved when coding this source?

b) Construct a Huffman code for single symbols from the source and calculate the average data rate in bits/symbol.

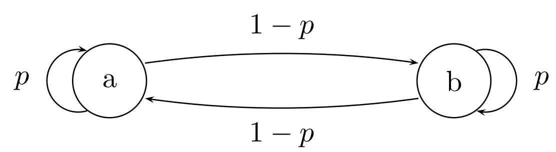
c) Construct a Huffman code for pairs of symbols from the source and calculate the average data rate in bits/symbol.

2.4 A memoryless source has the alphabet $\mathcal{A} = \{a,\ b,\ c,\ d\}$. The symbol probabilities are

$$p(a) = 0.5,\ p(b) = 0.2,\ p(c) = 0.2,\ p(d) = 0.1$$

What is the resulting average data rate (in bits/symbol) if we code pairs of symbols from the source using a Huffman code?

2.5 Consider the following Markov source of order 1, where $p = \sqrt[8]{0.5}$:



Construct Huffman codes for the source where 2 and 3 symbols are coded at a time. Calclulate the rates of the two codes. Which code is the best?

2.6 Consider the source in problem 2.5. It produces runs of a and b. Instead of coding symbols, we can code the length of each run. We create a new source R which has an infinite alphabet of run lengths $\mathcal{B} = \{1, 2, 3, \ldots\}$.

a) What is the probability of a run of length r?

b) What is the average run length (in symbols/run) ?

c) What is the entropy of R (in bits/run)?

d) What is the entropy rate of the source (in bits/symbol) ?

2.7 We now want to construct a simple systematic code for the run lengths from the source in problem 2.5.

a) Construct a four bit fixlength code for run lengths 1 to 14. Longer runs are coded as 15 followed by the codeword for run of length-15, ie the run length 15 is coded as 15 0, the run length 17 is coded as 15 2, the run length 40 is coded as 15 15 10 and so on.

   Calculate the data rate for the code in bits/symbol.

b) Change the codeword length to five bits and do the same as above.

2.8 We now want to code the run lengths from the source in problem 2.5 using Golomb coding.

a) How should the parameter m be chosen so that we get an optimal code?

b) Calculate the resulting data rate.

2.9 We want to send documents using a fax machine. The fax handles the colours black and white. Experiments have shown that text areas and image areas of documents have different statistical properties. The documents are scanned and coded row by row, according to a time discrete stationary process. The following conditional probabilities have been estimated from a large set of test data.

Colour of current pixel	*Probability of colour of next pixel*			
	Text area		*Image area*	
	black	white	black	white
black	0.5	0.5	0.7	0.3
white	0.1	0.9	0.2	0.8

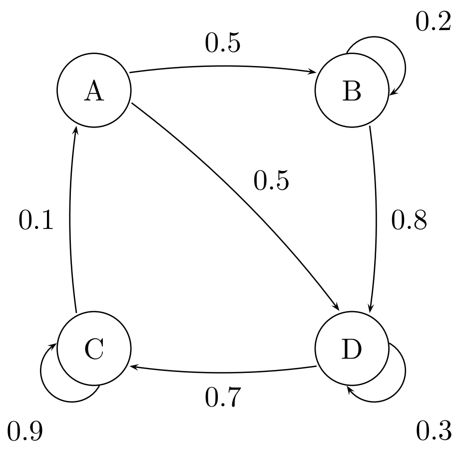
The probability that we are in a text area is $\frac{4}{5}$ and the probability that we are in an image area is $\frac{1}{5}$.

Assume that the estimated probabilities are correct and answer the following questions.

a) Assume that we can neglect the cost of coding what areas of the document that are text and what areas that are images. Calculate an upper bound for the lowest data rate that we can get when coding documents, in bits/pixel.

b) Construct a Huffman code for text areas that has a data rate of 0.65 bits/pixel or less.

2.10 Consider the following Markov source of order 1



a) Show that it is possible to code the signal from the source with a data rate that is less than 0.6 bits/symbol.

b) Construct optimal tree codes for single symbols and for pairs of symbols from the source. Calculate the resulting data rates.

2.11 A memoryless source has the alphabet $\mathcal{A} = \{a_1, a_2, a_3\}$ and probabilities $P = \{0.6, 0.2, 0.2\}$. We want to code the sequence $a_3 a_1 a_1 a_2$ using arithmetic coding. Find the codeword. Assume that all calculations can be done with unlimited precision.

2.12 Do the same as in problem 2.11, but let all probabilities and limits be stored with six bits precision. Shift out codeword bits as soon as possible.

2.13 Assume that the source is the same as in problem 2.11 and problem 2.12. Let all probabilities and limit be stored with six bits precision. Decode the codeword **1011101010000**. We know that the codeword corresponds to five symbols.

2.14 A system for transmitting simple colour images is using the colours white, black, red, blue, green and yellow. The source is modelled as a Markov source of order 1 with the following transition probabilities

State	probability for next state					
	white	black	red	blue	green	yellow
white	0.94	0.02	0.01	0.01	0.01	0.01
black	0.05	0.50	0.15	0.10	0.15	0.05
red	0.03	0.02	0.90	0.01	0.01	0.03
blue	0.02	0.02	0.02	0.90	0.03	0.01
green	0.02	0.02	0.01	0.03	0.90	0.02
yellow	0.03	0.01	0.03	0.01	0.03	0.89

We use arithmetiv coding to code sequences from the source. The coder uses the conditional probabilities while coding.

a) A sequence starts with red, white, white. What intervall does this correspond to? Assume that the previous pixel was red.

b) The decoder is waiting for a new sequence. The bit stream 110100111100001100 is received. What are the first two colours in this sequence? The last pixel in the previous sequence was black.

2.15 We want to code a stationary binary memory source with alphabet $\mathcal{A} = \{a, b\}$. The following block probabilities $p(x_n, x_{n+1})$ have been estimated and can be assumed to be correct:

$$
\begin{aligned}
p(aa) &= 1/7 \\
p(ab) &= 1/7 \\
p(ba) &= 1/7 \\
p(bb) &= 4/7
\end{aligned}
$$

Construct a codeword for the sequence *bbab* by using arithmetic coding. The coder should utilize conditional probabilities. Assume that the symbol before the sequence to be coded is b.

2.16 Consider the source in problem 1.17. Use arithmetic coding to code the sequence

$$wbbwww$$

The memory of the source should be utilized in the coder. Imaginary pixels before the first pixel can be considered to be white. You can assume that the coder can store all probabilities and interval limits exactly. Give both the resulting interval and the codeword.

2.17 Consider the following code for coding sequences from a binary source (alphabet $\mathcal{A} = \{0,\ 1\}$): Choose a block length n. Divide the sequence into blocks of length n. For each block, count the number of ones, w. Encode the number w as a binary number using a fixed length code. Then code the index of the block in an enumeration of all possible blocks of length n containing exactly w ones, also using a fix length code.

a) How many bits are needed to code a block of w ones and $n - w$ zeros?

b) Assuming $n = 15$, how many bits are needed to code the sequence x=000010000000100?

c) Note that the coding algorithm does not take the symbol probabilities into account.

Assuming that the source is memory-less, show that the coding algorithm is *universal*, ie that the average rate approaches the entropy rate of the source when $n \to \infty$?

2.18 A source has the alphabet $\mathcal{A} = \{a, b\}$. Code the sequence

$$ababbaaababbbbaaabaaaaaababba...$$

using LZ77. The history buffer has the size 16. The maximum match length is 15. Check your solution by decoding the codewords.

