MT341/441/5441 Channels

Mark Wildon, mark.wildon@rhul.ac.uk

Administration:

- Sign-in sheet. **Please return to the lecturer after each lecture.**
- Make sure you get the printed notes for the introduction **at the end of this lecture** (remind me if I forget).
- **Please take a clicker and use it!**
- All handouts will be put on Moodle. The first marked problem sheet will be on Moodle by Wednesday.
- **Lectures:** Monday 2pm–4pm (McCrea 0-04), Thursday 9am (BLT2)
- **Drop-in times in McCrea LGF025:** Tuesday 3.30pm, Wednesday 11am, Thursday 11.30am.

§1 Introduction

Example 1.1

A friend has chosen a number x between 0 and 15. How many 'Yes'/'No' questions do you need to find x?

(A) ≤ 3 (B) 4 (C) 5 (D) > 5

Exercise 1.2

Is there a questioning strategy that can guarantee to use three or fewer questions?

(A) No (B) Yes

Can you prove that your answer is correct?

§1 Introduction

Example 1.1

A friend has chosen a number x between 0 and 15. How many 'Yes'/'No' questions do you need to find x?

(A) ≤ 3 (B) 4 (C) 5 (D) > 5

Exercise 1.2

Is there a questioning strategy that can guarantee to use three or fewer questions?

(A) No (B) Yes

Can you prove that your answer is correct?

Binary number s

One simple strategy uses binary. If

$$x = 2^{m-1}x_{m-1} + \cdots + 2x_1 + x_0.$$

then we say that x is $x_{m-1} \ldots x_1 x_0$ in binary, and write, for example, $13 = 01101 = 1101 = \ldots$. (You can write 1101_2 if you want to emphasise the base 2 of binary.) The *bi*nary digi*ts* 0 and 1 are called *bits*.

x	binary form	x	binary form
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

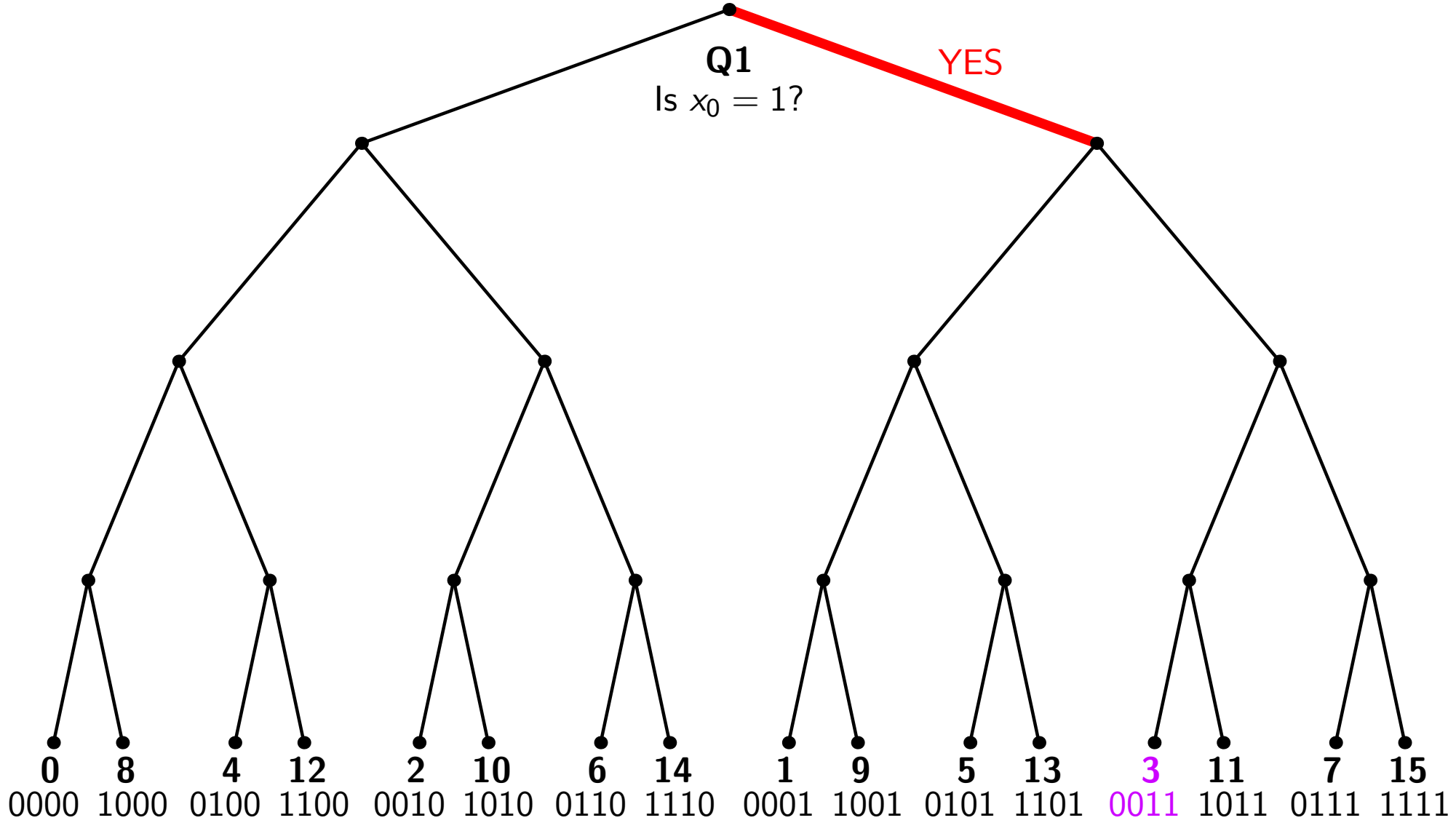
Binary Questioning Strategy

- The binary form of 0, 1, . . . , 15 gives an easy four question strategy: just ask one question about each bit of your friend's number in turn.

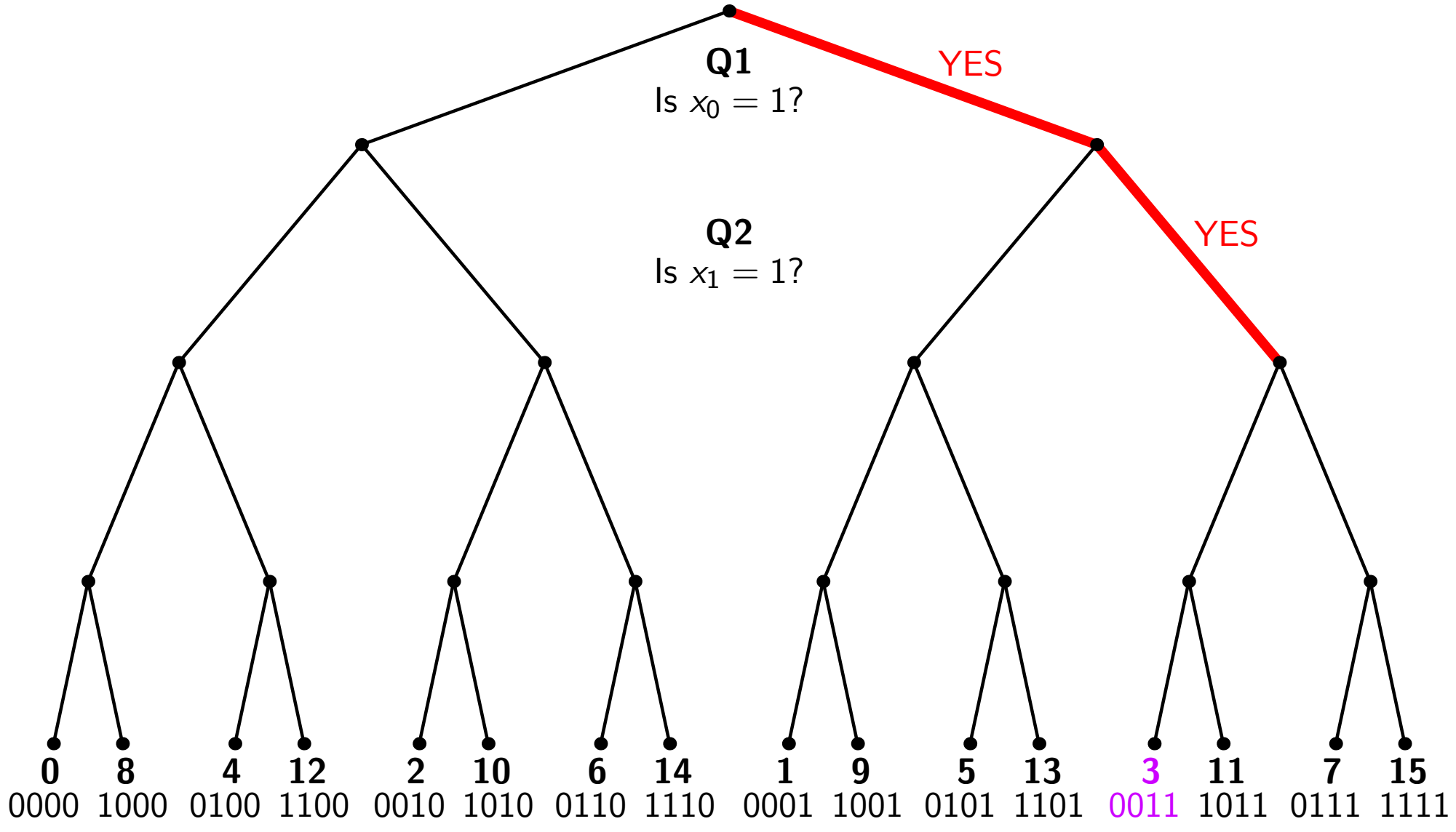
Binary Questioning Strategy

- The binary form of 0, 1, . . . , 15 gives an easy four question strategy: just ask one question about each bit of your friend's number in turn.
- It also solves Exercise 1.2: there is no way to learn four bits by asking three questions!

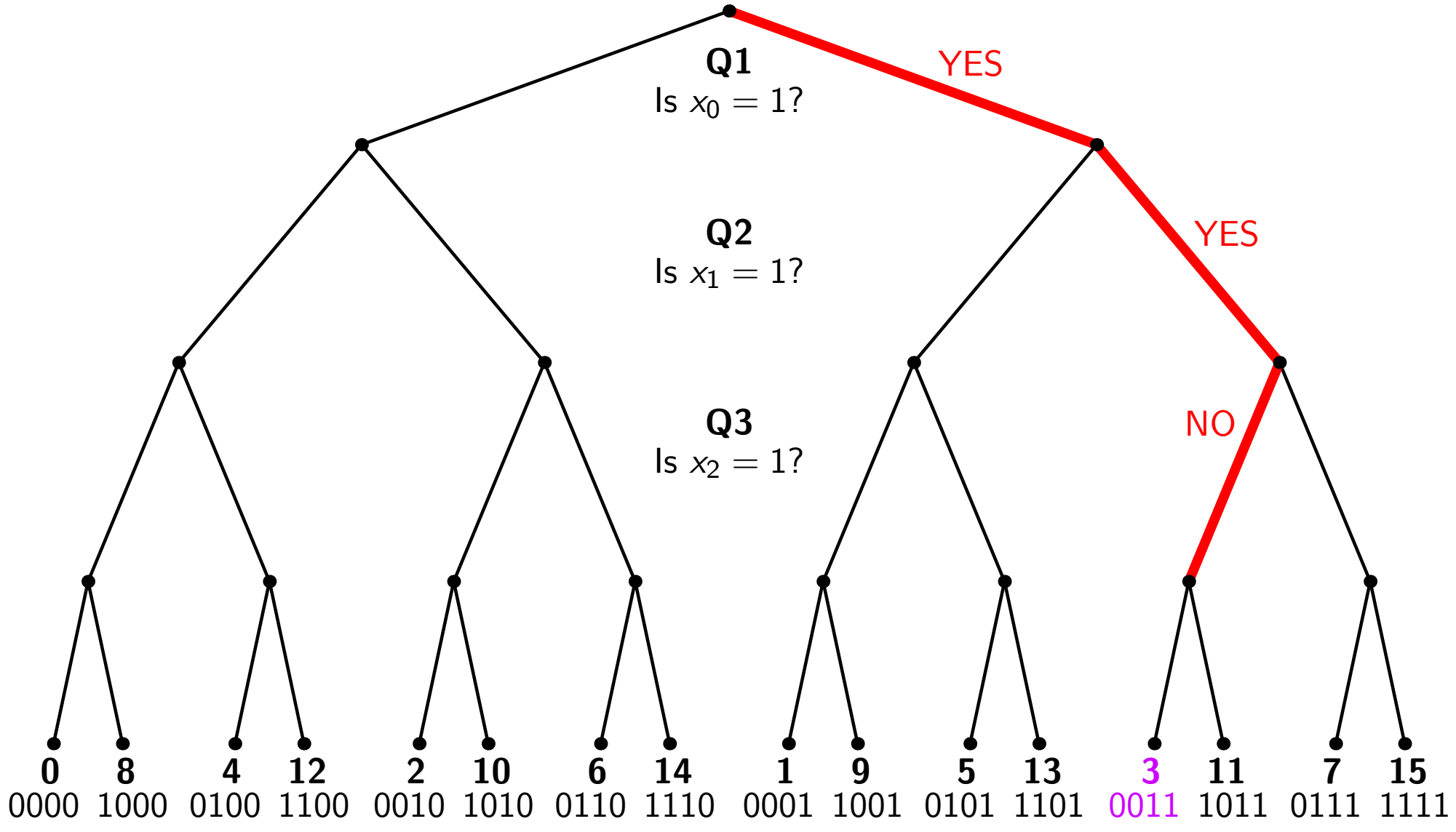
4 Yes/No Questions for 4 Bits of Information



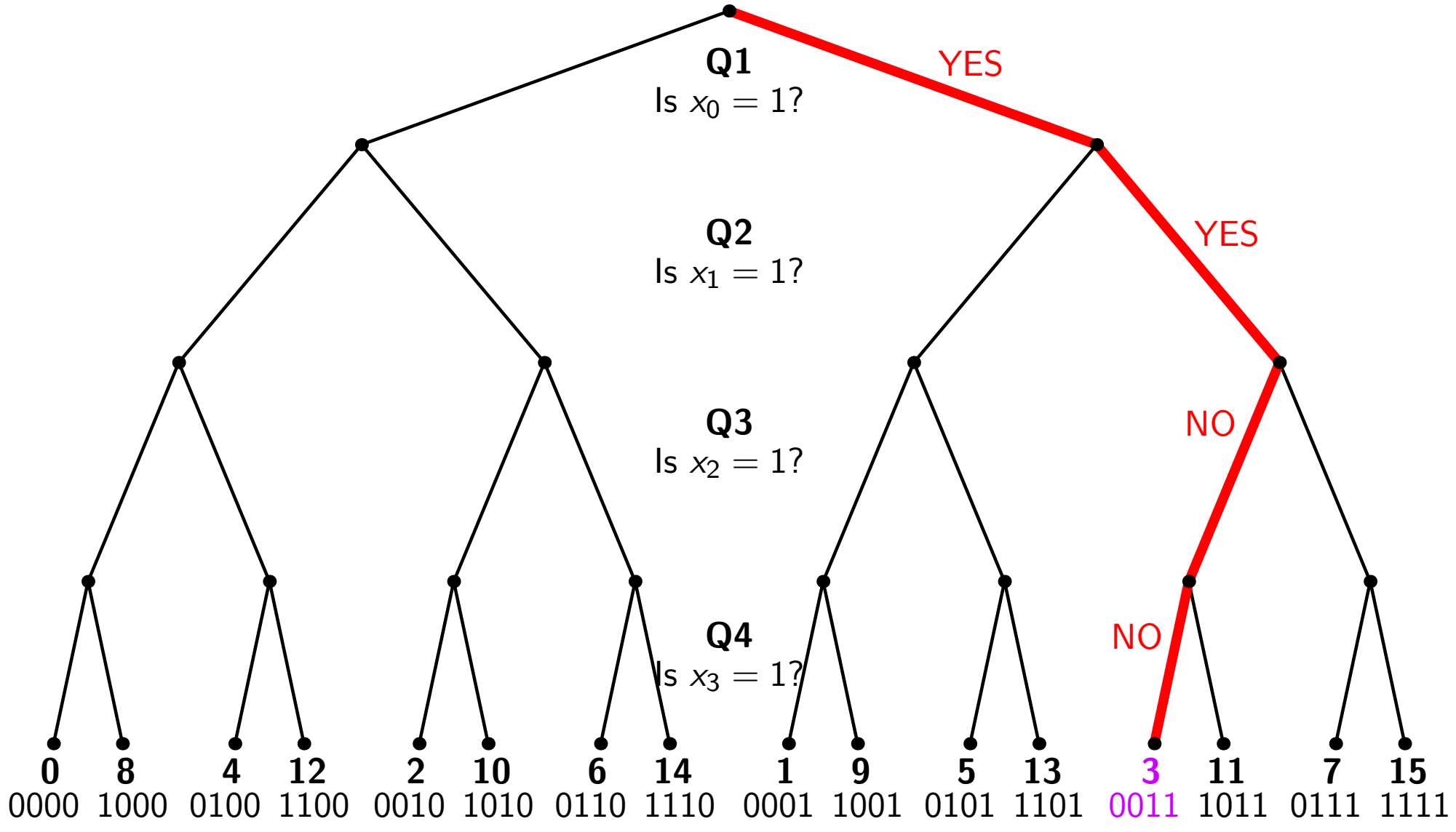
4 Yes/No Questions for 4 Bits of Information