2.19 Code the sequence in problem 2.18 using LZ78.

2.20 Code the sequence in problem 2.18 using LZW.

2.21 A source has the alphabet $\{a, b, c, d, e, f, g, h\}$. A sequence from the source is coded using LZW and gives the following index sequence:

$$5, 0, 8, 10, 4, 3, 12, 12, 3, 6, 17, \ldots$$

The starting dictionary is:

index	sequence
0	a
1	b
2	c
3	d
4	e
5	f
6	g
7	h

Decode the index sequence. Also give the resulting dictionary.

2.22 A source has the alphabet $\mathcal{A} = \{a, b, c, d, e, f, g, h\}$.

Code the sequence beginning with

$$bagagagagebaggebadhadehadehaf\ \ldots$$

using LZSS. The history buffer pointer should be coded using 6 bit fixed length codewords and the match lengths should be coded using 3 bit fixed length codewords.

Give the resulting codewords.

2.23 a) A source has the alphabet $\mathcal{A} = \{a,\ b,\ c,\ d\}$. The sequence *dadadabbbc* is coded using Burrows-Wheelers block transform (BWT). What's the transformed sequence and the index?

b) The transformed sequence from problem a is coded with move-to-front coding (mtf). What's the coded sequence?

2.24 A source has the alphabet $\mathcal{A} = \{v,\ x,\ y,\ z\}$. A sequence is coded using BWT and mtf. The resulting index is 1 and the mtf-coded sequence is 2,2,3,0,0,1,1,0,2,0,0. Decode the sequence

3 Differential entropy and rate-distortion theory

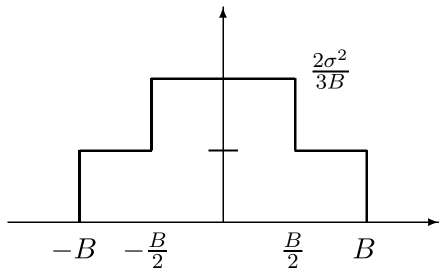
3.1 Calculate the differential entropy of the triangular distribution with probability density function $f(x)$

$$f(x) = \begin{cases} \frac{1}{a} + \frac{x}{a^2} & ; -a \leq x \leq 0 \\ \frac{1}{a} - \frac{x}{a^2} & ; 0 \leq x \leq a \\ 0 & ; \text{otherwise} \end{cases}$$

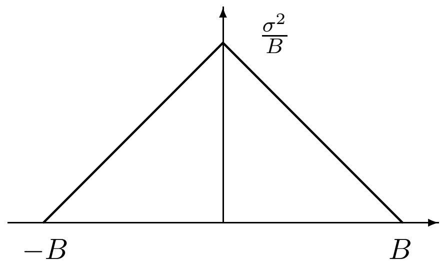
3.2 Calculate the differential entropy of the exponential distribution with probability density function $f(x)$

$$f(x) = \begin{cases} \frac{1}{\lambda} e^{-\frac{x}{\lambda}} & ; x \geq 0 \\ 0 & ; x < 0 \end{cases}$$

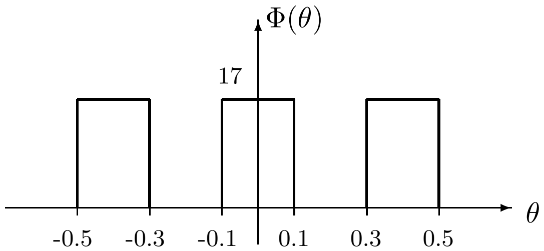
3.3 Determine the rate-distortion function $R(D)$ for a gaussian process with power spectral density,



3.4 Determine the rate-distortion function for a gaussian process with power spectral density,



3.5 A signal is modelled as a time-discrete gaussian stochastic process X_n with power spectral density $\Phi(\theta)$ as below.



Assume that we encode the signal at an average data rate of 3 bits/sample. What is the theoretically lowest possible distortion that we can achieve?

Solutions

1.1 $H(X) = -\sum_{i=1}^{4} p_X(x_i) \cdot \log p_X(x_i) = -4 \cdot \frac{1}{4} \cdot \log \frac{1}{4} = 2$

1.2 $H(Y) = -\sum_{i=1}^{4} p_Y(y_i) \cdot \log p_Y(y_i) = -\frac{1}{2} \cdot \log \frac{1}{2} - \frac{1}{4} \cdot \log \frac{1}{4} - 2 \cdot \frac{1}{8} \cdot \log \frac{1}{8} = 1.75$

1.3 a) Independence gives $p_{XY}(x,y) \;=\; p_X(x) \cdot p_Y(y)$. Thus

		y			
	$p(x,y)$	1	2	3	4
	1	1/8	1/16	1/32	1/32
x	2	1/8	1/16	1/32	1/32
	3	1/8	1/16	1/32	1/32
	4	1/8	1/16	1/32	1/32

b) Calculating the entropy from the 16 probabilities in a) gives

$$H(X,Y) = 3.75$$

Since X and Y are independent you can also calculate the entropy from

$$H(X,Y) = H(X) + H(Y) = 3.75$$

c)
$$\begin{aligned} H(X,Y) &= -\sum_x \sum_y p_{XY}(x,y) \cdot \log p_{XY}(x,y) \\ &= -\sum_x \sum_y p_X(x) \cdot p_Y(y)(\log p_X(x) + \log p_Y(y)) \\ &= -\sum_x \left(\sum_y p_Y(y) \right) p_X(x) \cdot \log p_X(x) - \sum_y \left(\sum_x p_X(x) \right) p_Y(y) \cdot \log p_Y(y) \\ &= H(X) + H(Y) \end{aligned}$$

d) View $(X_1, \ldots, X_{n-1})$ as one random variable and make an induction proof.

1.4 a) The solution can for instance be found by using Lagrange multipliers. Set $p_i = p_Z(i),\ i = 1, \ldots, 4$. We want to maximize

$$-\sum_{i=1}^{4} p_i \cdot \log p_i$$

subject to the constraint

$$\sum_{i=1}^{4} p_i = 1$$

Thus, we want to maximize

$$-\sum_{i=1}^{4} p_i \cdot \log p_i + \lambda(\sum_{i=1}^{4} p_i - 1)$$

Differentiating with respect to p_j, we require

$$\frac{\partial}{\partial p_j}(-\sum_{i=1}^{4} p_i \cdot \log p_i + \lambda(\sum_{i=1}^{4} p_i - 1)) = 0$$

which gives us

$$-\log p_j - \frac{1}{\ln 2} + \lambda = 0$$

This shows us that all p_j should be equal, since they depend only on λ. Using the constraint $\sum p_i = 1$, we get that all $p_i = \frac{1}{4}$.

Thus, the distribution that gives the maximum entropy is the uniform distribution.

b) It's easy to see that the minimal entropy $H(Z) = 0$ can be achieved when one probability is 1 and the other are 0. There are four such distributions, and thus it's not unique.