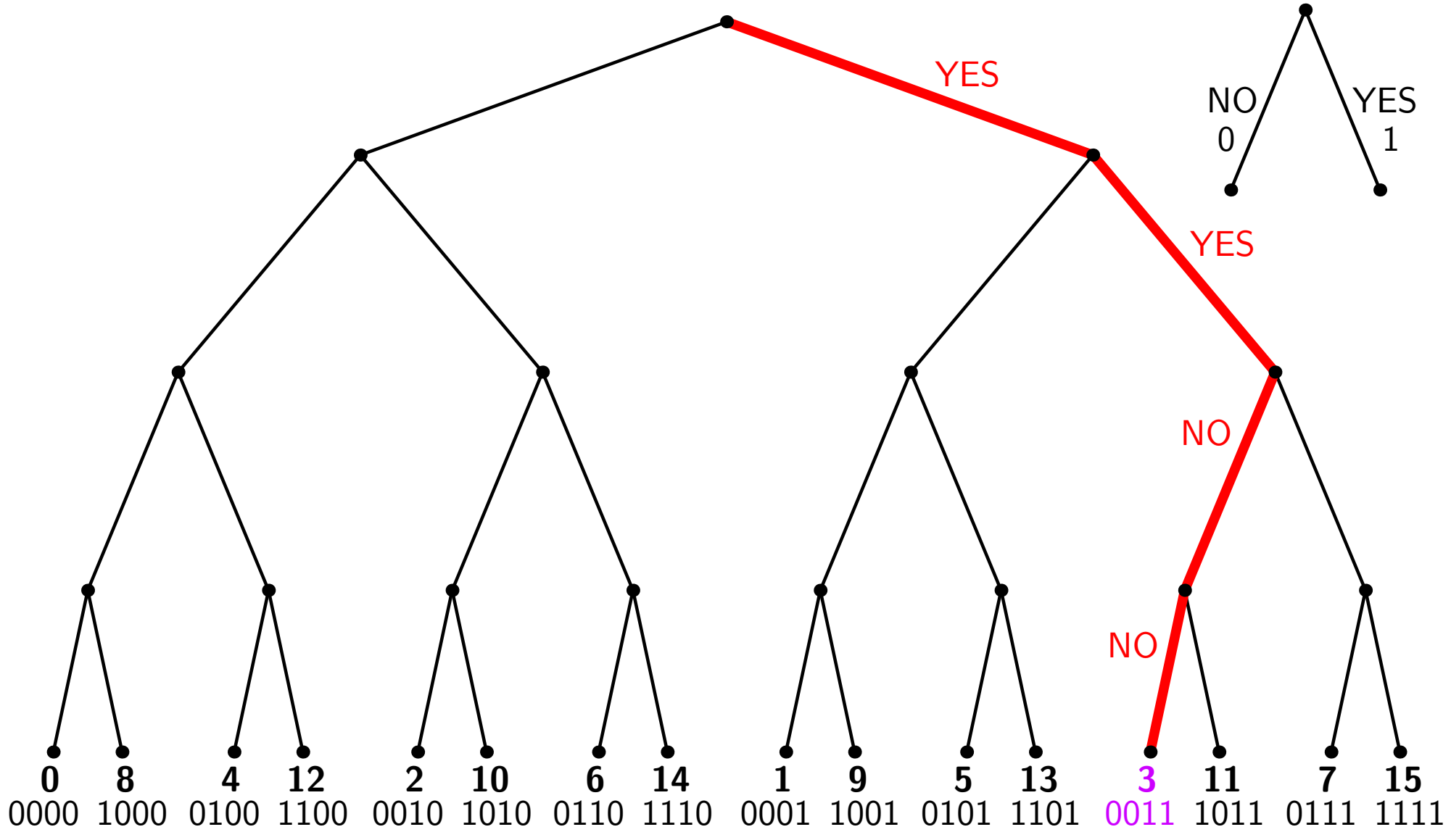
4 Yes/No Questions for 4 Bits of Information



4 Yes/No Questions for 4 Bits of Information



4 Yes/No Questions for 4 Bits of Information



Exercise on Binary

Exercise 1.3

Say that $x \in \mathbb{N}_0$ has *length m in binary* if its binary form is $x_{m-1} \ldots x_1 x_0$ with $x_{m-1} = 1$. For instance $35 = 100011_2$ has length 6.

(a) Which numbers x have length *at most m*?

(b) How many questions would you need if the game in Example 1.1 was changed so that $x \in \{0, 1, 2, \ldots, 99\}$?

(A) 6 (B) 7 (C) 8 (D) can't be sure

(c) Which numbers x have length exactly m?

Exercise on Binary

Exercise 1.3

Say that $x \in \mathbb{N}_0$ has *length m in binary* if its binary form is $x_{m-1} \ldots x_1 x_0$ with $x_{m-1} = 1$. For instance $35 = 100011_2$ has length 6.

(a) Which numbers x have length *at most m*?

(b) How many questions would you need if the game in Example 1.1 was changed so that $x \in \{0, 1, 2, \ldots, 99\}$?

(A) 6 (B) 7 (C) 8 (D) can't be sure

(c) Which numbers x have length exactly m?

Random Messages

We will see in this course that the bit is the fundamental unit of information. To make this rigorous we need to bring in ideas of randomness. For example, a number in $\{0, 1, 2, \ldots, 15\}$, chosen uniformly at random, has exactly 4 bits of information.

Exercise 1.4

Suppose that your friend's number is 0 with probability $\frac{1}{2}$, and each of 1, ..., 15 with equal probability $\frac{1}{30}$. Suggest a good questioning strategy. How many questions does it use on average? What is the corresponding encoder?

Random Messages

We will see in this course that the bit is the fundamental unit of information. To make this rigorous we need to bring in ideas of randomness. For example, a number in $\{0, 1, 2, \ldots, 15\}$, chosen uniformly at random, has exactly 4 bits of information.

Exercise 1.4

Suppose that your friend's number is 0 with probability $\frac{1}{2}$, and each of 1, ..., 15 with equal probability $\frac{1}{30}$. Suggest a good questioning strategy. How many questions does it use on average? What is the corresponding encoder?

The encoder defined by $0 \mapsto 1$, $1 \mapsto 0000$, $2 \mapsto 00010$, $3 \mapsto 00011$, ..., $14 \mapsto 01110$, $15 \mapsto 01111$ has the shortest possible expected length of codewords for the probability distribution $\frac{1}{2}, \frac{1}{30}, \ldots, \frac{1}{30}$. We will prove this in Part A, as a corollary of the optimality of Huffman codes.

Example 1.5

The expected length for the encoder $0 \mapsto 1$, $1 \mapsto 0000$, $2 \mapsto 00010$, $3 \mapsto 00011$, $\ldots$, $14 \mapsto 01110$, $15 \mapsto 01111$ is

$$\frac{1}{2} \times 1 + \frac{1}{30} \times 4 + \frac{14}{30} \times 5 = \frac{15 + 4 + 70}{30} = \frac{89}{30} = 3 - \frac{1}{30}.$$

Does this contradict Exercise 1.2?

(A) No (B) Yes

Exercise 1.6

Suppose that multiple numbers are encoded using this encoder by concatenating the codewords. You receive

$$10111\,10000\,11101\,10111.$$

Quiz: the sent number are 0, 15, 1, 0, 0, 0, 13, 0, 0

(A) False (B) True

Why can you be sure?

Exercise 1.7

Suppose that messages A, T, G, C have probabilities $\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$. Suggest an efficient binary code $u(A), u(T), u(G), u(C)$.

Example 1.5

The expected length for the encoder $0 \mapsto 1$, $1 \mapsto 0000$, $2 \mapsto 00010$, $3 \mapsto 00011$, $\ldots$, $14 \mapsto 01110$, $15 \mapsto 01111$ is

$$\frac{1}{2} \times 1 + \frac{1}{30} \times 4 + \frac{14}{30} \times 5 = \frac{15 + 4 + 70}{30} = \frac{89}{30} = 3 - \frac{1}{30}.$$

Does this contradict Exercise 1.2?

(A) No (B) Yes

Exercise 1.6

Suppose that multiple numbers are encoded using this encoder by concatenating the codewords. You receive

$$10111\,10000\,11101\,10111.$$

Quiz: the sent number are 0, 15, 1, 0, 0, 0, 13, 0, 0

(A) False (B) True

Why can you be sure?

Exercise 1.7

Suppose that messages A, T, G, C have probabilities $\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$. Suggest an efficient binary code $u(A), u(T), u(G), u(C)$.

Example 1.5

The expected length for the encoder $0 \mapsto 1$, $1 \mapsto 0000$, $2 \mapsto 00010$, $3 \mapsto 00011$, $\ldots$, $14 \mapsto 01110$, $15 \mapsto 01111$ is

$$\frac{1}{2} \times 1 + \frac{1}{30} \times 4 + \frac{14}{30} \times 5 = \frac{15 + 4 + 70}{30} = \frac{89}{30} = 3 - \frac{1}{30}.$$

Does this contradict Exercise 1.2?

(A) No (B) Yes

Exercise 1.6

Suppose that multiple numbers are encoded using this encoder by concatenating the codewords. You receive

$$10111\,10000\,11101\,10111.$$

Quiz: the sent number are 0, 15, 1, 0, 0, 0, 13, 0, 0

(A) False (B) True

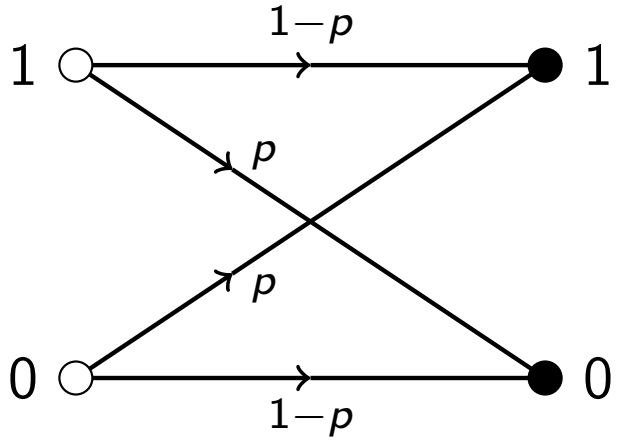
Why can you be sure?

Exercise 1.7

Suppose that messages A, T, G, C have probabilities $\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$. Suggest an efficient binary code $u(A), u(T), u(G), u(C)$.

Noisy Channel Coding

Suppose that Alice wants to send Bob a single 'Yes/No' message. She can only communicate by sending Bob the bits 0 and 1 through the *binary symmetric channel* with *cross-over probability* p, or $\mathrm{BSC}(p)$, that flips each bit with probability p.



Exercise 1.8

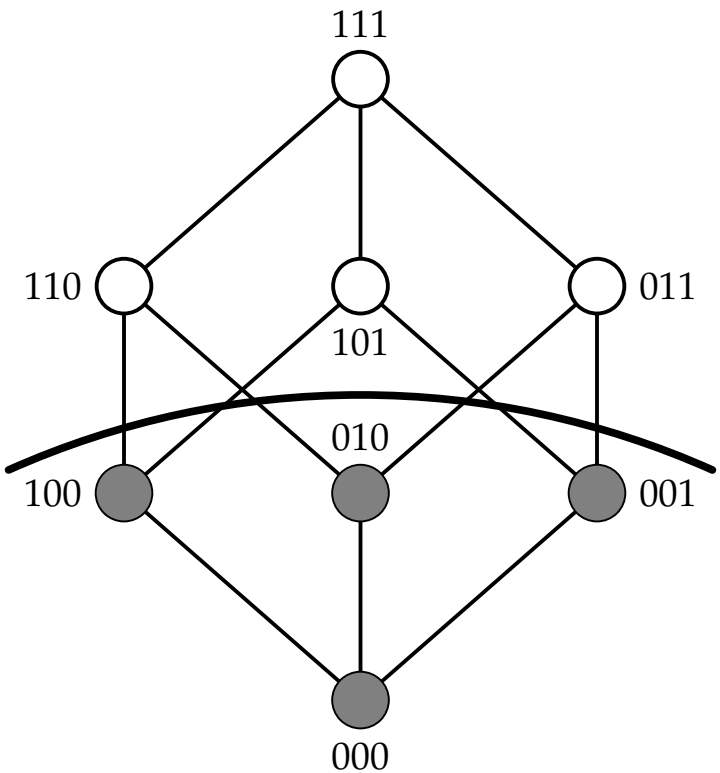
Why may we, and Alice and Bob, assume that $p < \frac{1}{2}$?

If Alice encodes her 'Yes'/'No' message by a single bit, then with probability p, Bob will not receive the intended message.

Repetition Codes

Instead Alice and Bob decide to pad out the single bit with some redundant bits using the binary repetition code of length 3, with codewords 000 and 111. The agreed encoder is 'No' $\mapsto$ 000 and 'Yes' $\mapsto$ 111. Bob decodes by assuming that the majority bit in a received word is correct. So

- 000, 001, 010, 100 are decoded as 000 meaning 'No';
- 111, 110, 101, 011 are decoded as 111 meaning 'Yes'.



Probability Calculations

Let X be Alice's sent codeword. Let Y be Bob's received word.

$$\mathbb{P}[Y = 000|X = 000] = (1-p)^3$$
$$\mathbb{P}[Y = 110|X = 000] = p^2(1-p).$$

Informally $\mathbb{P}[A|B]$ is 'the probability of event A, given that event B has occurred'.

Exercise 1.9

Find $\mathbb{P}[Y = 111|X = 000]$. What is

$$\mathbb{P}[Y \in \{111, 110, 101, 011\}|X = 000]?$$

(A) $3p^2(1-p)$ (B) $3p^2 - 2p^3$ (C) $3p^2 - p^3$ (D) can't say

The second probability is $\mathbb{P}$[Bob decodes as 111|Alice sends 000].

(a) Why is this equal to $\mathbb{P}$[Bob decodes wrongly]?

(b) Is this an improvement on Alice sending a single bit 0 or 1 to Bob?

(c) How does this probability change if instead Alice and Bob use the binary repetition code of length 5?

Probability Calculations

Let X be Alice's sent codeword. Let Y be Bob's received word.

$$\mathbb{P}[Y = 000|X = 000] = (1-p)^3$$
$$\mathbb{P}[Y = 110|X = 000] = p^2(1-p).$$

Informally $\mathbb{P}[A|B]$ is 'the probability of event A, given that event B has occurred'.

Exercise 1.9

Find $\mathbb{P}[Y = 111|X = 000]$. What is

$$\mathbb{P}[Y \in \{111, 110, 101, 011\}|X = 000]?$$

(A) $3p^2(1-p)$ (B) $3p^2 - 2p^3$ (C) $3p^2 - p^3$ (D) can't say

The second probability is $\mathbb{P}$[Bob decodes as 111|Alice sends 000].

(a) Why is this equal to $\mathbb{P}$[Bob decodes wrongly]?

(b) Is this an improvement on Alice sending a single bit 0 or 1 to Bob?

(c) How does this probability change if instead Alice and Bob use the binary repetition code of length 5?