1.5 a) Hint: $\displaystyle\sum_{i=0}^{\infty} q^i = \frac{1}{1-q}, \quad |q| < 1.$

b) Hint: $\displaystyle\sum_{i=0}^{\infty} i \cdot q^i = \frac{q}{(1-q)^2}, \quad |q| < 1.$

$$H(U) = \frac{-q \cdot \log q - (1-q) \cdot \log(1-q)}{1-q} = \frac{H_b(q)}{1-q}$$

c) $\displaystyle \mathrm{E}\{U\} = \frac{q}{1-q}$

1.6
$$-\ln 2\; I(X;Y) = \ln 2\Big(H(X \mid Y) - H(X)\Big) = \ln 2\Big(H(X,Y) - H(X) - H(Y)\Big)$$
$$= \mathrm{E}\left\{ \ln \frac{p(X)p(Y)}{p(X,Y)} \right\} \leq \mathrm{E}\left\{ \frac{p(X)p(Y)}{p(X,Y)} \right\} - 1 = 0$$
$$\text{since } \mathrm{E}\left\{ \frac{p(X)p(Y)}{p(X,Y)} \right\} = \sum_{x,y} p(x,y)\frac{p(x)p(y)}{p(x,y)} = 1.$$

1.7 a) $H + 1$

b) $H + p$

c) $2H$

1.8 a) $H(f(X)\,|\,X) \;=\; \big/ Z=f(X) \big/ \;=\; H(Z\,|\,X) =$

$$= -\sum_{b_j\in A_Z}\sum_{a_i\in A_X} \underbrace{p_{XZ}(a_i,b_j)}_{=0,f(a_i)\neq b_j} \log \underbrace{p_{Z|X}(b_j\,|\,a_i)}_{=1,f(a_i)=b_j}$$

$$= \; 0$$

b) $H(X,f(X)) \;=\; \big/\text{chain rule}\big/$

$$= \; H(X) + \underbrace{H(f(X)\,|\,X)}_{=0 \text{ according to a)}}$$

c) $H(X) - H(f(X)) \;=\; \big/\text{according to b)}\big/$

$$= \; H(X,f(X)) - H(f(X)) \;=\; \big/\text{chain rule}\big/$$

$$= \; H(X\,|\,f(X)) \;\geq\; 0$$

1.9 According to problem 1.8, applying a deterministic function to a random variable can not increase the entropy, which immediately gives us the right inequality.

For the left inequality we set $Z = X + Y$ and show that

$H(Z\,|\,Y) =$

$$= -\textstyle\sum_z\sum_y p_{ZY}(z,y)\ \log\ \frac{p_{ZY}(z,y)}{p_Y(y)} \;=\; \big/ x{=}z{-}y \big/$$

$$= \; -\textstyle\sum_x\sum_y p_{XY}(x,y)\ \log\ \frac{p_{XY}(x,y)}{p_Y(y)}$$

$$\stackrel{(1)}{=} \; -\textstyle\sum_x\sum_y p_X(x)p_Y(y)\ \log\ \frac{p_X(x)p_Y(y)}{p_Y(y)}$$

$$= \; -\Big(\textstyle\sum_y p_Y(y)\Big)\left(\sum_x p_X(x)\ \log\ p_X(x)\right) \;=\; H(X)$$

At (1) we used the independence. We can now write

$$H(X+Y) \;\geq\; H(X+Y\,|\,Y) \;=\; H(X)$$

See problem 1.6 for a proof that conditioning can't increase the entropy.

1.10 $H(X\,|\,Y) + H(Y\,|\,Z) \;\stackrel{(1)}{\geq}$

$$\geq \; H(X\,|\,Y,Z) + H(Y\,|\,Z)$$

$$= \; \underbrace{H(X\,|\,Y,Z) + H(Y\,|\,Z) + H(Z)}_{\text{chain rule } =H(X,Y,Z)} - H(Z)$$

$$= \; H(X,Y,Z) - H(Z)$$

$$\stackrel{(2)}{=} \; H(X\,|\,Z) + H(Y\,|\,XZ) \;\geq\; H(X\,|\,Z)$$

where we at (1) used the fact that conditioning never increases the entropy and at (2) we used the chain rule: $H(X,Y,Z) = H(Z) + H(X\,|\,Z) + H(Y\,|\,X,Z)$.

1.11 The chain rule is proved by

$$H(X,Y) \;=\; -\mathrm{E}\left\{\log p_{XY}(X,Y)\right\} \;=\; \mathrm{E}\left\{-\log p_{X|Y}(X|Y)p_Y(Y)\right\}$$

$$= \; \mathrm{E}\left\{-\log p_{X|Y}(X|Y)\right\} + \mathrm{E}\left\{-\log p_Y(Y)\right\} \;\stackrel{\triangle}{=}\; H(X\,|\,Y) \;+\; H(Y)$$

Now let $X = X_{n+1}$ and $Y = (X_1,\ldots,X_n)$.

1.12 a) Let (B_8, B_4, B_2, B_1) be four random variables describing the bits. Then the entropies will be

$$H(B_8) = -\frac{1}{3} \cdot \log \frac{1}{3} - \frac{2}{3} \cdot \log \frac{2}{3} \approx 0.9183$$
$$H(B_4) = -\frac{1}{3} \cdot \log \frac{1}{3} - \frac{2}{3} \cdot \log \frac{2}{3} \approx 0.9183$$
$$H(B_2) = -\frac{1}{2} \cdot \log \frac{1}{2} - \frac{1}{2} \cdot \log \frac{1}{2} = 1$$
$$H(B_1) = -\frac{1}{2} \cdot \log \frac{1}{2} - \frac{1}{2} \cdot \log \frac{1}{2} = 1$$

b)

$$H(X) = \log 12 \approx 3.5850$$

Note that this is smaller than the sum of the entropies for the different bits, since the bits are dependent of each other.

1.13 The transition matrix $\mathbf{P}$ of the source is

$$\mathbf{P} = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0.5 \end{pmatrix}$$

The stationary distribution $\bar{w} = (w_0,\ w_1,\ w_2)$ is given by the equation system $\bar{w} = \bar{w} \cdot \mathbf{P}$. Replace one of the equations with the the equation $w_0 + w_1 + w_2 = 1$ and solve the equation system. This gives us the solution

$$\bar{w} = \frac{1}{7}(5,\ 1,\ 1) \approx (0.714,\ 0.143,\ 0.143)$$

1.14 a) $H(X_i) = -\frac{5}{7} \cdot \log \frac{5}{7} - 2 \cdot \frac{1}{7} \cdot \log \frac{1}{7} \approx 1.1488$ [bits/symbol].

b) The block probabilities are given by $p(x_i, x_{i+1}) = p(x_i) \cdot p(x_{i+1}|x_i)$

symbol pair	probability
00	$5/7 \cdot 0.8 = 8/14$
01	$5/7 \cdot 0.1 = 1/14$
02	$5/7 \cdot 0.1 = 1/14$
10	$1/7 \cdot 0.5 = 1/14$
11	$1/7 \cdot 0.5 = 1/14$
12	0
20	$1/7 \cdot 0.5 = 1/14$
21	0
22	$1/7 \cdot 0.5 = 1/14$

$H(X_i, X_{i+1}) = -\frac{8}{14} \cdot \log \frac{8}{14} - 6 \cdot \frac{1}{14} \cdot \log \frac{1}{14} \approx 2.0931$ [bits/pair]. ($\rightarrow$ 1.0465 [bits/symbol]).

The entropy of pairs is less than twice the memory-less entropy.

c) Since the source is of order 1, the entropy rate is given by $H(X_i \,|\, X_{i-1})$. According to the chain rule we get

$H(X_i \,|\, X_{i-1}) = H(X_{i-1}, X_i) - H(X_{i-1}) \approx 0.9442$

1.15 The stationary distribution of the Markov source is

$$\bar{w} = [w_x \quad w_y \quad w_z] = \frac{1}{12}[3 \quad 4 \quad 5]$$

This gives us the memoryless entropy

$$H(X_n) = -\frac{3}{12} \cdot \log \frac{3}{12} - \frac{4}{12} \cdot \log \frac{4}{12} - \frac{5}{12} \cdot \log \frac{5}{12} \approx 1.5546$$

and the conditional entropy

$$\begin{aligned} H(X_n|X_{n-1}) &= \frac{3}{12}(-0.7 \cdot \log 0.7 - 0.2 \cdot \log 0.2 - 0.1 \cdot \log 0.1) + \\ & \quad \frac{4}{12}(-0.8 \cdot \log 0.8 - 0.1 \cdot \log 0.1 - 0.1 \cdot \log 0.1) + \\ & \quad \frac{5}{12}(-0.8 \cdot \log 0.8 - 0.1 \cdot \log 0.1 - 0.1 \cdot \log 0.1) \approx \\ &\approx 0.9806 \end{aligned}$$

From this we calculate

$$H(X_n, X_{n+1}, X_{n+2}, X_{n+3}, X_{n+4}) =$$
$$H(X_n) + H(X_{n+1}|X_n) + \ldots + H(X_{n+4}|X_n, X_{n+1}, X_{n+2}, X_{n+3}) =$$
$$H(X_n) + 4 \cdot H(X_{n+1}|X_n) \approx 5.4771$$

where we used the fact that source is of order 1 and stationary.

1.16 The transition matrix $\mathbf{P}$ of the source is

$$\mathbf{P} = \begin{pmatrix} 0.7 & 0 & 0.3 & 0 \\ 0.9 & 0 & 0.1 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0.2 & 0 & 0.8 \end{pmatrix}$$

The stationary distribution $\bar{w} = (w_{aa},\ w_{ab},\ w_{ba}, w_{bb})$ is given by the equation system $\bar{w} = \bar{w} \cdot \mathbf{P}$. Replace one of the equations with the equation $w_{aa} + w_{ab} + w_{ba} + w_{bb} = 1$ and solve the system. This gives us the solution

$$\bar{w} = \frac{1}{8}(3,\ 1,\ 1,\ 3)$$

The stationary distribution is of course also the probabilities for pairs of symbols, so we can directly calculate $H(X_i X_{i-1})$

$$H(X_i X_{i-1}) = -2 \cdot \frac{3}{8} \log \frac{3}{8} - 2 \cdot \frac{1}{8} \log \frac{1}{8} \approx 1.8113$$

The probabilities for single symbols are given by the marginal distribution

$$p(a) = p(aa) + p(ab) = 0.5, \qquad p(b) = p(ba) + p(bb) = 0.5$$

$$\Rightarrow \; H(X_i) = 1$$

The probabilities for three symbols are given by $p_{X_i X_{i-1} X_{i-2}} = p_{X_{i-1} X_{i-2}} \cdot p_{X_i | X_{i-1} X_{i-2}}$

$$\begin{aligned}
p(aaa) &= \frac{3}{8} \cdot 0.7 = \frac{21}{80}, \quad & p(baa) &= \frac{3}{8} \cdot 0.3 = \frac{9}{80} \\
p(aba) &= \frac{1}{8} \cdot 0.4 = \frac{4}{80}, \quad & p(bba) &= \frac{1}{8} \cdot 0.6 = \frac{6}{80} \\
p(aab) &= \frac{1}{8} \cdot 0.9 = \frac{9}{80}, \quad & p(bab) &= \frac{1}{8} \cdot 0.1 = \frac{1}{80} \\
p(abb) &= \frac{3}{8} \cdot 0.2 = \frac{6}{80}, \quad & p(bbb) &= \frac{3}{8} \cdot 0.8 = \frac{24}{80}
\end{aligned}$$

which gives us

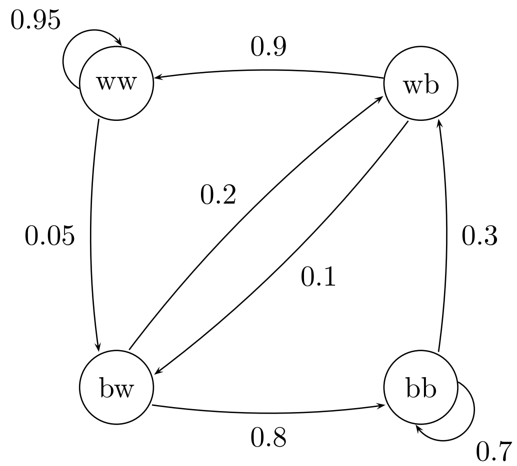
$$H(X_i X_{i-1} X_{i-2}) \approx 2.5925$$

Now we can calculate the conditional entropies using the chain rule

$$H(X_i | X_{i-1}) = H(X_i X_{i-1}) - H(X_{i-1}) \approx 0.8113$$

$$H(X_i | X_{i-1} X_{i-2}) = H(X_i X_{i-1} X_{i-2}) - H(X_{i-1} X_{i-2}) \approx 0.7812$$

1.17 a) Given states (x_i, x_{i-1}), the state diagram looks like



The stationary probabilities for this model is

$$w_{ww} = \frac{54}{68}, \; w_{wb} = \frac{3}{68}, \; w_{bw} = \frac{3}{68}, \; w_{bb} = \frac{8}{68}$$

These probabilities are also probabilities for pairs $p(x_i, x_{i-1})$.

b) The pair probabilities from above gives us the entropy $H(X_i, X_{i-1}) \approx 1.0246$. Probabilities for single symbols can be found as marginal probabilities

$$p(w) = p(w,w) + p(w,b) = \frac{57}{68}, \quad p(b) = p(b,w) + p(b,b) = \frac{11}{68}$$

which gives us the entropy $H(X_i) \approx 0.6385$. Using the chain rule, we find $H(X_i|X_{i-1}) = H(X_i, X_{i-1}) - H(X_{i-1}) \approx 0.3861$. Finally, we need probabilities for three symbols

$$p(x_i, x_{i-1}, x_{i-2}) = p(x_{i-1}, x_{i-2}) \cdot p(x_i|x_{i-1}, x_{i-2})$$

$$\begin{aligned}
p(w,w,w) &= \frac{513}{680}, \quad p(b,w,w) = \frac{27}{680} \\
p(w,w,b) &= \frac{27}{680}, \quad p(b,w,b) = \frac{3}{680} \\
p(w,b,w) &= \frac{6}{680}, \quad p(b,b,w) = \frac{24}{680} \\
p(w,b,b) &= \frac{24}{680}, \quad p(b,b,b) = \frac{56}{680}
\end{aligned}$$

This gives us the entropy $H(X_i, X_{i-1}, X_{i-2}) \approx 1.4083$.
Using the chain rule we find
$H(X_i|X_{i-1}, X_{i-2}) = H(X_i, X_{i-1}, X_{i-2}) - H(X_{i-1}, X_{i-2}) \approx 0.3837$.

1.18 a) We calculate the probabilities $p(x_n, x_{n-1})$ for pairs of symbols as marginal probabilities:

$$\begin{aligned}
p(aa) &= p(aaa) + p(aab) = 9/19 \\
p(ab) &= p(aba) + p(abb) = 2/19 \\
p(ba) &= p(baa) + p(bab) = 2/19 \\
p(bb) &= p(bba) + p(bbb) = 6/19
\end{aligned}$$

From these we can calculate the conditional probabilities $p(x_n|x_{n-1}, x_{n-2})$

$$p(a|aa) = \frac{p(aaa)}{p(aa)} = 0.8 \; ; \; p(b|aa) = \frac{p(baa)}{p(aa)} = 0.2$$

$$p(a|ab) = \frac{p(aab)}{p(ab)} = 0.9 \; ; \; p(b|ab) = \frac{p(bab)}{p(ab)} = 0.1$$

$$p(a|ba) = \frac{p(aba)}{p(ba)} = 0.1 \; ; \; p(b|ba) = \frac{p(bba)}{p(ba)} = 0.9$$

$$p(a|bb) = \frac{p(abb)}{p(bb)} = 0.3 \; ; \; p(b|bb) = \frac{p(bbb)}{p(bb)} = 0.7$$

The source can't be a Markov source of order 1, because $p(a|aa) \neq p(a|ab)$, for instance.