ISG Research Seminar

"We're All Happily Married Here!": Intimate Partner Violence as a Cybersecurity Issue

- Thu, 03 Oct 2019 11:00
- Shilling 0-04
- **Julia Slupska** (University of Oxford)

All welcome, M.Sc. students are particularly encouraged to attend. Thursday at 11am is the regular time.

Administration

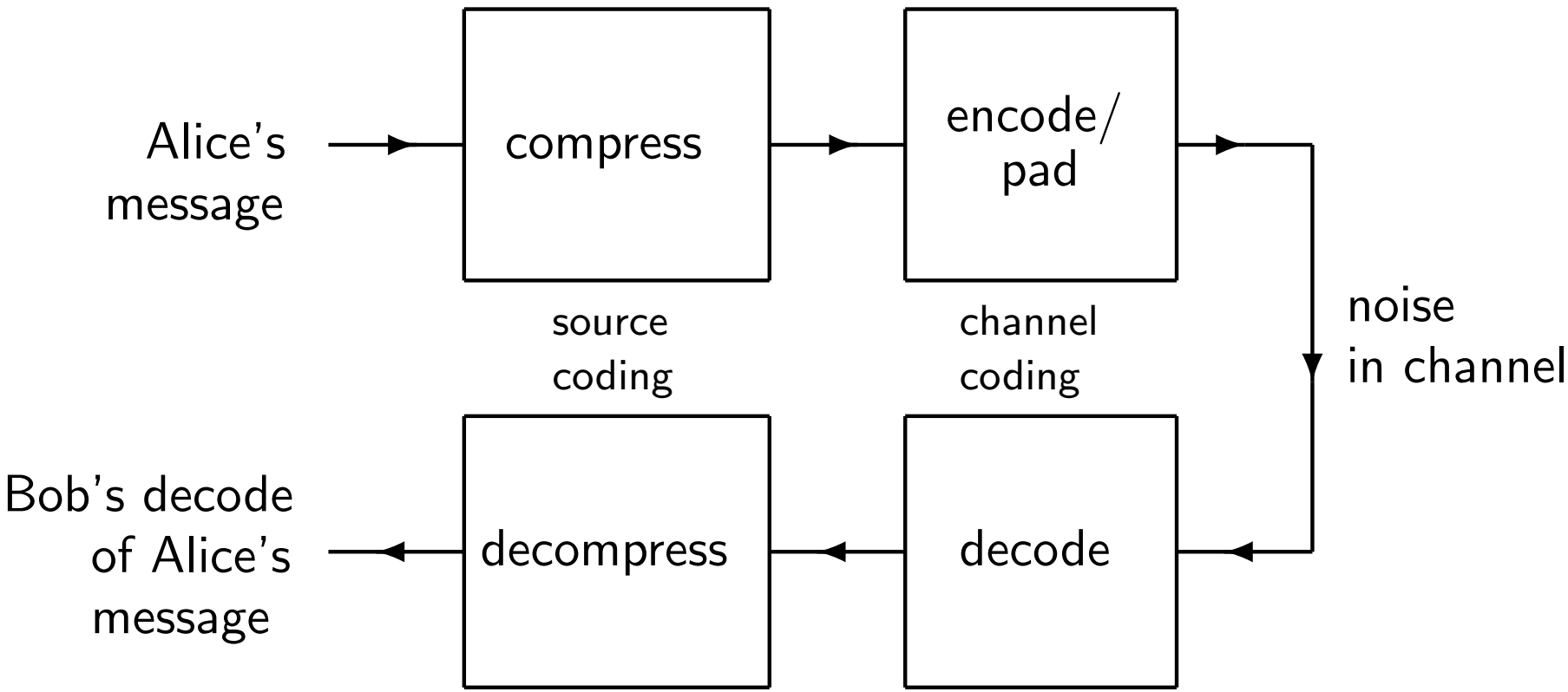
- Please take Problem Sheet 1
- Please take Part A printed notes pages 11 to 14
- Please take a clicker and use it

Channel Coding

The aim of channel coding is to minimize the (general version) of $\mathbb{P}$[Bob decodes wrongly]. In the Alice and Bob example with the 'Yes'/'No' message there are only two codewords, and repetition codes are optimal. In Part B we shall see the much richer theory when there are many messages and codewords.

The bigger picture

The diagram below shows how source coding and channel coding combine. Source coding *removes redundancy*. For example in Exercise 1.7 you replaced 8 bit ASCII with a much shorter code. Channel coding *adds redundancy*, in a controlled way to minimize the probability of a decoding error. Repetition codes are the simplest example.



Source and Channel Combined

Exercise 1.10

In Exercise 1.7 you solved the source coding problem for the messages A, T, G, C with probabilities $\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$.

(a) Using the binary repetition code of length 3 as the channel code, what would you send through the BSC to communicate $CCTGC$?

(b) Give an example of a received binary word and how it is decoded.

(c) How is redundancy removed and added in this process?

Source and Channel Combined

- MATHEMATICA demonstration using `ImageNoise.nb` to send black and white pictures through the BSC using repetition code.
- All MATHEMATICA notebooks will be put on Moodle.

Probability Revision

Definition 1.11

- A *probability measure* p on a finite set Ω assigns a real number p_ω to each $\omega \in \Omega$ so that $0 \leq p_\omega \leq 1$ for each ω and

$$\sum_{\omega \in \Omega} p_\omega = 1.$$

  We say that p_ω is the *probability of* ω.

- A *probability space* is a finite set Ω equipped with a probability measure. The elements of Ω are called *outcomes*.

- An *event* is a subset of Ω.

- The *probability* of an event $A \subseteq \Omega$, denoted $\mathbb{P}[A]$, is the sum of the probability of the outcomes in A; that is

$$\mathbb{P}[A] = \sum_{\omega \in A} p_\omega.$$

Dice Example

Example 1.12

(1) To model a throw of a single unbiased die, we take

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

and put $p_\omega = \frac{1}{6}$ for each outcome $\omega \in \Omega$. The event that we throw an even number is $A = \{2, 4, 6\}$ and as expected, $\mathbb{P}[A] = p_2 + p_4 + p_6 = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$.

(2) To model a throw of a pair of dice we could take

$$\Omega = \{1, 2, 3, 4, 5, 6\} \times \{1, 2, 3, 4, 5, 6\}$$

and give each element of Ω probability $\frac{1}{36}$. Alternatively, if we know we only care about the sum of the two dice, we could take $\Omega = \{2, 3, \ldots, 12\}$ with $p_2 = 1/36$, $p_3 = 2/36$, $\ldots$, $p_6 = 5/36$, $p_7 = 6/36$, $p_8 = 5/36$, $\ldots$, $p_{12} = 1/36$. The former is natural and more flexible.

Conditional Probability

Definition 1.13

Let Ω be a probability space, and let A, $B \subseteq \Omega$ be events. If $\mathbb{P}[B] \neq 0$ then we define the *conditional probability of A given B* by

$$\mathbb{P}[A|B] = \frac{\mathbb{P}[A \cap B]}{\mathbb{P}[B]}.$$

The events A, B are *independent* if $\mathbb{P}[A \cap B] = \mathbb{P}[A]\mathbb{P}[B]$.

Suppose that each element of Ω has equal probability p. Then

$$\mathbb{P}[A|B] = \frac{|A \cap B|p}{|B|p} = \frac{|A \cap B|}{|B|}$$

is the proportion of elements of B that also lie in A. This agrees with the intuitive idea that $\mathbb{P}[A|B]$ is the probability that, given B has occurred, then A has also occurred.

Exercise 1.14

Let A and B be events in a probability space such that $\mathbb{P}[B] \neq 0$. Show that $\mathbb{P}[A|B] = \mathbb{P}[A]$ if and only if A and B are independent.

Conditional Probability is Subtle

Exercise 1.15

Let $\Omega = \{HH, HT, TH, TT\}$ be the probability space for two flips of a fair coin, so each outcome has probability $\frac{1}{4}$. Let A be the event that both flips are heads, and let B be the event that at least one flip is a head. Write A and B as subsets of Ω.

Quiz: What is $\mathbb{P}[A|B]$?

(A) 1/3 (B) 1/2 (C) 2/3 (D) need more information

Example 1.16 (The Monty Hall Problem)

On a game show you are offered the choice of three doors. Behind one door is a car, and behind the other two are goats. You pick a door and then the host, *who knows where the car is*, opens another door to reveal a goat. You may then either open your original door, or change to the remaining unopened door. Assuming you want the car, should you change?

(A) No (B) Yes

Conditional Probability is Subtle

Exercise 1.15

Let $\Omega = \{HH, HT, TH, TT\}$ be the probability space for two flips of a fair coin, so each outcome has probability $\frac{1}{4}$. Let A be the event that both flips are heads, and let B be the event that at least one flip is a head. Write A and B as subsets of Ω.

Quiz: What is $\mathbb{P}[A|B]$?

(A) 1/3 (B) 1/2 (C) 2/3 (D) need more information

Example 1.16 (The Monty Hall Problem)

On a game show you are offered the choice of three doors. Behind one door is a car, and behind the other two are goats. You pick a door and then the host, *who knows where the car is*, opens another door to reveal a goat. You may then either open your original door, or change to the remaining unopened door. Assuming you want the car, should you change?

(A) No (B) Yes

Conditional Probability is Subtle

Exercise 1.15

Let $\Omega = \{HH, HT, TH, TT\}$ be the probability space for two flips of a fair coin, so each outcome has probability $\frac{1}{4}$. Let A be the event that both flips are heads, and let B be the event that at least one flip is a head. Write A and B as subsets of Ω.

Quiz: What is $\mathbb{P}[A|B]$?

(A) 1/3 (B) 1/2 (C) 2/3 (D) need more information

Example 1.16 (The Monty Hall Problem)

On a game show you are offered the choice of three doors. Behind one door is a car, and behind the other two are goats. You pick a door and then the host, *who knows where the car is*, opens another door to reveal a goat. You may then either open your original door, or change to the remaining unopened door. Assuming you want the car, should you change?

(A) No (B) Yes

Random Variables and Expectation

Definition 1.17

Let Ω be a probability space. A *random variable* on Ω is a function $X : \Omega \to \mathcal{R}$, for some set $\mathcal{R}$.

Often $\mathcal{R}$ will be the set $\mathbb{R}$ of real numbers. But it will be useful in this course to allow other sets: for instance in Exercise 1.9, X and Y took values in $\{000, 001, 010, 100, 011, 101, 110, 111\}$.

Definition 1.18

Let Ω be a probability space with probability measure p. The *expectation* $\mathbb{E}[X]$ of a random variable $X : \Omega \to \mathbb{R}$ is defined to be

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega) p_w.$$

Intuitively, the expectation of X is the average value of X on elements of Ω, if we choose $\omega \in \Omega$ with probability p_ω. We have

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} X(\omega) p_\omega = \sum_{x \in R} \sum_{\substack{\omega \\ X(\omega)=x}} x p_\omega = \sum_{x \in \mathbb{R}} x \mathbb{P}[X = x].$$

Linearity of Expectation

A critical property of expectation is that it is linear. Note that we *do not assume any independence* in this lemma. The proof is left as an exercise.

Lemma 1.19 (Linearity of expectation)

Let Ω be a probability space. If $X_1, X_2, \ldots, X_k : \Omega \to \mathbb{R}$ are random variables then

$$\mathbb{E}[a_1 X_1 + a_2 X_2 + \cdots + a_k X_k] = a_1 \mathbb{E}[X_1] + a_2 \mathbb{E}[X_2] + \cdots + a_k \mathbb{E}[X_k]$$

for any $a_1, a_2, \ldots, a_k \in \mathbb{R}$.

Quiz: What is the expectation of a single die?

(A) 3 (B) $3\frac{1}{2}$ (C) 4 (D) depends on the roll

Four dice are rolled. What is the expectation of their sum?

(A) 12 (B) 13 (C) 14 (D) 15

Linearity of Expectation

A critical property of expectation is that it is linear. Note that we *do not assume any independence* in this lemma. The proof is left as an exercise.

Lemma 1.19 (Linearity of expectation)

Let Ω be a probability space. If $X_1, X_2, \ldots, X_k : \Omega \to \mathbb{R}$ are random variables then

$$\mathbb{E}[a_1 X_1 + a_2 X_2 + \cdots + a_k X_k] = a_1 \mathbb{E}[X_1] + a_2 \mathbb{E}[X_2] + \cdots + a_k \mathbb{E}[X_k]$$

for any $a_1, a_2, \ldots, a_k \in \mathbb{R}$.

Quiz: What is the expectation of a single die?

(A) 3 (B) $3\frac{1}{2}$ (C) 4 (D) depends on the roll

Four dice are rolled. What is the expectation of their sum?

(A) 12 (B) 13 (C) 14 (D) 15

Linearity of Expectation

A critical property of expectation is that it is linear. Note that we *do not assume any independence* in this lemma. The proof is left as an exercise.

Lemma 1.19 (Linearity of expectation)

Let Ω be a probability space. If $X_1, X_2, \ldots, X_k : \Omega \to \mathbb{R}$ are random variables then

$$\mathbb{E}[a_1 X_1 + a_2 X_2 + \cdots + a_k X_k] = a_1 \mathbb{E}[X_1] + a_2 \mathbb{E}[X_2] + \cdots + a_k \mathbb{E}[X_k]$$

for any $a_1, a_2, \ldots, a_k \in \mathbb{R}$.

Quiz: What is the expectation of a single die?

(A) 3 (B) $3\frac{1}{2}$ (C) 4 (D) depends on the roll

Four dice are rolled. What is the expectation of their sum?

(A) 12 (B) 13 (C) 14 (D) 15

Variance and Chebyshev's Inequality

Definition 1.20

Let Ω be a probability space. The *variance* $\mathbf{Var}[X]$ of a random variable $X : \Omega \to \mathbb{R}$ is defined to be

$$\mathbf{Var}[X] = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big].$$

The variance measures how much X can be expected to depart from its mean value $\mathbb{E}[X]$. So it is a measure of the 'spread' of X. This is made precise by Chebyshev's inequality.

Lemma 1.21

If X is a random variable and $a > 0$ then

$$\mathbb{P}\big[|X - \mathbb{E}X| \geq a\big] \leq \frac{\mathbf{Var}[X]}{a^2}.$$

We shall use Chebyshev's inequality later in the course. A proof is outlined on the first problem sheet.

Reliable communication

In Exercise 1.9 we saw that

$$\mathbb{P}[\text{Bob decodes as } 111 | \text{Alice sent } 000] \approx 0.028$$

and this equals $\mathbb{P}[\text{Bob decodes wrongly}]$. By using a longer repetition code this probability can be made arbitrarily small.

Exercise 1.23

Suppose that Alice's message is 'No' with probability $\frac{1}{100}$ and 'Yes' with probability $\frac{99}{100}$ and that the cross-over probability in the BSC is $\frac{1}{10}$. Taking $p = \frac{1}{10}$ in Question 5 on Problem Sheet 1 you should find:

$$\mathbb{P}[\text{Bob decodes as 'No'} | \text{Alice's message is 'No'}] = 0.972$$

$$\mathbb{P}[\text{Alice's message is 'No'} | \text{Bob decodes as 'No'}] \approx 0.260$$

Bob receives 001. Should he believe that Alice's message is 'No'?

(A) No he shouldn't (B) Yes he should

See the handout and Question 7 (optional) on Problem Sheet 1 for an analogy with medical testing.

Reliable communication

In Exercise 1.9 we saw that

$$\mathbb{P}[\text{Bob decodes as } 111 | \text{Alice sent } 000] \approx 0.028$$

and this equals $\mathbb{P}[\text{Bob decodes wrongly}]$. By using a longer repetition code this probability can be made arbitrarily small.

Exercise 1.23

Suppose that Alice's message is 'No' with probability $\frac{1}{100}$ and 'Yes' with probability $\frac{99}{100}$ and that the cross-over probability in the BSC is $\frac{1}{10}$. Taking $p = \frac{1}{10}$ in Question 5 on Problem Sheet 1 you should find:

$$\mathbb{P}[\text{Bob decodes as 'No'} | \text{Alice's message is 'No'}] = 0.972$$

$$\mathbb{P}[\text{Alice's message is 'No'} | \text{Bob decodes as 'No'}] \approx 0.260$$

Bob receives 001. Should he believe that Alice's message is 'No'?