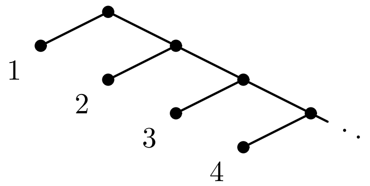
b) The best estimate we can make is to calculate the entropy $H(X_n|X_{n-1}X_{n-2})$ since we don't have access to more statistics.

$$H(X_n|X_{n-1}X_{n-2}) = \frac{9}{19} \cdot H_b(0.8) + (\frac{2}{19} + \frac{2}{19}) \cdot H_b(0.9) + \frac{6}{19} \cdot H_b(0.7) \approx 0.7190$$

2.1 Yes, since Kraft's inequality is fulfilled:

$$\sum_{i=1}^{8} 2^{-l_i} = \frac{53}{64} < 1$$

2.2 Since the probabilities are dyadic, it is possible to construct a code with the codeword lengths $l_i = -\log p(i) = i$. This code will have a rate that is equal to the entropy of the source and will thus be optimal.



$$R = 2 \text{ [bits/symbol]}.$$

2.3 a) The lowest rate is given by the entropy rate of the source. Since the source is memoryless, the entropy rate is

$$-0.6 \cdot \log 0.6 - 0.3 \cdot \log 0.3 - 0.1 \cdot \log 0.1 \approx 1.2955 \quad \text{[bits/symbol]}$$

b) Codeword lengths and one possible assignment of codewords:

symbol	length	codeword
x	1	0
y	2	10
z	2	11

The average codeword length is 1.4 bits/codeword and the average rate is 1.4 bits/symbol.

c) Codeword lengths (not unique) and one possible assignment of codewords:

symbol	length	codeword
xx	1	0
xy	3	100
xz	4	1100
yx	3	101
yy	4	1110
yz	5	11110
zx	4	1101
zy	6	111110
zz	6	111111

The average codeword length is 2.67 bits/codeword and the average rate is 1.335 bits/symbol.

2.4 Probabilities $p(x_i, x_{i+1}) = p(x_i) \cdot p(x_{i+1})$ for pairs of symbols:

$$p(a,a) = 0.25, \quad p(a,b) = 0.10, \quad p(a,c) = 0.10, \quad p(a,d) = 0.05$$

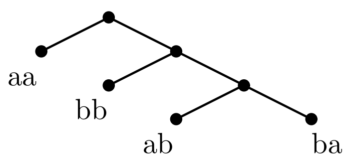
$$p(b,a) = 0.10, \quad p(b,b) = 0.04, \quad p(b,c) = 0.04, \quad p(b,d) = 0.02$$

$$p(c,a) = 0.10, \quad p(c,b) = 0.04, \quad p(c,c) = 0.04, \quad p(c,d) = 0.02$$

$$p(d,a) = 0.05, \quad p(d,b) = 0.02, \quad p(d,c) = 0.02, \quad p(d,d) = 0.01$$

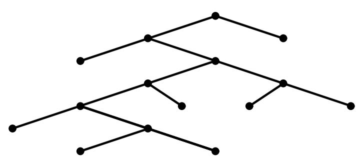
A Huffman code for this distribution gives the mean codeword length $\bar{l} = 3.57$ bits/codeword and average data rate $R = \frac{\bar{l}}{2} = 1.785$ bits/symbol.

2.5 Approximate probabilities for pairs of symbols: $\{0.4585, 0.0415, 0.0415, 0.4585\}$



$$\bar{l} \approx 1.6245 \text{ [bits/codeword]} \Rightarrow R = \frac{\bar{l}}{2} \approx 0.8122 \text{ [bits/symbol]}$$

Approximate probabilities for three symbols: {0.4204, 0.0381, 0.0034, 0.0381, 0.0381, 0.0034, 0.0381, 0.4204}



$$\bar{l} \approx 1.9496 \text{ [bits/codeword]} \Rightarrow R = \frac{\bar{l}}{3} \approx 0.6499 \text{ [bits/symbol]}$$

It is better to code three symbols at a time.

2.6 a) $p(r) = p^{r-1} \cdot (1-p)$

b) $\bar{r} = \sum_{r=1}^{\infty} r \cdot (1-p) \cdot p^{r-1} = \frac{1}{1-p} = \frac{1}{1-\sqrt[8]{0.5}} \approx 12.05 \quad \text{[symbols/run]}$

c)
$$\begin{aligned} H(R) &= -\sum_{r=1}^{\infty} (1-p) \cdot p^{r-1} \cdot \log((1-p) \cdot p^{r-1}) \\ &= -(1-p) \cdot \log(1-p) \cdot \sum_{r=1}^{\infty} p^{r-1} - (1-p) \cdot \log p \cdot \sum_{r=1}^{\infty} (r-1) \cdot p^{r-1} \end{aligned}$$

$$= -(1-p)\cdot\frac{1}{1-p}\cdot\log(1-p) - (1-p)\cdot\frac{p}{(1-p)^2}\cdot\log p$$
$$= \frac{H_b(p)}{1-p} \approx 4.972 \quad \text{[bits/run]}$$

d) $\frac{H(R)}{\bar{r}} = H_b(p) \approx 0.4126 \quad$ [bits/symbol]
We could of course get the same answer by calculating the entropy rate directly from the source.

2.7 a) If we do a direct mapping to the binary representation we get
$\bar{l} = \mathrm{E}\{\,\text{codeword length/run}\,\} =$
$$4\sum_{r=1}^{14}(1-p)p^{r-1} + 8\sum_{r=15}^{29}(1-p)p^{r-1} + 12\sum_{r=30}^{44}(1-p)p^{r-1} + \ldots =$$
$$4\sum_{r=1}^{\infty}(1-p)p^{r-1} + 4p^{14}\sum_{r=1}^{\infty}(1-p)p^{r-1} + 4p^{29}\sum_{r=1}^{\infty}(1-p)p^{r-1} + \ldots =$$
$$4\sum_{r=1}^{\infty}(1-p)p^{r-1}(1+p^{14}\sum_{i=0}^{\infty}p^{15i}) = 4\cdot(1+\frac{p^{14}}{1-p^{15}}) \approx 5.635 \quad \text{[bits/run]}$$

From problem 2.5 we know that $\bar{r} \approx 12.05$ [symbols/run], therefore the data rate is $R = \frac{\bar{l}}{\bar{r}} \approx 0.468$ [bits/symbol].

b) In the same way as we did in a) we get
$$\bar{l} = 5\cdot(1+\frac{p^{30}}{1-p^{31}}) \approx 5.399 \quad \text{[bits/run]}$$

and the data rate $R \approx 0.448$ [bits/symbol].

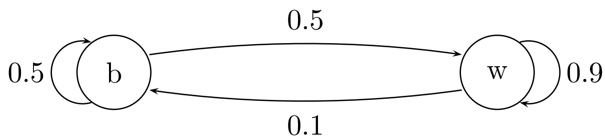
2.8 a) Since runs of a and b have the same probabilities, we can use the same Golomb code for both types of runs. Since the probability of a run of length r is $p(r) = p^{r-1}\cdot(1-p)$ we get an optimal code if we choose m as
$$m = \lceil -\frac{1}{\log p} \rceil = 8$$

b) A Golomb code with parameter $m = 8$ has 8 codewords of length 4, 8 codewords of length 5, 8 codewords of length 6 etc.

The average codeword length is
$\bar{l} = \mathrm{E}\{\,\text{codeword length/run}\,\} =$
$$4\sum_{r=1}^{8}(1-p)p^{r-1} + 5\sum_{r=9}^{16}(1-p)p^{r-1} + 6\sum_{r=17}^{24}(1-p)p^{r-1} + \ldots =$$
$$4\sum_{r=1}^{\infty}(1-p)p^{r-1} + p^{8}\sum_{r=1}^{\infty}(1-p)p^{r-1} + p^{16}\sum_{r=1}^{\infty}(1-p)p^{r-1} + \ldots = 4 + \sum_{i=1}^{\infty}p^{8i} =$$
$$4 + \frac{p^8}{1-p^8} = 5 \quad \text{[bits/run]}$$

From problem 2.6 we know that $\bar{r} \approx 12.05$ [symbols/run], therefore the data rate is $R = \frac{\bar{l}}{\bar{r}} \approx 0.415$ [bits/symbol].

2.9 a) The theoretical lowest bound on the data rate is given by the entropy rate of the source. The best model of the source that we can get, according to the given information, is a Markov model of order 1. In the text areas the model looks like



The stationary probabilities for this source are $w_b = \frac{1}{6}$ and $w_w = \frac{5}{6}$ and the entropy rate when we are in a text area is therefore

$$H_t = w_b \cdot H_b(0.5) + w_w \cdot H_b(0.9) \approx 0.5575$$

In the same way, for image areas the entropy rate is

$$H_p \approx 0.7857$$

The total entropy rate for the source is therefore

$$H = \frac{4}{5}H_t + \frac{1}{5}H_p \approx \underline{0.60313}$$

This is the best estimate we can make. The real entropy rate of the source can be lower, if the memory is longer than just one pixel.

b) The desired data rate can be achieved by coding blocks of three symbols. The probabilities for the 8 different combinations of three symbols can be calculated from $p_{X_i X_{i+1} X_{i+2}} = p_{X_i} \cdot p_{X_{i+1}|X_i} \cdot p_{X_{i+2}|X_i X_{i+1}}$

The eight probabilities are

$$\frac{1}{120}\{1 \quad 5 \quad 5 \quad 5 \quad 5 \quad 9 \quad 9 \quad 81\}$$

Construct the code using the Huffman algorithm. The resulting data rate is $\underline{0.625}$ bits/pixel.

2.10 a) It is possible to get arbitrarily close to the entropy rate of the source. For the given source, the entropy rate is given by $H(S_{n+1}|S_n) = w_A H(S_{n+1}|S_n = A) + w_B H(S_{n+1}|S_n = B) + w_C H(S_{n+1}|S_n = C) + w_D H(S_{n+1}|S_n = D)$

where w_A etc. are the stationary probabilities for the states. These are calculated from the following equation system, plus the fact that they should sum to

1.

$$\begin{pmatrix} w_A & w_B & w_C & w_D \end{pmatrix} \begin{pmatrix} 0 & 0.5 & 0 & 0.5 \\ 0 & 0.2 & 0 & 0.8 \\ 0.1 & 0 & 0.9 & 0 \\ 0 & 0 & 0.7 & 0.3 \end{pmatrix} = \begin{pmatrix} w_A & w_B & w_C & w_D \end{pmatrix}$$

$\Rightarrow \quad (w_A \; w_B \; w_C \; w_D) = \frac{1}{731}(56 \; 35 \; 560 \; 80)$

The entropy rate is

$$\begin{aligned} H(S_{n+1}|S_n) &= \frac{1}{731}(56 \cdot H_b(0.5) + 35 \cdot H_b(0.2) + 560 \cdot H_b(0.1) + 80 \cdot H_b(0.3)) \\ &\approx 0.567 \quad \text{[bits/symbol]} \end{aligned}$$

b) Optimal tree codes can be constructed using Huffman's algorithm. When we code single symbols we use the stationary probabilities. For example we can get the following code

symbol	probability	codeword	length
A	56/731	110	3
B	35/731	111	3
C	560/731	0	1
D	80/731	10	2

which gives a data rate of $\frac{993}{731} \approx 1.36$ bits/symbol

The probabilities of pairs of symbols are gotten from $p_{X_i X_{i+1}} = p_{X_i} \cdot p_{X_{i+1}|X_i}$. For instance we can get the following code

symbols	probability	codeword	length
AB	28/731	1010	4
AD	28/731	1110	4
BB	7/731	10110	5
BD	28/731	1111	4
CA	56/731	100	3
CC	504/731	0	1
DC	56/731	110	3
DD	24/731	10111	5

The resulting data rate is $\frac{1}{2} \cdot \frac{1331}{731} \approx 0.910$ bits/symbol

2.11

$$F(0) = 0, \; F(1) = 0.6, \; F(2) = 0.8, \; F(3) = 1$$

$$
\begin{aligned}
l^{(0)} &= 0 \\
u^{(0)} &= 1 \\
\\
l^{(1)} &= 0 + (1-0) \cdot 0.8 = 0.8 \\
u^{(1)} &= 0 + (1-0) \cdot 1 = 1 \\
\\
l^{(2)} &= 0.8 + (1-0.8) \cdot 0 = 0.8 \\
u^{(2)} &= 0.8 + (1-0.8) \cdot 0.6 = 0.92 \\
\\
l^{(3)} &= 0.8 + (0.92-0.8) \cdot 0 = 0.8 \\
u^{(3)} &= 0.8 + (0.92-0.8) \cdot 0.6 = 0.872 \\
\\
l^{(4)} &= 0.8 + (0.872-0.8) \cdot 0.6 = 0.8432 \\
u^{(4)} &= 0.8 + (0.872-0.8) \cdot 0.8 = 0.8576
\end{aligned}
$$

The interval corresponding to the sequence is $[0.8432,\ 0.8576)$. The size of the interval is 0.0144, which means that we need to use at least $\lceil -\log 0.0144 \rceil = 7$ bits in the codeword.

Alternative 1: The smallest number with 7 bits in the interval is $(0.1101100)_2 = 0.84375$. Since $(0.1101101)_2 = 0.8515625$ is also in the interval 7 bits will be enough, and the codeword is **1101100**.

Alternative 2: The middle point of the interval is $0.8504 = (0.110110011\ldots)_2$. Truncate to 7+1=8 bits, which gives us the codeword **11011001**.

2.12 With six bits all values are stored as 1/64:ths. The distribution is then

$$F(0) = 0,\ F(1) = 38,\ F(2) = 51,\ F(3) = 64$$

$$
\begin{aligned}
l^{(0)} &= 0 = (000000)_2 \\
u^{(0)} &= 63 = (111111)_2 \\
\\
l^{(1)} &= 0 + \lfloor \frac{(63-0+1) \cdot 51}{64} \rfloor = 51 = (110011)_2 \\
u^{(1)} &= 0 + \lfloor \frac{(63-0+1) \cdot 64}{64} \rfloor - 1 = 63 = (111111)_2
\end{aligned}
$$

Shift out 1 to the codeword, shift a 0 into l and a 1 into u

$$
\begin{aligned}
l^{(1)} &= (100110)_2 = 38 \\
u^{(1)} &= (111111)_2 = 63
\end{aligned}
$$