(A) No he shouldn't (B) Yes he should

See the handout and Question 7 (optional) on Problem Sheet 1 for an analogy with medical testing.

§2 Prefix-free Binary Codes and Kraft's Inequality

Question. What properties should a binary code have so that we can decode it easily? How do these restrict its codewords?

Recall that if $\mathcal{A}$ is a set then $\mathcal{A}^\ell$ is the set of all ℓ-tuples of elements of $\mathcal{A}$. For example $\mathbb{R}^3 = \{(x, y, z) : x, y, z \in \mathbb{R}\}$ is 3-dimensional space.

In this course we number positions in tuples from 1 as usual, so z is the 3rd coordinate in (x, y, z).

Quiz: One of these statements is false. Which one?

(A) $\{1, 2, 3, 3\}$ is a set of size 3 and it's equal to $\{1, 2, 3\}$.

(B) $(1, 1, 1, 1) \in \{0, 1\}^4$ is a binary form of $8 + 4 + 2 + 1 = 15$ and is the largest number with a binary form using 4 bits.

(C) $(1, 2, 3, 3) = (1, 3, 2, 3)$,

(D) If $u = (0, 1, 2, \ldots, 25)$ then $u_i = i - 1$ for $i \in \{0, 1, \ldots, 25\}$.

(A) (B) (C) (D)

§2 Prefix-free Binary Codes and Kraft's Inequality

Question. What properties should a binary code have so that we can decode it easily? How do these restrict its codewords?

Recall that if $\mathcal{A}$ is a set then $\mathcal{A}^\ell$ is the set of all ℓ-tuples of elements of $\mathcal{A}$. For example $\mathbb{R}^3 = \{(x, y, z) : x, y, z \in \mathbb{R}\}$ is 3-dimensional space.

In this course we number positions in tuples from 1 as usual, so z is the 3rd coordinate in (x, y, z).

Quiz: One of these statements is false. Which one?

(A) $\{1, 2, 3, 3\}$ is a set of size 3 and it's equal to $\{1, 2, 3\}$.

(B) $(1, 1, 1, 1) \in \{0, 1\}^4$ is a binary form of $8 + 4 + 2 + 1 = 15$ and is the largest number with a binary form using 4 bits.

(C) $(1, 2, 3, 3) = (1, 3, 2, 3)$,

(D) If $u = (0, 1, 2, \ldots, 25)$ then $u_i = i - 1$ for $i \in \{0, 1, \ldots, 25\}$.

(A) (B) (C) (D)

Binary Words and Codes

Definition 2.1

Let $\mathcal{A}$ be a set. A *word of length ℓ from $\mathcal{A}$* is an element of $\mathcal{A}^\ell$. We write $\mathcal{A}^\star$ for the set of all words from $\mathcal{A}$. We write $\ell(u)$ for the length of the word u. We write $\varnothing$ for the empty word.

Definition 2.2

A *binary word* is a word from $\{0, 1\}$. A *binary code* is a non-empty finite set of binary words. The words in a code are called *codewords*.

We usually write binary words omitting some of the tuple notation. For example $000, 001, 010, 011$, 100, $101, 110, 111$ all have length 3 and form a binary code of size 8.

Exercise 2.3

How many binary words are there of length n?

Binary Words and Codes

Definition 2.1

Let $\mathcal{A}$ be a set. A *word of length ℓ from $\mathcal{A}$* is an element of $\mathcal{A}^\ell$. We write $\mathcal{A}^\star$ for the set of all words from $\mathcal{A}$. We write $\ell(u)$ for the length of the word u. We write $\varnothing$ for the empty word.

Definition 2.2

A *binary word* is a word from $\{0, 1\}$. A *binary code* is a non-empty finite set of binary words. The words in a code are called *codewords*.

We usually write binary words omitting some of the tuple notation. For example $000, 001, 010, 011$, 100, $101, 110, 111$ all have length 3 and form a binary code of size 8.

Exercise 2.3

How many binary words are there of length n?

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

Binary Words and Codes

Definition 2.1

Let $\mathcal{A}$ be a set. A *word of length ℓ from $\mathcal{A}$* is an element of $\mathcal{A}^\ell$. We write $\mathcal{A}^\star$ for the set of all words from $\mathcal{A}$. We write $\ell(u)$ for the length of the word u. We write $\varnothing$ for the empty word.

Definition 2.2

A *binary word* is a word from $\{0,1\}$. A *binary code* is a non-empty finite set of binary words. The words in a code are called *codewords*.

We usually write binary words omitting some of the tuple notation. For example $000, 001, 010, 011$, 100, $101, 110, 111$ all have length 3 and form a binary code of size 8.

Exercise 2.3

How many binary words are there of length n?

(A) n (B) 2^n (C) $n!$ (D) 2^{2^n}

Binary Words and Codes

Definition 2.1

Let $\mathcal{A}$ be a set. A *word of length ℓ from $\mathcal{A}$* is an element of $\mathcal{A}^\ell$. We write $\mathcal{A}^\star$ for the set of all words from $\mathcal{A}$. We write $\ell(u)$ for the length of the word u. We write $\varnothing$ for the empty word.

Definition 2.2

A *binary word* is a word from $\{0, 1\}$. A *binary code* is a non-empty finite set of binary words. The words in a code are called *codewords*.

We usually write binary words omitting some of the tuple notation. For example $000, 001, 010, 011$, 100, $101, 110, 111$ all have length 3 and form a binary code of size 8.

Exercise 2.3

How many binary words are there of length n?

(A) n (B) 2^n (C) $n!$ (D) 2^{2^n}

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0$

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 010$

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 010110$

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0101100$

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0101100111$ unambiguously

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0101100111$ unambiguously

Definition 2.4

Let u and w be binary words, of lengths ℓ and m, respectively. We say that u is a *prefix* of w is $\ell \leq m$ and $w = u_1 \dots u_\ell w_{\ell+1} \dots w_m$. A binary code C is *prefix-free* if no codeword in C is a prefix of another codeword in C.

True or false: a concatenation of codewords from a binary code C can always be decoded by this left-to-right strategy

- if C is prefix-free.

(A) False (B) True

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0101100111$ unambiguously

Definition 2.4

Let u and w be binary words, of lengths ℓ and m, respectively. We say that u is a *prefix* of w is $\ell \leq m$ and $w = u_1 \dots u_\ell w_{\ell+1} \dots w_m$. A binary code C is *prefix-free* if no codeword in C is a prefix of another codeword in C.

True or false: a concatenation of codewords from a binary code C can always be decoded by this left-to-right strategy

- if C is prefix-free.

  (A) False (B) True

- only if C is prefix-free.

  (A) False (B) True

Prefix-free Codes

In Exercise 1.7 we saw the binary code $C = \{0, 10, 110, 111\}$. When codewords from C are concatenated, one can read the concatenated word from left to right and decode it by splitting it as soon as a codeword is seen. For example

- $0101100111 = 0101100111$ unambiguously

Definition 2.4

Let u and w be binary words, of lengths ℓ and m, respectively. We say that u is a *prefix* of w is $\ell \leq m$ and $w = u_1 \dots u_\ell w_{\ell+1} \dots w_m$. A binary code C is *prefix-free* if no codeword in C is a prefix of another codeword in C.

True or false: a concatenation of codewords from a binary code C can always be decoded by this left-to-right strategy

- if C is prefix-free.

  (A) False (B) True

- only if C is prefix-free.

  (A) False (B) True

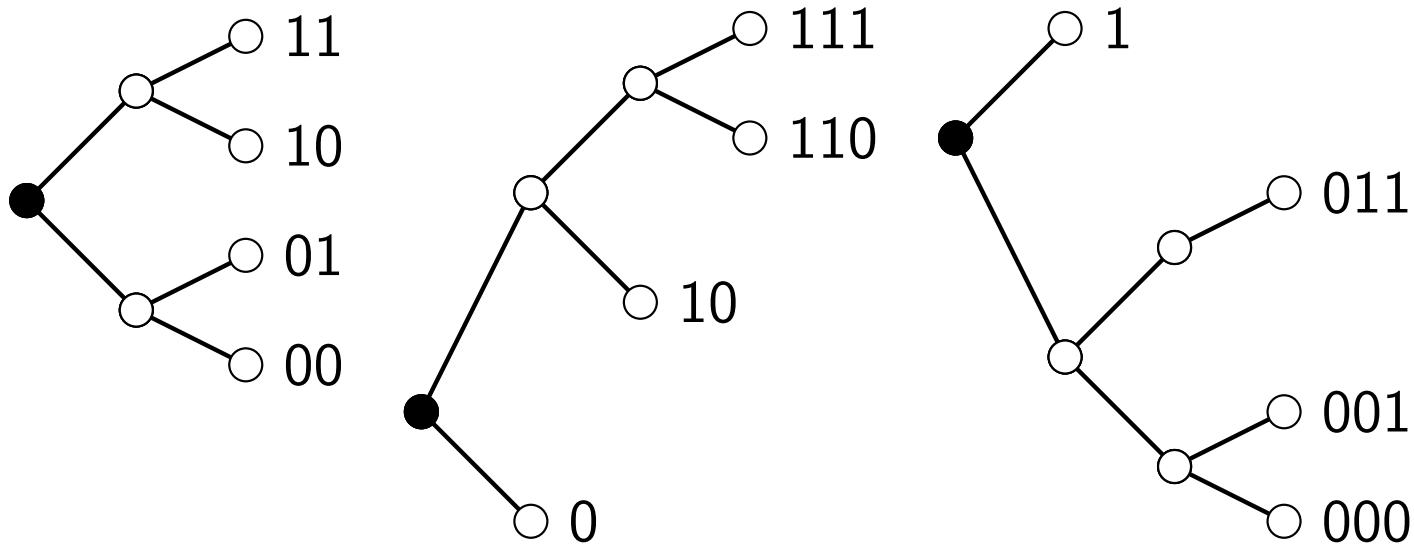
Uniquely Decipherable Codes

A code where there is a unique way to decode every concatenation of codewords is said to be *uniquely decipherable*. (The decoding might be more complicated than just reading left-to-right.) Please do Exercise 2.5 in your own time.

- The optional 'extras' for this part (to come in printed notes) have more on uniquely decipherable codes.
- We shall concentrate on prefix-free codes, which are by far the most important in practice.

Prefix-free Codes and Binary Trees

It is useful to represent prefix-free binary codes by oriented rooted binary trees. We read codewords left to right, so we grow the tree the same way, stepping up for 1 and down for 0.

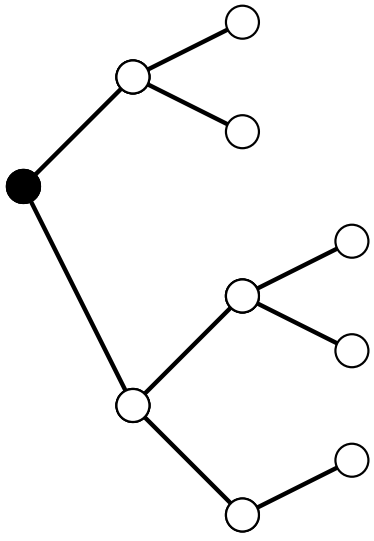


Exercise 2.6

(a) How could the third code $\{000, 001, 011, 1\}$ be made more efficient, while keeping it prefix-free? How can this improvement be seen from the tree?

Exercise 2.6

(b) What is the code corresponding to the tree below?
Remember: up for 1, down for 0.



(A) $\{11, 10, 011, 010, 00\}$

(B) $\{11, 10, 011, 010, 000\}$

(C) $\{11, 10, 011, 010, 001\}$

(D) something else

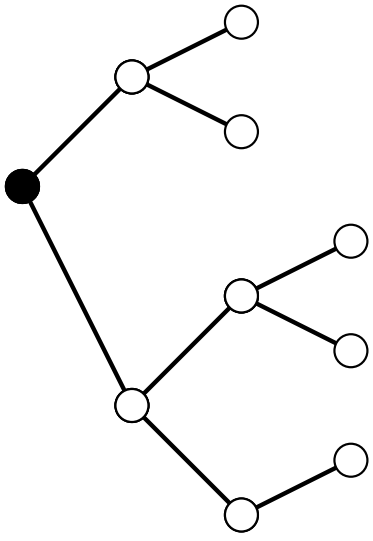
(A) (B) (C) (D)

Exercise 2.7

Draw the tree corresponding to the prefix-free code in Exercise 1.4 with codewords $\{1, 0000, 00010, 00011, \ldots, 01111\}$. What is the corresponding questioning strategy for the guessing game?

Exercise 2.6

(b) What is the code corresponding to the tree below?
Remember: up for 1, down for 0.



(A) $\{11, 10, 011, 010, 00\}$

(B) $\{11, 10, 011, 010, 000\}$

(C) $\{11, 10, 011, 010, 001\}$

(D) something else

(A) (B) (C) (D)

Exercise 2.7

Draw the tree corresponding to the prefix-free code in Exercise 1.4 with codewords $\{1, 0000, 00010, 00011, \ldots, 01111\}$. What is the corresponding questioning strategy for the guessing game?

Hint: you really only need one question!

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes

(ii) (A) No (B) Yes

(iii) (A) No (B) Yes

(iv) (A) No (B) Yes

(v) (A) No (B) Yes

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i) (A) No (B) Yes
(ii) (A) No (B) Yes
(iii) (A) No (B) Yes
(iv) (A) No (B) Yes
(v) (A) No (B) Yes

Quiz: One of these statements is false: which one?

(A) 'P if Q' means $P \Leftarrow Q$;
(B) 'P only if Q' means $P \Rightarrow Q$;
(C) 'P if and only if Q' means $P \iff Q$
(D) 'P if Q' means $P \Rightarrow Q$;

(A) (B) (C) (D)

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i)	(A) No	(B) Yes	$7/8$
(ii)	(A) No	(B) Yes	1
(iii)	(A) No	(B) Yes	$9/8$
(iv)	(A) No	(B) Yes	$7/8$
(v)	(A) No	(B) Yes	1

Quiz: One of these statements is false: which one?

(A) 'P if Q' means $P \Leftarrow Q$;

(B) 'P only if Q' means $P \Rightarrow Q$;

(C) 'P if and only if Q' means $P \iff Q$

(D) 'P if Q' means $P \Rightarrow Q$;

(A) (B) (C) (D)

Kraft's Inequality

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i)	(A) No	(B) Yes	$7/8$
(ii)	(A) No	(B) Yes	1
(iii)	(A) No	(B) Yes	$9/8$
(iv)	(A) No	(B) Yes	$7/8$
(v)	(A) No	(B) Yes	1

Proposition 2.9 (Kraft's Inequality)

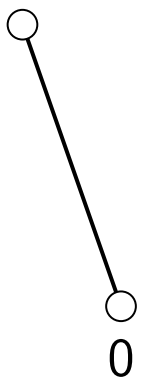
Let $\ell_1, \ell_2, \ldots, \ell_s \in \mathbb{N}$. There is a prefix-free binary code whose codewords have lengths $\ell_1, \ell_2, \ldots, \ell_s$ if and only if

$$2^{-\ell_1} + 2^{-\ell_2} + \cdots + 2^{-\ell_s} \leq 1.$$

Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.

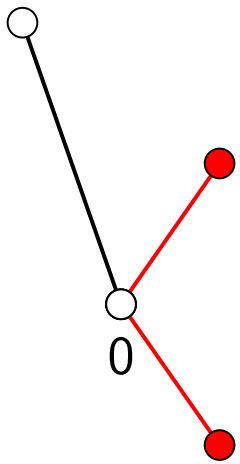
(1) Choose 0:



Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.

(1) Choose 0: now $00, 01$ forbidden

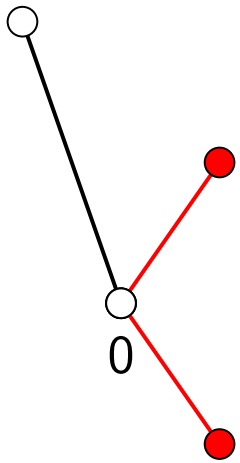


Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.