Shift out 1 to the codeword, shift a 0 into l and a 1 into u

$$\begin{aligned} l^{(1)} &= (001100)_2 = 12 \\ u^{(1)} &= (111111)_2 = 63 \end{aligned}$$

$$\begin{aligned} l^{(2)} &= 12 + \lfloor \frac{(63-12+1)\cdot 0}{64} \rfloor = 12 = (001100)_2 \\ u^{(2)} &= 12 + \lfloor \frac{(63-12+1)\cdot 38}{64} \rfloor - 1 = 41 = (101001)_2 \end{aligned}$$

$$\begin{aligned} l^{(3)} &= 12 + \lfloor \frac{(41-12+1)\cdot 0}{64} \rfloor = 12 = (001100)_2 \\ u^{(3)} &= 12 + \lfloor \frac{(41-12+1)\cdot 38}{64} \rfloor - 1 = 28 = (011100)_2 \end{aligned}$$

Shift out 0 to the codeword, shift a 0 into l and a 1 into u

$$\begin{aligned} l^{(3)} &= (011000)_2 = 24 \\ u^{(3)} &= (111001)_2 = 57 \end{aligned}$$

$$\begin{aligned} l^{(4)} &= 24 + \lfloor \frac{(57-24+1)\cdot 38}{64} \rfloor = 44 = (101100)_2 \\ u^{(4)} &= 24 + \lfloor \frac{(57-24+1)\cdot 51}{64} \rfloor - 1 = 50 = (110010)_2 \end{aligned}$$

Since there are no more symbols we don't need to do any more shift operations. The codeword are the bits that have been shifted out before, plus all of $l^{(4)}$, ie **110101100**.

2.13 With six bits all values are stored as 1/64:ths. The distribution is then

$$F(0) = 0, \; F(1) = 38, \; F(2) = 51, \; F(3) = 64$$

This means that the interval 0-37 belongs to symbol a_1, the interval 38-50 to symbol a_2 and the interval 51-63 to symbol a_3.

$$\begin{aligned} l^{(0)} &= (000000)_2 = 0 \\ u^{(0)} &= (111111)_2 = 63 \\ t &= (101110)_2 = 46 \end{aligned}$$

$$\lfloor \frac{(46-0+1)\cdot 64 - 1}{63-0+1} \rfloor = 46 \Rightarrow a_2$$

$$\begin{aligned} l^{(1)} &= 0 + \lfloor \frac{(63-0+1)\cdot 38}{64} \rfloor = 38 = (100110)_2 \\ u^{(1)} &= 0 + \lfloor \frac{(63-0+1)\cdot 51}{64} \rfloor - 1 = 50 = (110010)_2 \end{aligned}$$

Shift out 1, shift 0 into l, 1 into u and a new bit from the codeword into t.

$$\begin{aligned} l^{(1)} &= (001100)_2 = 12 \\ u^{(1)} &= (100101)_2 = 37 \\ t &= (011101)_2 = 29 \end{aligned}$$

$$\lfloor \frac{(29-12+1)\cdot 64 - 1}{37-12+1} \rfloor = 44 \Rightarrow a_2$$

$$\begin{aligned} l^{(2)} &= 12 + \lfloor \frac{(37-12+1)\cdot 38}{64} \rfloor = 27 = (011011)_2 \\ u^{(2)} &= 12 + \lfloor \frac{(37-12+1)\cdot 51}{64} \rfloor - 1 = 31 = (011111)_2 \end{aligned}$$

The first three bits are the same in l and u. Shift them out, shift zeros into l, ones into u and three new bits from the codeword into t.

$$\begin{aligned} l^{(2)} &= (011000)_2 = 24 \\ u^{(2)} &= (111111)_2 = 63 \\ t &= (101010)_2 = 42 \end{aligned}$$

$$\lfloor \frac{(42-24+1)\cdot 64 - 1}{63-24+1} \rfloor = 30 \Rightarrow a_1$$

$$\begin{aligned} l^{(3)} &= 24 + \lfloor \frac{(63-24+1)\cdot 0}{64} \rfloor = 24 = (011000)_2 \\ u^{(3)} &= 24 + \lfloor \frac{(63-24+1)\cdot 38}{64} \rfloor - 1 = 46 = (101110)_2 \end{aligned}$$

The first two bits in l are 01 and the first two in u are 10. Shift l, u and t one step, shift 0 into l, 1 into l and a new bit from the codeword into t. Invert the new most significant bits of l, u and t.

$$\begin{aligned} l^{(3)} &= (010000)_2 = 16 \\ u^{(3)} &= (111101)_2 = 61 \\ t &= (110100)_2 = 52 \end{aligned}$$

$$\lfloor \frac{(52-16+1)\cdot 64 - 1}{61-16+1} \rfloor = 51 \Rightarrow a_3$$

$$\begin{aligned} l^{(4)} &= 16 + \lfloor \frac{(61-16+1)\cdot 51}{64} \rfloor = 52 = (110100)_2 \\ u^{(4)} &= 16 + \lfloor \frac{(61-16+1)\cdot 64}{64} \rfloor - 1 = 61 = (111101)_2 \end{aligned}$$

The first two bits are the same in l and u. Shift them out, shift zeros into l, ones into u and two new bits from the codeword into t.

$$\begin{aligned} l^{(4)} &= (010000)_2 = 16 \\ u^{(4)} &= (110111)_2 = 55 \\ t &= (010000)_2 = 16 \end{aligned}$$

$$\lfloor\frac{(16-16+1)\cdot 64-1}{55-16+1}\rfloor = 1 \Rightarrow a_1$$

Since we have now decoded five symbols we don't have to do any more calculations. The decoded sequence is

$$a_2 a_2 a_1 a_3 a_1$$

2.14 a) Under the assumption that we have ordered the colours in the same order as in the table, with white closest to 0, the interval is $[0.05 \;\; 0.07538)$.

b) green, green

2.15 The probabilities for single symbols $p(x_n)$ can be found as the marginal distribution, ie

$$p(a) = p(aa) + p(ab) = 2/7, \quad p(b) = p(ba) + p(bb) = 5/7$$

The conditional probabilities $p(x_{n+1}|x_n)$ are

$$p(a|a) = \frac{p(aa)}{p(a)} = \frac{1/7}{2/7} = 0.5$$

$$p(b|a) = \frac{p(ab)}{p(a)} = \frac{1/7}{2/7} = 0.5$$

$$p(a|b) = \frac{p(ba)}{p(b)} = \frac{1/7}{5/7} = 0.2$$

$$p(b|b) = \frac{p(bb)}{p(b)} = \frac{4/7}{5/7} = 0.8$$

The interval corresponding to the sequence is $[0.424, 0.488)$

The interval size is 0.064, which means we need at least $\lceil -\log 0.064 \rceil = 4$ bits in our codeword.

Alternative 1: The smallest number with 4 bits in the interval is $(0.0111)_2 = 0.4375$. Since $(0.1000)_2 = 0.5$ is not inside the interval 4 bits will not be enough, instead we must use 5 bits. We will use the number $(0.01110)_2 = 0.4375$ and thus the codeword is **01110**.

Alternative 2: The middle point of the interval is $0.456 = (0.0111010\ldots)_2$ Truncate to 4+1=5 bits, which gives us the codeword **01110**.

2.16 Assuming that we always place the b interval closest to 0, the sequence corresponds to the interval $[0.078253 \;\; 0.088)$. The interval size is 0.009747 and thus we will need at least $\lceil -\log_2 0.009747 \rceil = 7$ bits in our codeword, maybe one more.

Write the lower limit as a binary number:
$0.078253 = (0.0001010000001\ldots)_2$

The smallest binary number with seven bits that is larger than the lower limit is $(0.0001011)_2 = 0.0859375$.

Since $(0.0001100)_2 = 0.09375 > 0.088$, seven bits will not be enough to ensure that we have a prefix code. Thus we will have to use eight bits.