(1) Choose 0: now $00, 01$ forbidden

(2) Nothing to choose:

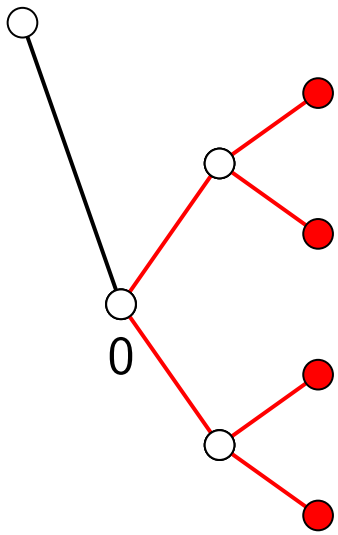


Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.

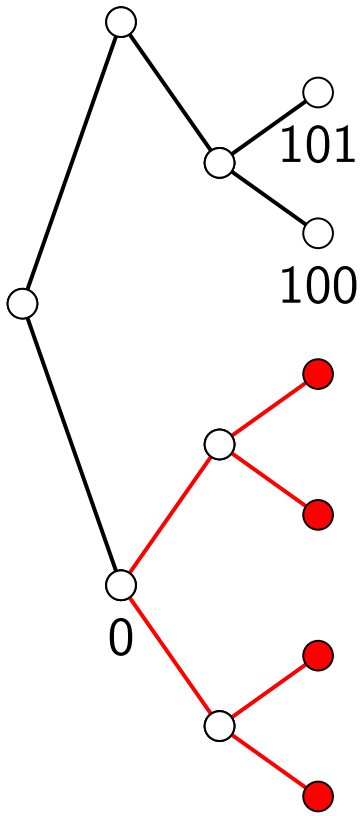
(1) Choose 0: now $00, 01$ forbidden

(2) Nothing to choose: now $000, 001, 010, 011$ forbidden



Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.



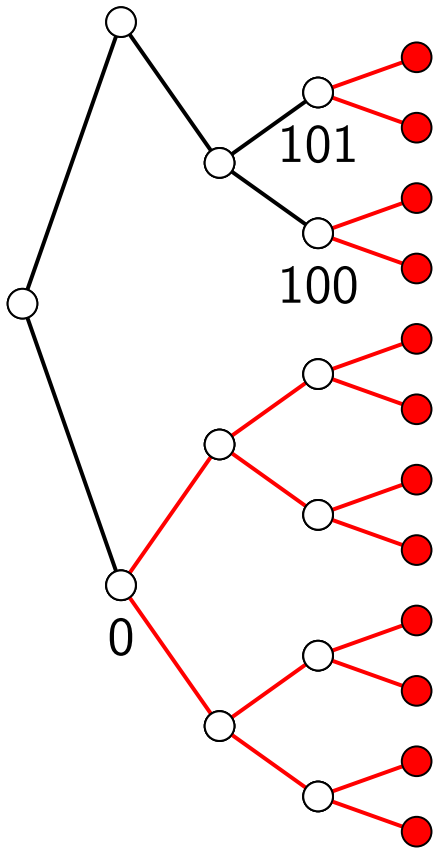
(1) Choose 0: now $00, 01$ forbidden

(2) Nothing to choose: now $000, 001, 010, 011$ forbidden

(3) Choose 100, 101:

Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.



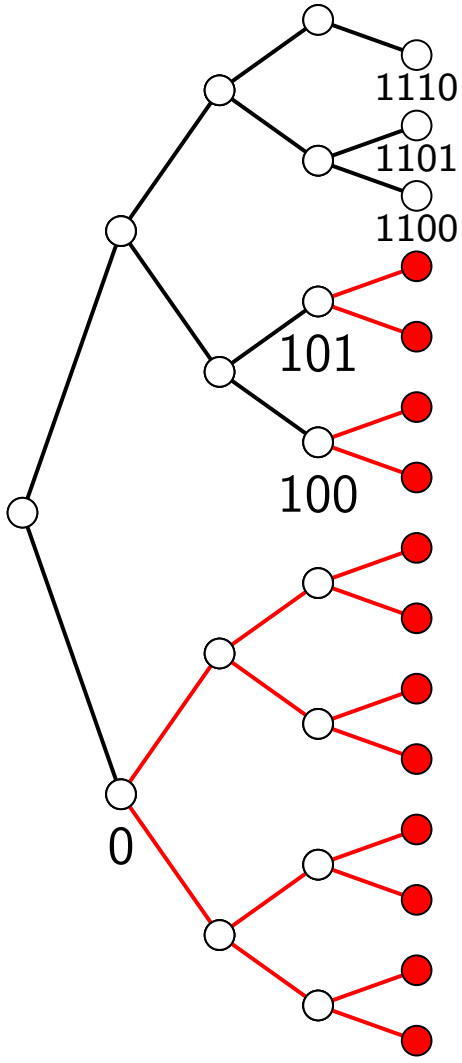
(1) Choose 0: now $00, 01$ forbidden

(2) Nothing to choose: now $000, 001, 010, 011$ forbidden

(3) Choose 100, 101: now $0000, 0001,$ $\ldots$, 0110, 0111, 1000, 1001, 1010, 1011 forbidden

Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.



(1) Choose 0: now $00, 01$ forbidden

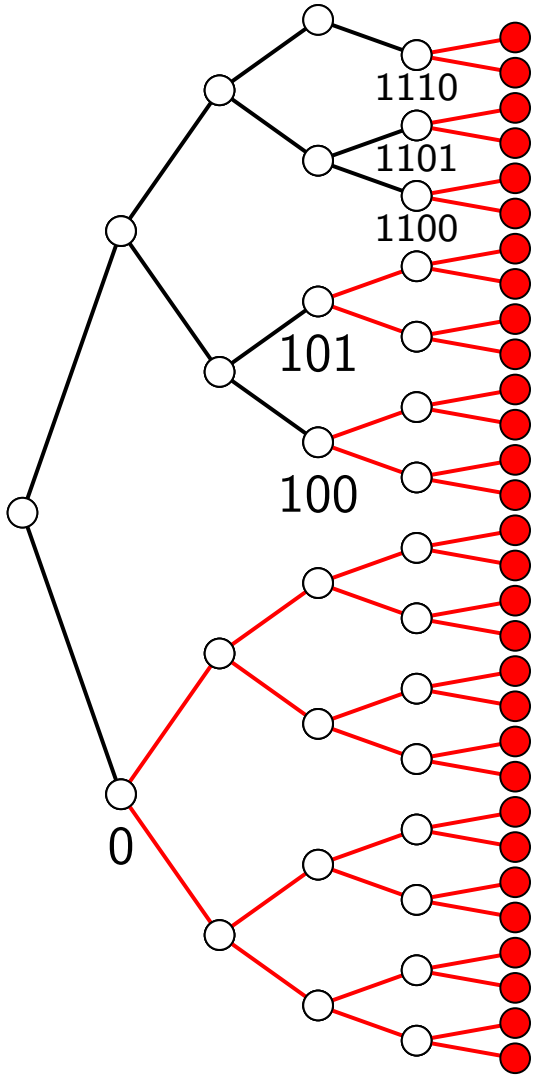
(2) Nothing to choose: now $000, 001, 010, 011$ forbidden

(3) Choose 100, 101: now $0000, 0001,$ $\ldots,$ 0110, 0111, 1000, 1001, 1010, 1011 forbidden

(4) Choose 1100, 1101, 1110:

Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.



(1) Choose 0: now $00, 01$ forbidden

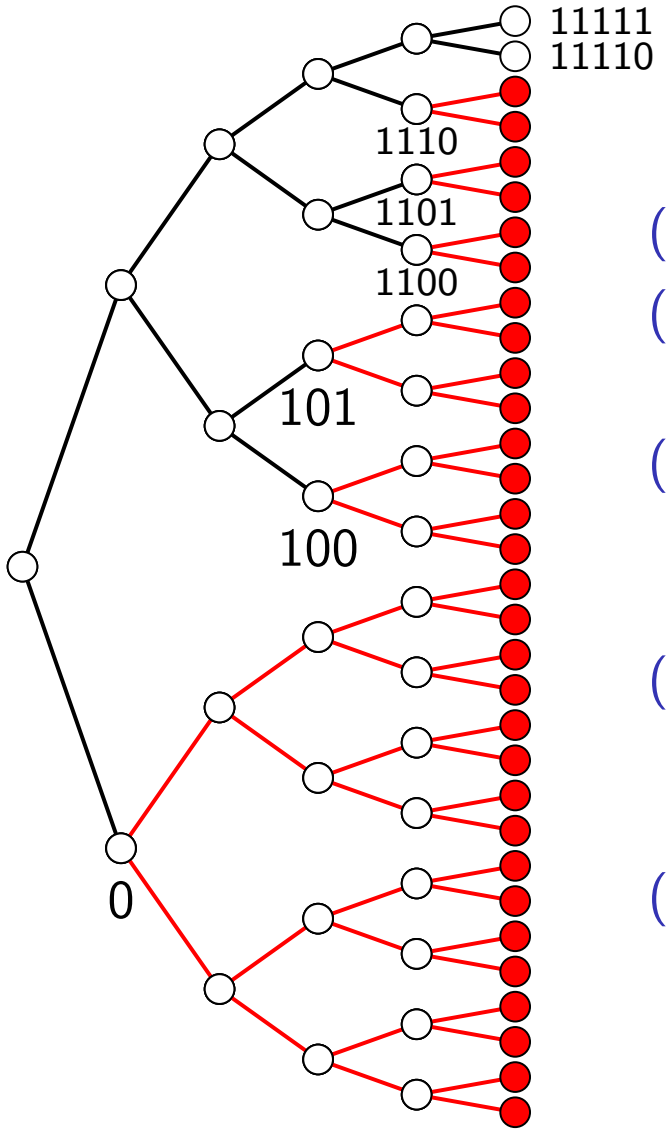
(2) Nothing to choose: now $000, 001, 010, 011$ forbidden

(3) Choose 100, 101: now $0000, 0001,$ $\ldots,$ 0110, 0111, 1000, 1001, 1010, 1011 forbidden

(4) Choose 1100, 1101, 1110: now all but two words of length 5 are forbidden prefixes

Example 2.10: Prefix-free lengths $1, 3, 3, 4, 4, 4, 5, 5$

We construct 'greedily', choosing lowest (smallest) codeword each time. Red vertices show the forbidden prefixes of each length.



(1) Choose 0: now $00, 01$ forbidden

(2) Nothing to choose: now $000, 001, 010, 011$ forbidden

(3) Choose 100, 101: now $0000, 0001,$ $\ldots,$ 0110, 0111, 1000, 1001, 1010, 1011 forbidden

(4) Choose 1100, 1101, 1110: now all but two words of length 5 are forbidden prefixes

(5) Choose 11110, 11111

Kraft's Inequality (Reminder of Exercise and Statement)

Exercise 2.8

For each sequence decide if it is the lengths of the codewords in a prefix-free binary code: (i) $1, 3, 3, 3$; (ii) $1, 3, 3, 2$; (iii) $1, 2, 2, 3$; (iv) $3, 2, 2, 2$; (v) $1, 3, 3, 4, 4, 4, 5, 5$.

(i)	(A) No	(B) Yes	$7/8$
(ii)	(A) No	(B) Yes	1
(iii)	(A) No	(B) Yes	$9/8$
(iv)	(A) No	(B) Yes	$7/8$
(v)	(A) No	(B) Yes	1

Proposition 2.9 (Kraft's Inequality)

Let $\ell_1, \ell_2, \ldots, \ell_s \in \mathbb{N}$. There is a prefix-free binary code whose codewords have lengths $\ell_1, \ell_2, \ldots, \ell_s$ if and only if

$$2^{-\ell_1} + 2^{-\ell_2} + \cdots + 2^{-\ell_s} \leq 1.$$

§3 Sources and Entropy

Question. What is an efficient way to code messages when some are much more frequent than others?

To answer this question we must set up some definitions.

Definition 3.1

An *alphabet* is a finite non-empty set of *symbols*. A *source* is a random process producing a sequence $U_1, U_2, \ldots$ of symbols from an alphabet. A source is *memoryless* if the U_i are independent and identically distributed.

Memoryless Property

Example 3.2

(1) A coin that lands heads with probability p, independently of previous flips, is a memoryless source producing symbols from the alphabet $\{\mathrm{H}, \mathrm{T}\}$. We have $\mathbb{P}[U_t = \mathrm{H}] = p$ for all t.

(2) A binary source emits $0, 0, 0, \ldots$ or $1, 1, 1, \ldots$ with equal probability $\frac{1}{2}$. Like the previous example with $p = \frac{1}{2}$, we have $\mathbb{P}[U_t = 0] = \mathbb{P}[U_t = 1] = \frac{1}{2}$ for all t. But since U_1 and U_2 are not independent, the source is not memoryless.

(3) A source produces random meaningful English messages in lower case with all punctuation except spaces deleted. The alphabet is the Roman alphabet together with space. After receiving '`the source is not memoryles`' you can easily guess the next character. Therefore ...

Source Coding for a Memoryless Source

Example 3.3

A memoryless source produces symbols from the alphabet $\{1, 2, 3, 4\}$ such that $\mathbb{P}[U_t = i] = p_i$ for each $i \in \{1, 2, 3, 4\}$, where $(p_1, p_2, p_3, p_4) = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$. We encode symbols using the prefix-free binary code $\{0, 10, 110, 111\}$ by

$$1 \mapsto 0,\ 2 \mapsto 10,\ 3 \mapsto 110,\ 4 \mapsto 111$$

The expected length of a codeword is then $\frac{1}{2}1 + \frac{1}{4}2 + \frac{1}{8}3 + \frac{1}{8}3 = \frac{7}{4}$.

Feedback on Sheet 2

(2) You draw two cards from a deck of 52 cards. As usual it has 4 suits each of 13 cards. The first card is not replaced before drawing the second. What is the probability of drawing two cards of the same suit?

Feedback on Sheet 2

(2) You draw two cards from a deck of 52 cards. As usual it has 4 suits each of 13 cards. The first card is not replaced before drawing the second. What is the probability of drawing two cards of the same suit?

After drawing the first card, there are 12 remaining cards of its suit and 51 cards in the deck. The probability is therefore $\frac{12}{51} = \frac{4}{17}$.

Common error: A very common error was to write down $\frac{13}{52} \times \frac{12}{51}$ without much explanation. This is the probability that both cards are spades. But the question asks for the probability that the second card has the same suit as the first: it could be any suit.

Feedback on Sheet 2

(2) You draw two cards from a deck of 52 cards. As usual it has 4 suits each of 13 cards. The first card is not replaced before drawing the second. What is the probability of drawing two cards of the same suit?

After drawing the first card, there are 12 remaining cards of its suit and 51 cards in the deck. The probability is therefore $\frac{12}{51} = \frac{4}{17}$.

> **Common error:** A very common error was to write down $\frac{13}{52} \times \frac{12}{51}$ without much explanation. This is the probability that both cards are spades. But the question asks for the probability that the second card has the same suit as the first: it could be any suit.

Question 4 is like Exercise 3.4(iii). See feedback please on Question 5(e).

You Can Discover Entropy!

Exercise 3.4

For each of the following alphabets and probability measures find a binary encoder using a prefix-free code. Try to minimize the expected length of the codeword. [*Hint:* Kraft's Inequality tells you what lengths are possible; (iii) is related to Exercise 2.8(v).]

(i) $\{1,2,3,4\}$: $(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$;

(ii) $\{1,2,3\}$: $(\frac{1}{2},\frac{1}{4},\frac{1}{4})$;

(iii) $\{1,2,3,4,5,6,7,8\}$: $(\frac{1}{2},\frac{1}{2^3},\frac{1}{2^3},\frac{1}{2^4},\frac{1}{2^4},\frac{1}{2^4},\frac{1}{2^5},\frac{1}{2^5})$;

(iv) $\{1,2,3,4,5\}$: $(\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5})$.

Exercise 3.5

With the setup of Exercise 3.4, suppose that the alphabet is $\{1,\ldots,s\}$ and that $p_i = 1/2^{c_i}$ for each i. Show that there is a prefix-free binary code with codewords $u(1),\ldots,u(s)$ such that $u(i)$ has length c_i for each i. What is the expected codeword length? Write this expectation just using $p_1,\ldots,p_s$.

You Can Discover Entropy!

Exercise 3.4

For each of the following alphabets and probability measures find a binary encoder using a prefix-free code. Try to minimize the expected length of the codeword. [*Hint:* Kraft's Inequality tells you what lengths are possible; (iii) is related to Exercise 2.8(v).]

(i) $\{1,2,3,4\}$: $(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$; $\{00,01,10,11\}$, $\bar{\ell}=2$

(ii) $\{1,2,3\}$: $(\frac{1}{2},\frac{1}{4},\frac{1}{4})$; $\{0,10,11\}$, $\bar{\ell}=\frac{3}{2}$

(iii) $\{1,2,3,4,5,6,7,8\}$: $(\frac{1}{2},\frac{1}{2^3},\frac{1}{2^3},\frac{1}{2^4},\frac{1}{2^4},\frac{1}{2^4},\frac{1}{2^5},\frac{1}{2^5})$; Ex 2.10, $\bar{\ell}=\frac{37}{16}$

(iv) $\{1,2,3,4,5\}$: $(\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5},\frac{1}{5})$.

Exercise 3.5

With the setup of Exercise 3.4, suppose that the alphabet is $\{1,\ldots,s\}$ and that $p_i = 1/2^{c_i}$ for each i. Show that there is a prefix-free binary code with codewords $u(1),\ldots,u(s)$ such that $u(i)$ has length c_i for each i. What is the expected codeword length? Write this expectation just using $p_1,\ldots,p_s$.

Entropy

Definition 3.6 (Entropy of probability measure)

Let p_x for $x \in \Omega$ be a probability measure on a set Ω. The *entropy* of p is

$$H(p) = -\sum_{\omega \in \Omega} p_\omega \log_2 p_\omega.$$

To deal with the case when $p_x = 0$, we use the convention that $0 \log_2 0 = 0$. This is consistent with the graph of $-x \log_2 x$. In Exercise 3.5 you might have discovered the equivalent form $H(p) = \sum_{x \in \Omega} p_\omega \log_2 \frac{1}{p_\omega}$.

Exercise 3.7

(a) Suppose that $p_i = \frac{1}{s}$ for $i \in \{1, \ldots, s\}$. What is $H(p)$?

(b) Show that in each case in Example 3.4, the expected length of the code is at least $H(p)$, and that equality holds for (i), (ii) and (iii).

Shannon Codes

By Exercise 3.5, when all the probabilities in a probability measure p are powers of 2 there is a prefix-free binary code with expected length $h(p)$. In general, we cannot do quite so well, but using almost the same idea, we can still get a good code.

Recall that if $x \in \mathbb{R}$ then $\lceil x \rceil$ is the least natural number n such that $x \leq n$. The function $x \mapsto \lceil x \rceil$ is called the *ceiling function*. For example

$$\lceil 3\tfrac{1}{4} \rceil = \lceil \pi \rceil = \lceil 4 \rceil = 4.$$

Proposition 3.8 (Shannon Code)

Let p be a probability measure on $\{1, \ldots, s\}$.

(i) *There is a prefix-free binary code with codewords $u(1), \ldots, u(s)$ such that $u(i)$ has length $\lceil \log_2 \frac{1}{p_i} \rceil$.*

(ii) *When $u(i)$ is used to encode i for each $i \in \{1, \ldots, s\}$, the expected codeword length is less than $1 + H(p)$.*

Gibbs' Inequality

Motivated by Exercise 3.7, we now show that the expected length of a prefix-free binary code for a probability measure p is always at least $H(p)$. Thus Shannon Codes have expected length within 1 of the best possible. We need the following fundamental inequality.

Lemma 3.9 (Gibbs' Inequality)

Let p and q be probability measures on the set $\{1, \ldots, s\}$. Then

$$-\sum_{i=1}^{s} p_i \log_2 p_i \leq -\sum_{i=1}^{s} p_i \log_2 q_i$$

where the right-hand side is interpreted as $+\infty$ if $q_i = 0$ for some $p_i \neq 0$.

Gibbs' Inequality

Motivated by Exercise 3.7, we now show that the expected length of a prefix-free binary code for a probability measure p is always at least $H(p)$. Thus Shannon Codes have expected length within 1 of the best possible. We need the following fundamental inequality.

Lemma 3.9 (Gibbs' Inequality)

Let p and q be probability measures on the set $\{1, \ldots, s\}$. Then

$$-\sum_{i=1}^{s} p_i \log_2 p_i \leq -\sum_{i=1}^{s} p_i \log_2 q_i$$

where the right-hand side is interpreted as $+\infty$ if $q_i = 0$ for some $p_i \neq 0$.

Corollary 3.10

Suppose that a prefix-free binary code with codewords of lengths $\ell_1, \ldots, \ell_s$ is used to encode symbols from $\{1, \ldots, s\}$. If symbol i has probability p_i then the expected codeword length $\bar{\ell}$ is at least $H(p)$.

Summary

Proposition 3.8 (Shannon Code)

Let p be a probability measure on $\{1, \ldots, s\}$.

(i) *There is a prefix-free binary code with codewords $u(1), \ldots, u(s)$ such that $u(i)$ has length $\lceil \log_2 \frac{1}{p_i} \rceil$.*

(ii) *When $u(i)$ is used to encode i for each $i \in \{1, \ldots, s\}$, the expected codeword length is less than $1 + H(p)$.*

Corollary 3.10

Suppose that a prefix-free binary code with codewords of lengths $\ell_1, \ldots, \ell_s$ is used to encode symbols from $\{1, \ldots, s\}$. If symbol i has probability p_i then the expected codeword length $\bar{\ell}$ is at least $H(p)$.

Summary. If symbols come from an alphabet with probability measure p then the expected length of a prefix-free binary code is at least $H(p)$. A Shannon code has expected length less than $H(p) + 1$.

§4 Entropy and the Noiseless Coding Theorem

Definition 4.1 (Entropy of random variable)

Let X be a random variable taking values in a set $\mathcal{X}$. The *entropy* of X is

$$H(X) = -\sum_{x \in \mathcal{X}} \mathbb{P}[X = x] \log_2 \mathbb{P}[X = x].$$

Equivalently, $H(X) = H(p)$ where the probability measure p on $\mathcal{X}$ is defined by $p_\omega = \mathbb{P}[X = X]$ and $H(p)$ is as defined in Definition 3.6.

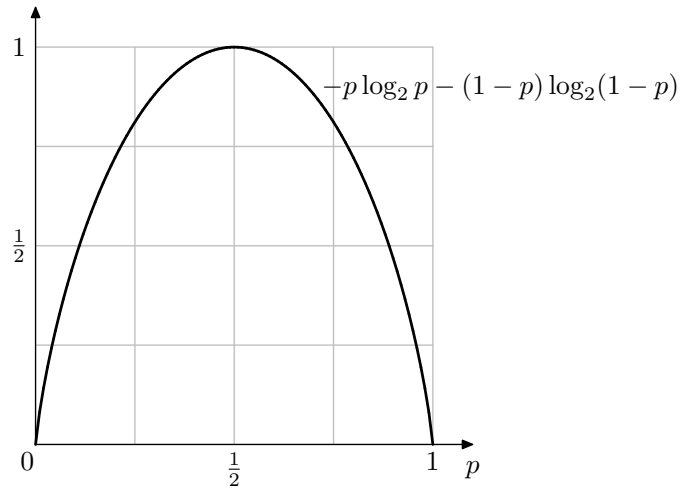
The intuitive idea of entropy is that it is the amount of information, measured in bits, that we learn by observing a random variable.

Entropy Examples

Example 4.2

(1) Let X and Y be independent tosses of a fair coin. Then $H(X) = H(Y) = 1$ and $H\big((X, Y)\big) = 2$.

(2) Let U be a toss of a coin biased to land heads with probability p. Then $H(U) = -p \log p - (1-p)\log(1-p)$ as shown in the graph below.



Prove that, as suggested by the graph, the entropy is 0 when $p = 0$ and when $p = 1$ and is maximized at 1 when $p = \frac{1}{2}$. Is it intuitive that the graph is symmetric about $\frac{1}{2}$?

Joint Entropy

We define the *joint* entropy of random variables X and Y taking values in sets $\mathcal{X}$ and $\mathcal{Y}$ by

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \mathbb{P}[X = x, Y = y] \log_2 \mathbb{P}[X = x, Y = y].$$

Equivalently, $H(X, Y)$ is the entropy of the random variable (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$.

Exercise 4.3

Let X and Y be two independent flips of a coin biased to land heads with probability p. What is the joint distribution of X and Y? Express $H(X)$, $H(Y)$, $H(X, Y)$ and $H(X, X)$ in terms of $h = -p \log_2 p - (1 - p) \log_2 (1 - p)$.

Joint Entropy

We define the *joint* entropy of random variables X and Y taking values in sets $\mathcal{X}$ and $\mathcal{Y}$ by

$$H(X,Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \mathbb{P}[X=x, Y=y]\log_2 \mathbb{P}[X=x, Y=y].$$

Equivalently, $H(X,Y)$ is the entropy of the random variable (X,Y) taking values in $\mathcal{X}\times\mathcal{Y}$.

Exercise 4.3

Let X and Y be two independent flips of a coin biased to land heads with probability p. What is the joint distribution of X and Y? Express $H(X)$, $H(Y)$, $H(X,Y)$ and $H(X,X)$ in terms of $h = -p\log_2 p - (1-p)\log_2(1-p)$.

Lemma 4.4

If X and Y are independent random variables then

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

Feedback on Sheet 2, Correction to Sheet 3

- In Question 2(iii), probability measures on the alphabet $\{1, \ldots, s\}$ were specified. Thus in (i), $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$ has $s = 4$. In (iii) the probability measure is $(\frac{1}{2^m}, \ldots, \frac{1}{2^m})$. Since the sum of probabilities is 1, the size of the alphabet is 2^m. The Shannon code is all binary words of length m.
- Question 4(b) and (c): will do today.
- Style: please use words and/or implication signs $\Longrightarrow$, $\Longleftrightarrow$ to show the logic of your argument. See feedback on Question 4 for an example where this is helpful.
- Please use technical terms accurately. They have precise mathematical definitions that help us to communicate clearly. For instance, a *codeword* is an element of a *code*. Do not write *codes* if you mean *codewords*.
- Correction to Question 3(a) on Sheet 3: the expected length is

$$\tfrac{1}{5}\big(\ell\big(u(\mathtt{a})\big) + \ell\big(u(\mathtt{b})\big) + \ell\big(u(\mathtt{c})\big) + \ell\big(u(\mathtt{d})\big) + \ell\big(u(\mathtt{e})\big)\big)$$

not $\tfrac{1}{5}\big(\ell\big(u(\mathtt{a})\big) + \ell\big(u(\mathtt{a})\big) + \ell\big(u(\mathtt{b})\big) + \ell\big(u(\mathtt{c})\big) + \ell\big(u(\mathtt{d})\big)\big)$.

Example 4.5

A memoryless source produces symbols from the alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ so that $\mathbb{P}[U_t = \mathtt{a}] = \frac{1}{2}$, $\mathbb{P}[U_t = \mathtt{b}] = \frac{2}{5}$, $\mathbb{P}[U_t = \mathtt{c}] = \frac{1}{10}$ for all times t. We have

$$H(U_1) = \tfrac{1}{2}\log_2 2 + \tfrac{2}{5}\log_2 \tfrac{5}{2} + \tfrac{1}{10}\log_2 10 = \tfrac{1}{5} + \tfrac{1}{2}\log_2 5 \approx 1.361.$$

The Shannon code for the probability distribution $(\frac{1}{2}, \frac{2}{5}, \frac{1}{10})$ has codewords of lengths $\lceil \log_2 2 \rceil = 1$, $\lceil \log_2 \frac{3}{2} \rceil = 2$, $\lceil \log_2 10 \rceil = 4$. With one choice of codewords,

$$\mathtt{a} \mapsto 0, \mathtt{b} \mapsto 10, \mathtt{c} \mapsto 1111.$$

The expected length is $\frac{1}{2}1 + \frac{2}{5}2 + \frac{1}{10}4 = \frac{17}{10}$. As expected by Proposition 3.8 (upper bound) and Corollary 3.10 (lower bound),

$$H(U_1) \leq \tfrac{17}{10} < H(U_1) + 1.$$

Since $1.7 - H(U_1) \approx 0.339$, for every symbol encoded, the Shannon code is worse by 0.339 bits compared to the entropy bound.

Example 4.5 continued

Suppose we now encode pairs of symbols. By a similar argument to Exercise 4.3, the probability distribution is:

	a	b	c
a	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{20}$
b	$\frac{1}{5}$	$\frac{4}{25}$	$\frac{1}{25}$
c	$\frac{1}{20}$	$\frac{1}{25}$	$\frac{1}{100}$

The Shannon code has codewords of lengths $\lceil \log_2 4 \rceil$, $\lceil \log_2 5 \rceil$, $\lceil \log_2 20 \rceil$, $\lceil \log_2 5 \rceil$, $\lceil \log_2 \frac{25}{4} \rceil$, $\lceil \log_2 25 \rceil$ $\lceil \log_2 20 \rceil$, $\lceil \log_2 25 \rceil$, $\lceil \log_2 100 \rceil$, namely $2, 3, 5, 3, 3, 5, 5, 5, 7$. Expected length:

$$\tfrac{1}{4}2 + \tfrac{1}{5}3 + \tfrac{1}{20}5 + \tfrac{1}{5}3 + \tfrac{4}{25}3 + \tfrac{1}{25}5 + \tfrac{1}{20}5 + \tfrac{1}{25}5 + \tfrac{1}{100}7 = \tfrac{315}{100}.$$

By Lemma 4.4, the entropy of a pair of symbols is $2H(U_1) \approx 2.722$. Since $3.15 - 2H(U_1) \approx 0.428$, $0.428/2 = 0.214$ bits are wasted per symbol. This improves on 0.339 earlier encoding symbols one at a time.

Example 4.5 concluded

The table below shows what happens when we encode symbols r at a time: $\bar{\ell}^{(r)}$ is the expected length of the Shannon code and the final column shows

$$\epsilon^{(r)} = \frac{\bar{\ell}^{(r)}}{r} - H(U_1),$$

the number of wasted bits per symbol.

r	$\bar{\ell}^{(r)}$	$rH(U_1)$	$\bar{\ell}^{(r)} - rH(U_1)$	$\epsilon^{(r)}$
1	1.700	1.361	0.339	0.339
2	3.150	2.722	0.428	0.214
3	4.475	4.083	0.392	0.131
4	5.800	5.444	0.356	0.089
5	7.156	6.805	0.351	0.070
6	8.528	8.1658	0.362	0.060
7	9.900	9.5267	0.373	0.053
8	11.268	10.889	0.380	0.048
9	12.634	12.249	0.386	0.043
10	14.000	13.610	0.390	0.039

Preliminaries for Shannon's Noiseless Coding Theorem

Given a source $U_1, U_2, \ldots$ producing symbols from an alphabet $\mathcal{A}$, a binary code $C^{(r)}$ and an encoder $f^{(r)} : \mathcal{A}^r \to C^{(r)}$, let $\bar{f}^{(r)}$ be the expected length of a codeword encoding $(U_1, \ldots, U_r)$. In symbols

$$\bar{f}^{(r)} = \sum_{(u_1,\ldots,u_r)\in\mathcal{A}^r} \ell\big(f^{(r)}(u_1, \ldots, u_r)\big)\mathbb{P}\big[(U_1, \ldots, U_r) = (u_1, \ldots, u_r)\big].$$

When $r = 1$, we encode symbols one at a time and write f rather than $f^{(1)}$.

Preliminaries for Shannon's Noiseless Coding Theorem

Given a source $U_1, U_2, \ldots$ producing symbols from an alphabet $\mathcal{A}$, a binary code $C^{(r)}$ and an encoder $f^{(r)} : \mathcal{A}^r \to C^{(r)}$, let $\bar{f}^{(r)}$ be the expected length of a codeword encoding $(U_1, \ldots, U_r)$. In symbols

$$\bar{f}^{(r)} = \sum_{(u_1,\ldots,u_r)\in\mathcal{A}^r} \ell\big(f^{(r)}(u_1,\ldots,u_r)\big)\mathbb{P}\big[(U_1,\ldots,U_r) = (u_1,\ldots,u_r)\big].$$

When $r = 1$, we encode symbols one at a time and write f rather than $f^{(1)}$.