The codeword is thus: 00010101

If we instead place the w interval closest to 0, the interval for the sequence becomes $[0.912 \quad 0.921747)$ and the codeword becomes 11101010

2.17 a) In a block of length n, there can be between 0 and n ones. w ones can be placed in $\binom{n}{w}$ different ways in a block of length n. Thus we will need $\lceil \log(n+1) \rceil + \lceil \log \binom{n}{w} \rceil$ bits to code the block.

b) To code how many ones there are, we need 4 bits (we need to send a number between 0 and 15). The sequence has 2 ones. There are $\binom{15}{2} = 105$ different ways to place two ones in a block of length 15. Thus, we need $\lceil \log 105 \rceil = 7$ bits to code how the 2 ones are placed. In total we need $4 + 7 = 11$ bits to code the given sequence.

c) Assume that the symbol probabilities are p and $1-p$. The entropy rate of the source is $-p \cdot \log p - (1-p) \cdot \log(1-p) = H_b(p)$.

The number of bits/symbol is given by

$$R = \frac{1}{n}\sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w (\lceil \log(n+1) \rceil + \lceil \log \binom{n}{w} \rceil) <$$

$$\frac{1}{n}(2 + \log(n+1) + \sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log \binom{n}{w})$$

where we used the fact that $\lceil x \rceil < x + 1$

$$\sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log \binom{n}{w} =$$

$$\sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log(\binom{n}{w} p^{n-w}(1-p)^w) - \quad (1)$$

$$\sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log p^{n-w} - \quad (2)$$

$$\sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log(1-p)^w \quad (3)$$

(1) is the negative entropy of a binomial distribution, thus it is ≤ 0.

$$(3) = \sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w \log(1-p)^w =$$

$$\log(1-p) \sum_{w=0}^{n} \binom{n}{w} p^{n-w}(1-p)^w w =$$

$$n \cdot \log(1-p) \cdot (1-p) \sum_{w=1}^{n} \binom{n-1}{w-1} p^{n-w}(1-p)^{w-1} = n \cdot (1-p) \cdot \log(1-p)$$

where we used the fact that the last sum is just the sum of a binomial distribution, which of course is 1. In the same way we can show that $(2) = n \cdot p \cdot \log p$. Since the source is memoryless, the entropy $H_b(p)$ is a lower

bound for R. Thus we get
$H_b(p) \leq R < \frac{1}{n}(2 + \log(n+1) - n \cdot p \cdot \log p - n \cdot (1-p) \cdot \log(1-p)) =$
$H_b(p) + \frac{1}{n}(2 + \log(n+1))$
So when $n \to \infty$ the rate R will approach the entropy of the source, showing that the coding method is universal.

2.18 Assuming offset 0 at the right of the history buffer

Codewords:

(offset, length, new symbol)	Binary codeword
(0,0,a)	0000 0000 0
(0,0,b)	0000 0000 1
(1,2,b)	0001 0010 1
(2,1,a)	0010 0001 0
(6,5,b)	0110 0101 1
(8,6,a)	1000 0110 0
(1,4,b)	0001 0100 1
(15,3,a)	1111 0011 0

2.19 The coded sequence of pairs <index, new symbol> are:

$$< 0, a > < 0, b > < 1, b > < 2, a > < 1, a > < 4, b > < 2, b >$$
$$< 4, a > < 3, a > < 5, a > < 5, b > < 3, b > \ \ldots$$

If we assume that the dictionary is of size 16 we will need 4+1=5 bits to code each pair.

The dictionary now looks like

index	sequence	index	sequence	index	sequence
0	-	5	aa	10	aaa
1	a	6	bab	11	aab
2	b	7	bb	12	abb
3	ab	8	baa		
4	ba	9	aba		

2.20 The coded sequence of <index> is:

$$< 0 > < 1 > < 2 > < 3 > < 0 > < 2 > < 4 > < 1 >$$
$$< 5 > < 7 > < 6 > < 12 > < 3 > < 9 > \ \ldots$$

If we assume that the dictionary is of size 16 we will need 4 bits to code each index.

The dictionary now looks like

index	sequence	index	sequence	index	sequence
0	a	6	aa	12	aaa
1	b	7	aba	13	$aaab$
2	ab	8	$abbb$	14	bab
3	ba	9	bb	15	bba
4	abb	10	$baaa$		
5	baa	11	$abaa$		

2.21 The decoded sequence is

$$fafafafedededdggg\ldots$$

and the dictionary looks like

index	word	index	word	index	word	index	word
0	a	5	f	10	faf	15	edd
1	b	6	g	11	$fafe$	16	dg
2	c	7	h	12	ed	17	gg
3	d	8	fa	13	de	18	
4	e	9	af	14	ede	19	

and the next word to add to position 18 is $gg*$, where $*$ will be the first symbol in the next decoded word.

2.22 The history buffer size will be $2^6 = 64$. The alphabet size is 8, thus we will use $\log_2 8 = 3$ bits to code a symbol. If we code a single symbol we will use a total of $1+3=4$ bits and if we code a match we will use a total of $1+6+3=10$ bits. This means that it is better to code matches of length 1 and 2 as single symbols. Since we use 3 bits for the lengths, we can then code match lengths between 3 and 10.

f	o	l	c	codeword	sequence
0			b	0 001	b
0			a	0 000	a
0			g	0 110	g
1	1	6		1 000001 011	agagag
0			e	0 100	e
1	9	3		1 001001 000	bag
1	4	4		1 000100 001	geba
0			d	0 011	d
0			h	0 111	h
0			a	0 000	a
0			d	0 011	d
0			e	0 100	e
1	3	6		1 000011 011	hadeha
0			f	0 101	f

2.23 a) The transformed sequence is *dddabbbaac* and the index is 9 (assuming that the first index is 0).

b) After mtf the sequence is 3,0,0,1,2,0,0,1,0,3.

2.24 Inverse mtf gives the sequence $yxzzzzxzzyyy$.
Inverse BWT gives the sequence $xyzzyzyzzyx$ (assuming that the first index is 0).

3.1 $h(X) = \frac{1}{2} \log ea^2 = \frac{1}{2} \log 6e\sigma^2$

3.2 $h(X) = \log e\lambda = \frac{1}{2} \log e^2\sigma^2$

3.3 $$R(D) = \begin{cases} B \log \frac{2\sigma^2\sqrt{2}}{3D} & ; 0 \leq D \leq \frac{2\sigma^2}{3} \\ \frac{1}{2} B \log \frac{2\sigma^2}{3D-\sigma^2} & ; \frac{2\sigma^2}{3} \leq D \leq \sigma^2 \\ 0 & ; D > \sigma^2 \end{cases}$$

3.4 $$R(D) = \begin{cases} \frac{B}{\ln 2}(\ln \frac{1}{1-\sqrt{1-\frac{D}{\sigma^2}}} - \sqrt{1 - \frac{D}{\sigma^2}}) & ; 0 \leq D \leq \sigma^2 \\ 0 & ; D > \sigma^2 \end{cases}$$

3.5 We use the rate-distortion function to calculate the theoretically lowest distortion. The process has the variance

$$\sigma^2 = \int_{-0.5}^{0.5} \Phi(\theta)\ d\theta = 17 \cdot 0.6 = 10.2$$

The rate-distortion function is

$$R(D) = 0.6 \cdot \frac{1}{2} \log \frac{17 \cdot 0.6}{D} = 0.3 \cdot \log \frac{\sigma^2}{D} \quad ; \quad 0 \leq D \leq \sigma^2$$

The minimum distortion for $R = 3$ is

$$D = \sigma^2 \cdot 2^{-3/0.3} = \frac{10.2}{1024} \approx 0.009961$$