For instance in Example 4.5, we had $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and saw prefix-free encoders $f^{(r)}$ for the r-tuples of symbols from $\mathcal{A}$. When $r = 1$ we encoded symbols one at a time with $f(\mathtt{a}) = 0$, $f(\mathtt{b}) = 10$, $f(\mathtt{c}) = 1111$; the expected length was $\bar{f} = \bar{f}^{(1)} = \frac{1}{2}1 + \frac{2}{5}2 + \frac{1}{10}4 = 1.7$. The second column in the table shows the values of $\bar{f}^{(r)}$; for instance $\bar{f}^{(10)} = 14.000$.

Shannon's Noiseless Coding Theorem

Theorem 4.6 (Shannon's Noiseless Coding Theorem, Memoryless Case)

A memoryless source produces symbols $U_1, U_2, \ldots$ from an alphabet $\mathcal{A}$ such that each symbol has positive probability. Let $h = H(U_1)$.

(i) *There exists a prefix-free binary code C and an injective encoder $f : \mathcal{A} \to C$ such that $\bar{f} < h + 1$.*

(ii) *For any prefix-free injective encoder g, we have $\bar{g} \geq h$.*

Theorem 4.7 (Shannon's Noiseless Coding Theorem, Asymptotic Memoryless Case)

A memoryless source produces symbols $U_1, U_2, \ldots$ from an alphabet $\mathcal{A}$ such that each symbol has positive probability. Let $h = H(U_1)$.

(i) *For every $\epsilon > 0$ there exists $r \in \mathbb{N}$, a prefix-free binary code $C^{(r)}$ and an injective encoder $f^{(r)} : \mathcal{A}^r \to C^{(r)}$ such that*

$$\frac{\bar{f}^{(r)}}{r} < h + \epsilon.$$

(ii) *For any r and any prefix-free injective encoder $g^{(r)}$,*

$$\frac{\bar{g}^{(r)}}{r} \geq h.$$

§5 Huffman codes

Returning to encoding symbol by symbol, we end by defining Huffman codes. We prove in Corollary 5.9 that they have the shortest possible expected length of prefix-free codes. Huffman codes are widely used because they are efficient to construct: they are used in JPEG image compression, MP3 audio compression and ZIP file compression.

Huffman Sets

The following definition is non-standard, but very useful.

Definition 5.1

A *Huffman set* for a probability measure $(p_1, \ldots, p_s)$ is a set of pairs (i, u) where, in each pair, $i \in \{1, \ldots, s\}$ and u is a binary word. The *weight* of a Huffman set is the sum of the p_i for those i in a pair in the set.

The input and output to each step of the Huffman algorithm is a collection of Huffman sets. We shall see that (i, v) appears in a Huffman set if and only if the codeword $v(i)$ encoding i **ends** with v. We say that v is a *suffix* of $v(i)$.

For example, $\{(1, \varnothing)\}$ and $\{(2, 001), (3, 101)\}$ are Huffman sets having weights p_1 and $p_2 + p_3$. From the second Huffman set we know that the codeword $v(3)$ encoding 3 has 101 as a suffix. For instance, $v(3)$ might be 101 or 0101 or 1101, and so on.

Huffman Algorithm

Algorithm 5.2 (Huffman)

The input is a probability measure $(p_1, \ldots, p_s)$ with $s \geq 2$.

- **Begin**. Take the s Huffman sets: $\{(1, \varnothing)\}, \ldots, \{(s, \varnothing)\}$.
- **Step**. Let X and Y be Huffman sets of the least two weights. Let

$$Z = \big\{(i, 0v) : (i, v) \in X\big\} \cup \big\{(j, 1w) : (j, w) \in Y\big\}.$$

  Replace X and Y with Z. [**Note** $1w$ **was mistyped as** $0w$**, please correct in printed notes!**]
- **End**. End when there is only one Huffman set. Its pairs are $\big(1, v(1)\big), \ldots, \big(s, v(s)\big)$ where $v(i)$ is the codeword encoding i.

Example of Huffman Algorithm

Example 5.3

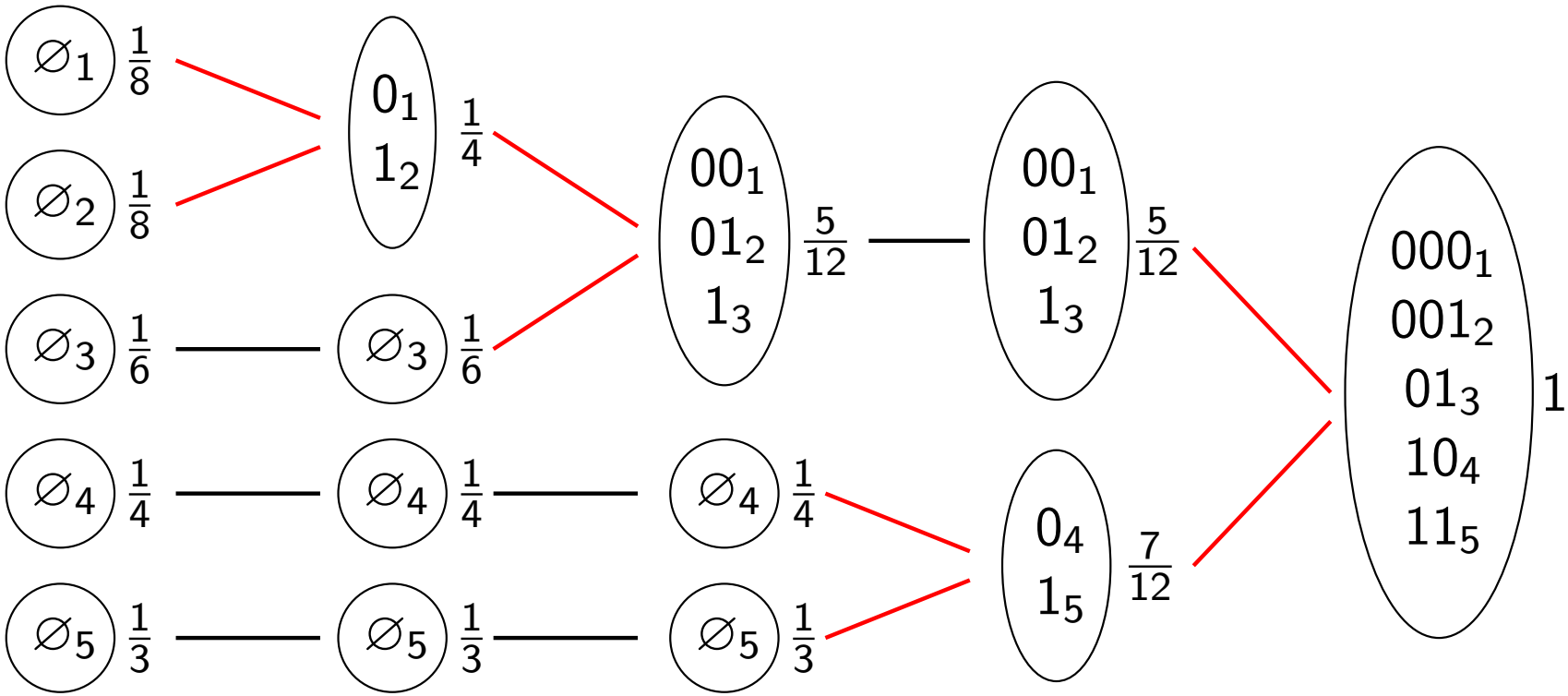
Let $(p_1, p_2, p_3, p_4, p_5) = (\frac{1}{8}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3})$. The table shows the full Huffman algorithm. Note that after step (1) there are two Huffman sets of the second least weight $\frac{1}{4}$. We chose $\{(1,0),(2,1)\}$ (rather than $\{(4,\varnothing)\}$); each of its suffixes 0 and 1 then had 0 prepended.

Begin	$\{(1,\varnothing)\},\{(2,\varnothing)\},\{(3,\varnothing)\},\{(4,\varnothing)\},\{(5,\varnothing)\}$
(1)	$\{(1,0),(2,1)\},\{(3,\varnothing)\},\{(4,\varnothing)\},\{(5,\varnothing)\}$
(2)	$\{(1,00),(2,01),(3,1)\},\{(4,\varnothing)\},\{(5,\varnothing)\}$
(3)	$\{(1,00),(2,01),(3,1)\},\{(4,0),(5,1)\}$
(4)	$\{(1,000),(2,001),(3,01),(4,10),(5,11)\}$
End	$1 \mapsto 000, 2 \mapsto 001, 3 \mapsto 01, 4 \mapsto 10, 5 \mapsto 11$

Example 5.3 [continued]

It is convenient to perform the algorithm by constructing an oriented rooted binary tree, starting with s leaves, and finishing at the root. We follow the same convention from §2 that we step up for 1 and down for 0 (and now horizontally for no change).

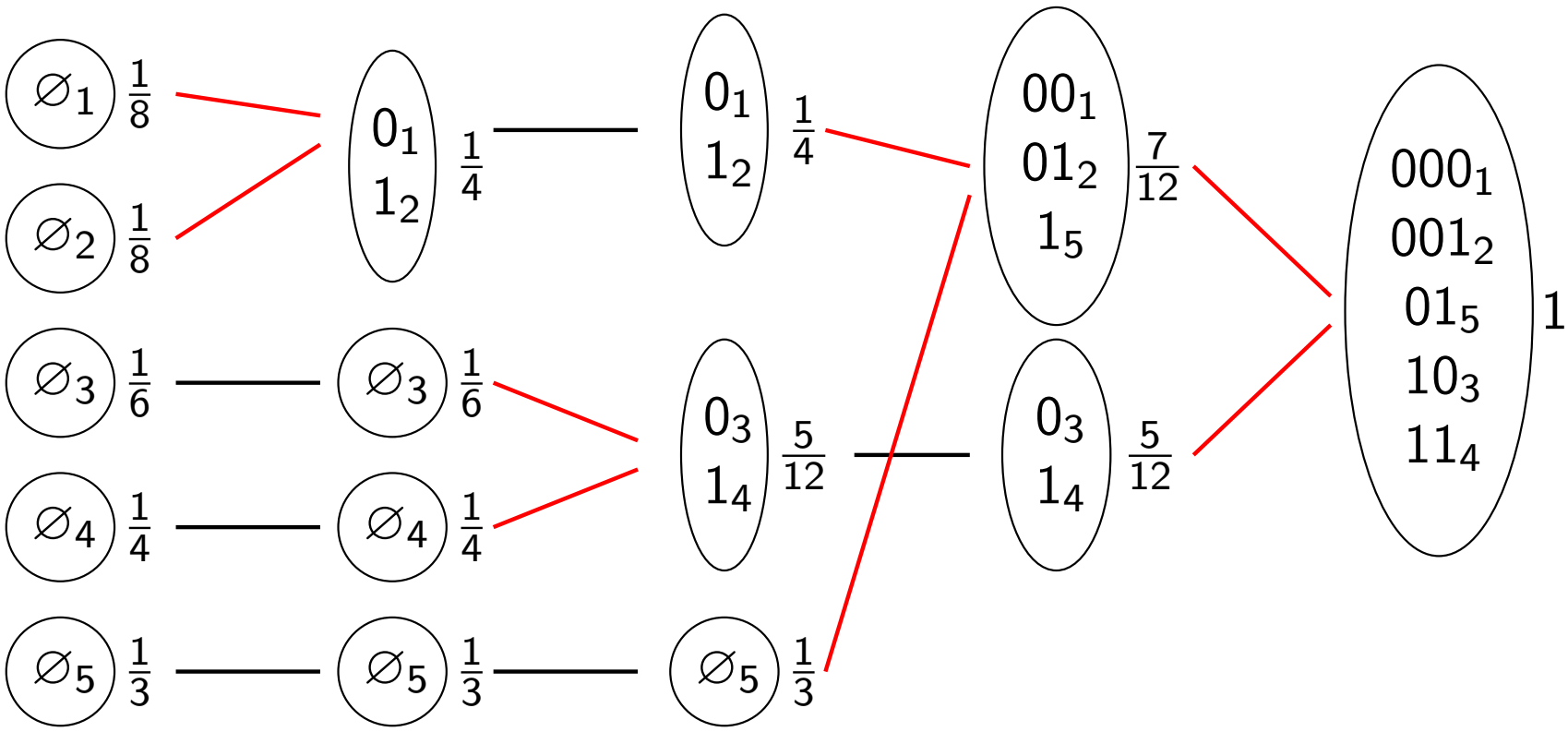
Huffman sets are shown in ellipses with the weight to the right, denoting the pair $(2, \varnothing)$ by $\varnothing_2$ and the pair $(1, 00)$ by 00_1, and so on, to save space.



Choices in the Huffman Algorithm

Exercise 5.4

Use the tree method to construct the Huffman code for the probability distribution $(p_1, p_2, p_3, p_4, p_5) = (\frac{1}{8}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3})$ in Example 5.3 choosing at step (2) the Huffman sets $\{(3, \varnothing)\}$ and $\{(4, \varnothing)\}$. Is there a more efficient code?



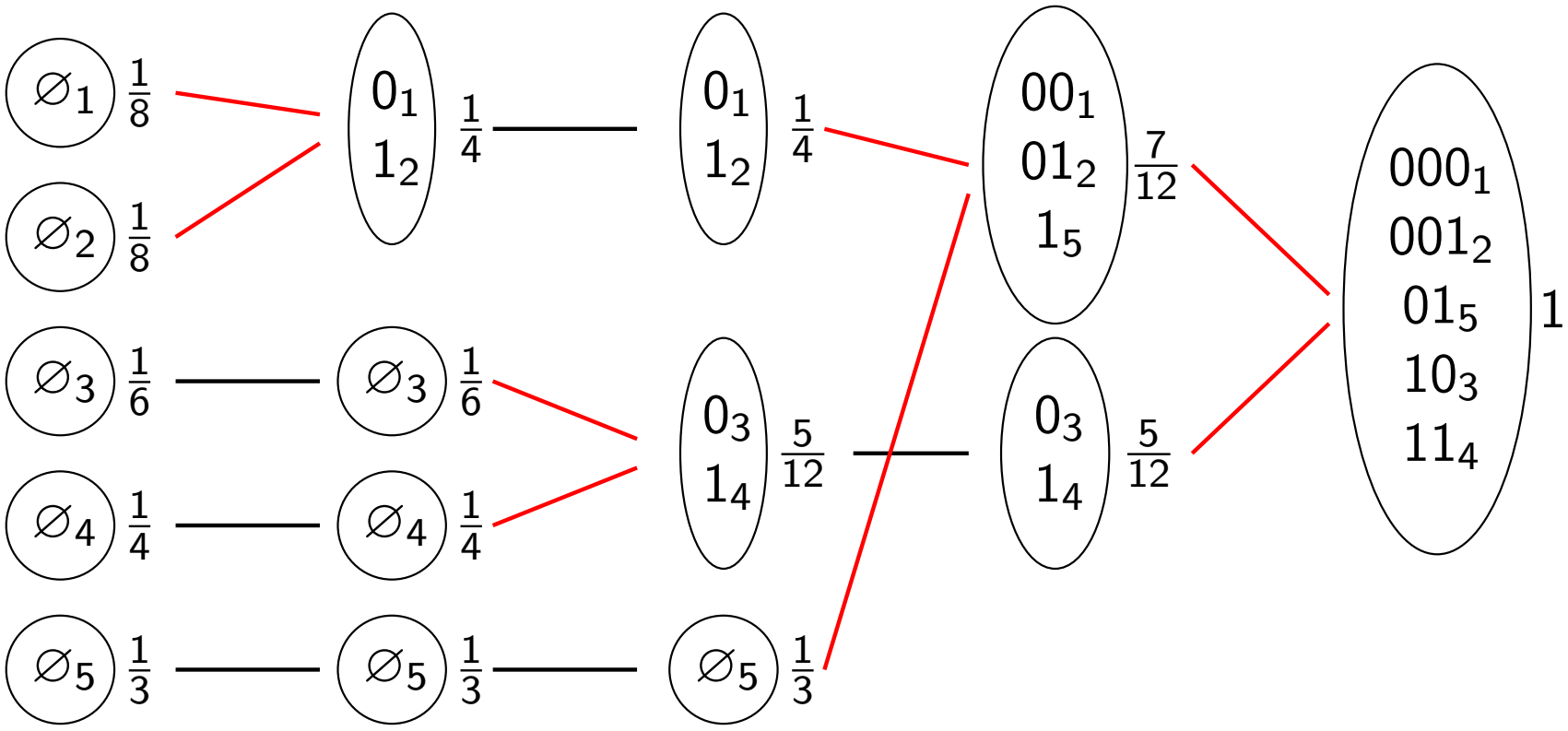
Choices in the Huffman Algorithm

Exercise 5.4

Use the tree method to construct the Huffman code for the probability distribution $(p_1, p_2, p_3, p_4, p_5) = (\frac{1}{8}, \frac{1}{8}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3})$ in Example 5.3 choosing at step (2) the Huffman sets $\{(3, \varnothing)\}$ and $\{(4, \varnothing)\}$. Is there a more efficient code?

Question 5 on Sheet 3 gives a bigger example where choices in the Huffman algorithm lead to codes having different codeword lengths and different maximum length. By Corollary 5.9, these choices do not change the expected codeword length.

Another Interpretation of the Huffman Tree



Optional exercise. Check that reading the tree right-to-left, interpreting steps up (which were steps down from left-to-right) as 0 and steps down (which were steps up from left-to-right) as 1 it becomes the oriented rooted binary tree for the Huffman code.

Convincingly explained, this gives a different proof of Lemma 5.5.

Results on Huffman Codes

Lemma 5.5

Huffman codes are prefix-free

Results on Huffman Codes

Lemma 5.5

Huffman codes are prefix-free

Definition 5.6

We say that the binary code $v(1), \ldots, v(s)$ is *optimal* for the probability measure $(p_1, \ldots, p_s)$ if its prefix-free, and no other prefix-free code with s codewords has a smaller expected length.

We require the following lemma. Part (a) should be intuitive: it simply says that in an optimal code less probable symbols get longer codewords.

Lemma 5.7

In an optimal code for the probability measure $p_1 \leq p_2 \leq \ldots \leq p_s$ *where the codewords* $v(1), v(2), \ldots, v(s)$ *have lengths* $\ell_1, \ell_2, \ldots, \ell_s$,

(a) $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_s$;

(b) $\ell_1 = \ell_2$ *and two of the codewords of this length differ only in their final positions.*

Huffman Codes are Optimal

Lemma 5.7

In an optimal code for the probability measure $p_1 \leq p_2 \leq \ldots \leq p_s$ where the codewords $v(1), v(2), \ldots, v(s)$ have lengths $\ell_1, \ell_2, \ldots, \ell_s$,

(a) $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_s$;

(b) $\ell_1 = \ell_2$ *and two of the codewords of this length differ only in their final positions.*

Proposition 5.8

Suppose that the probability measure $p_1, p_2, p_3, \ldots, p_s$ has an optimal code with expected length $\overline{L}$ and that the probability measure $p_1 + p_2, p_3, \ldots, p_s$ has an optimal prefix-free code with expected length $\overline{M}$. Then

$$\overline{L} \geq \overline{M} + p_1 + p_2.$$

Corollary 5.9

Huffman codes are optimal.

§7 Noisy channels

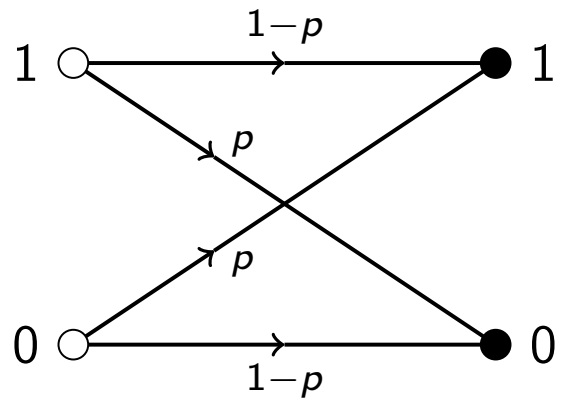
Question. How quickly can we communicate reliably through a noisy channel?

Definition 7.1

Let $\mathcal{A}$ and $\mathcal{B}$ be alphabets. A *discrete memoryless channel* sends a symbol $\alpha \in \mathcal{A}$ to $\beta \in \mathcal{B}$ with a fixed probability $p_{\alpha\beta}$.

Here 'memoryless' is the property that each transmission through the channel is independent of those before.

In the introduction we saw the binary symmetric channel, in which $\mathcal{A} = \mathcal{B} = \{0, 1\}$ and each bit flips independent with probability p.



Further Examples of Channels

Denote by X the input symbol and Y the output symbol. Thus

$$\mathbb{P}[Y = \beta | X = \alpha] = p_{\alpha\beta}$$

for all $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$.

Example 7.2

(1) In the *binary erasure channel* with *erasure probability* p, $\mathcal{A} = \{0, 1\}$ and $\mathcal{B} = \{0, 1, \star\}$. Each sent bit is received correctly with probability $1 - p$, and otherwise erased by the channel: we model this by supposing that the special symbol $\star$ is received. (Thus the receiver knows a bit was sent, but not what it is.)

The matrix of channel probabilities is

$$\begin{array}{c} 0 \\ 1 \end{array} \begin{pmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{pmatrix}$$

Example 7.2 continued

(2) The *lazy typist* channel with s symbols has $\mathcal{A} = \mathcal{B} = \{0, 1, \ldots, s-1\}$. The transition probabilities are specified by

$$\mathbb{P}[Y = x | X = x] = \tfrac{1}{2}, \quad \mathbb{P}[Y = x + 1 \bmod s | X = x] = \tfrac{1}{2}.$$

Exercise: find the matrix of channel probabilities when $s = 4$.

Example 7.2 continued

(2) The *lazy typist* channel with s symbols has $\mathcal{A} = \mathcal{B} = \{0, 1, \ldots, s-1\}$. The transition probabilities are specified by

$$\mathbb{P}[Y = x | X = x] = \tfrac{1}{2}, \quad \mathbb{P}[Y = x + 1 \bmod s | X = x] = \tfrac{1}{2}.$$

Exercise: find the matrix of channel probabilities when $s = 4$.

Observe that in each case the matrix with entries $p_{\alpha\beta}$ is stochastic, i.e. its rows all sum to 1.

Exercise 7.3

Take $s = 4$ in the lazy typist channel, so the input and output alphabets are $\{0, 1, 2, 3\}$.

(a) Find a way to encode the four messages $\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}$ so that the receiver can decode with zero probability of error.

(b) Specify the decoding rule as a function from words in the output symbols $\{0, 1, 2, 3\}$ to $\{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}$.

(c) For each message sent, how many symbols are required? For a perfect typist, how many symbols are required per message?

Conditional Entropy

Definition 7.4

Let X and Y be random variables taking values in finite sets $\mathcal{A}$ and $\mathcal{B}$, respectively. The *conditional entropy of X given that $Y = \beta$* is defined by

$$H(X|Y=\beta) = -\sum_{\alpha\in\mathcal{A}} \mathbb{P}[X=\alpha|Y=\beta]\log_2 \mathbb{P}[X=\alpha|Y=\beta].$$

The *conditional entropy of X given Y* is defined by

$$H(X|Y) = \sum_{\beta\in\mathcal{B}} \mathbb{P}[Y=\beta]H(K|Y=y).$$

Conditional Entropy Exercise

$$H(X|Y=\beta) = -\sum_{\alpha\in\mathcal{A}} \mathbb{P}[X=\alpha|Y=\beta]\log_2 \mathbb{P}[X=\alpha|Y=\beta]$$

$$H(X|Y) = \sum_{\beta\in\mathcal{B}} \mathbb{P}[Y=\beta]H(X|Y=\beta).$$

Exercise 7.5

Fix $s \in \mathbb{N}$. Take the lazy typist channel with input and output alphabets $\mathcal{A} = \mathcal{B} = \{0, 1, \ldots, s-1\}$. As usual, let X be the input symbol and let Y be the output symbol. Suppose that X is uniformly distributed on $\{0, 1, \ldots, s-1\}$, so $\mathbb{P}[X = x] = \frac{1}{s}$ for each x. Find

$$H(X), H(Y), H(X|Y=0), H(X|Y), H(X,Y).$$

Now suppose that $s = 4$ and X is 0 with probability p and 1 with probability $1-p$. Find

(a) $\mathbb{P}[Y=1|X=0], H(Y|X=0), H(Y|X), \mathbb{P}[Y=1], H(Y)$.

(b) Find $\mathbb{P}[X=0|Y=1], H(X|Y=1), H(X|Y)$.

Conditional Entropy Exercise

$$H(X|Y=\beta) = -\sum_{\alpha\in\mathcal{A}} \mathbb{P}[X=\alpha|Y=\beta]\log_2 \mathbb{P}[X=\alpha|Y=\beta]$$

$$H(X|Y) = \sum_{\beta\in\mathcal{B}} \mathbb{P}[Y=\beta]H(X|Y=\beta).$$

Here is a summary of what we found in the second case $s=4$ and $\mathbb{P}[X=0]=p$, $\mathbb{P}[X=1]=1-p$. For (a)

- $\mathbb{P}[Y=1|X=0]=\frac{1}{2}$ and $\mathbb{P}[Y=0|X=0]=\frac{1}{2}$
- $H(Y|X=0)=H(\frac{1}{2},\frac{1}{2})=1$
- $H(Y|X=1)=H(\frac{1}{2},\frac{1}{2})=1$
- $H(Y|X)=pH(Y|X=0)+(1-p)H(Y|X=1)=1$
- $\mathbb{P}[Y=1]=\mathbb{P}[Y=1|X=0]\mathbb{P}[X=0]+\mathbb{P}[Y=1|X=1]\mathbb{P}[x=1]$. Note use of conditioning argument. Hence

$$\mathbb{P}[Y=1]=\tfrac{1}{2}p+\tfrac{1}{2}(1-p)=\tfrac{1}{2}.$$

- $\mathbb{P}[Y=0]=\frac{1}{2}p$ and $\mathbb{P}[Y=2]=\frac{1}{2}(1-p)$
- $H(Y)=H(\frac{1}{2}p,\frac{1}{2},\frac{1}{2}(1-p))=1+\frac{1}{2}H(p,1-p)$.

Conditional Entropy Exercise

$$H(X|Y=\beta) = -\sum_{\alpha\in\mathcal{A}} \mathbb{P}[X=\alpha|Y=\beta]\log_2 \mathbb{P}[X=\alpha|Y=\beta]$$

$$H(X|Y) = \sum_{\beta\in\mathcal{B}} \mathbb{P}[Y=\beta]H(X|Y=\beta).$$

Here is a summary of what we found in the second case $s=4$ and $\mathbb{P}[X=0]=p$, $\mathbb{P}[X=1]=1-p$. For (b)

- $\mathbb{P}[X=0|Y=1] = \dfrac{\mathbb{P}[Y=1|X=0]\mathbb{P}[X=0]}{\mathbb{P}[Y=1]} = \dfrac{\frac{1}{2}p}{\frac{1}{2}} = p.$
- $\mathbb{P}[X=1|Y=1] = 1-p$. (Complementary event.)
- $\mathbb{P}[X=0|Y=0] = 1$; note this is not $\mathbb{P}[X=0]$.
- $\mathbb{P}[X=1|Y=2] = 1$; note this is not $\mathbb{P}[X=1]$.
- $H(X|Y=1) = H(p,1-p)$ and
$$H(X|Y=0) = H(X|Y=2) = 0.$$
- $H(X|Y) = \mathbb{P}[Y=1]H(X|Y=1) = \frac{1}{2}H(p,1-p)$.

Conditional Entropy Exercise

$$H(X|Y=\beta) = -\sum_{\alpha\in\mathcal{A}} \mathbb{P}[X=\alpha|Y=\beta]\log_2 \mathbb{P}[X=\alpha|Y=\beta]$$

$$H(X|Y) = \sum_{\beta\in\mathcal{B}} \mathbb{P}[Y=\beta]H(X|Y=\beta).$$

Cheat Sheet!

- $\mathbb{P}[Y=y|X=x]$ is a channel probability: it depends only on the channel matrix.
- $\mathbb{P}[X=x|Y=y]$ must be computed using conditional probability and depends both on the channel matrix and the distribution of X.

Chaining Rule

Lemma 7.6 (Chaining Rule)

Let X and Y be random variables. Then

$$H(X|Y) + H(Y) = H(X, Y).$$

If you are doing MT361/461/5461 you will have seen a proof in this course. Otherwise please do Question 5 on Sheet 4. (Each step is quite small: there will of course be a model answer.)

Chaining Rule

Lemma 7.6 (Chaining Rule)

Let X and Y be random variables. Then

$$H(X|Y) + H(Y) = H(X, Y).$$

In Example 7.5 we saw that when $s = 4$ and $\mathbb{P}[X = 0] = p$, $\mathbb{P}[X = 1] = 1 - p$ we have

$$H(Y) = H\left(\tfrac{1}{2}p, \tfrac{1}{2}, \tfrac{1}{2}(1-p)\right) = 1 + \tfrac{1}{2}H(p, 1-p)$$
$$H(X|Y) = \tfrac{1}{2}H(p, 1-p)$$

Hence $H(X, Y) = H(X|Y) + H(Y) = 1 + H(p, 1-p)$.

Two symmetries are worth noting:

- Compute $H(X, Y)$ using the Chaining Rule the other way round, using

$$H(X) = H(p, 1-p)$$
$$H(Y|X) = 1$$

- How much information does learning Y give about X? How much information does learning X give about Y?

Mutual Information

Definition 7.7

The *mutual information* of random variables X and Y is

$$I(X;Y) = H(X) - H(X|Y).$$

Since $H(X|Y)$ is the uncertainty in X *after we have learned* Y, we have the following interpretation.

> $I(X;Y)$ is the amount of information that Y tells us about X.

In Question 4 on Sheet 4 you are asked to show that $H(X) \geq H(X|Y)$ with equality if and only if X and Y are independent. Hence $I(X;Y) \geq 0$, with equality if and only if X and Y are independent.

Exercise 7.8

Since entropies are positive, $I(X;Y) \leq H(X)$. When does $I(X;Y) = H(X)$ hold?

Mutual Information

Definition 7.7

The *mutual information* of random variables X and Y is

$$I(X;Y) = H(X) - H(X|Y).$$

Since $H(X|Y)$ is the uncertainty in X *after we have learned* Y, we have the following interpretation.

> $I(X;Y)$ is the amount of information that Y tells us about X.

Example 7.9

Let X be the roll of a fair die and let Y be the answer to the question 'Did you roll 1 or 2?'. Then

$$H(X|Y) = \tfrac{1}{3}\log_2 2 + \tfrac{2}{3}\log_2 4 = \tfrac{5}{3}$$

and so $I(X;Y) = H(X) - H(X|Y) = \log_2 6 - \tfrac{5}{3}$.

Other Formulae for Mutual Information

By the chaining rule $H(X|Y) + H(Y) = H(X, Y)$. Therefore the mutual information can be written in the more symmetric form

$$I(X; Y) = H(X) + H(Y) - H(X, Y).$$

Exercise 7.10

Deduce that $I(X; Y) = H(Y) - H(Y|X)$ and hence that $I(X; Y) = I(Y; X)$.

It is perhaps a little surprising that the mutual information is symmetric in X and Y. *Exercise:* check this for Example 7.9. This fact necessary to justify calculating with conditional entropy and mutual information using Venn diagrams.

Other Formulae for Mutual Information

By the chaining rule $H(X|Y) + H(Y) = H(X, Y)$. Therefore the mutual information can be written in the more symmetric form

$$I(X; Y) = H(X) + H(Y) - H(X, Y).$$

Exercise 7.10

Deduce that $I(X; Y) = H(Y) - H(Y|X)$ and hence that $I(X; Y) = I(Y; X)$.

It is perhaps a little surprising that the mutual information is symmetric in X and Y. *Exercise:* check this for Example 7.9. This fact necessary to justify calculating with conditional entropy and mutual information using Venn diagrams.

For noisy channels, the probabilities $\mathbb{P}[Y = \beta|X = \alpha] = p_{\alpha\beta}$ are given by the channel. In contrast, $\mathbb{P}[X = \alpha|Y = \beta]$ has to be calculated using conditional probability (or Bayes' Law). We saw this in Exercise 7.5,

So it will often be useful to use $I(X; Y) = H(Y) - H(Y|X)$.

Mutual Information for a Noisy Channel

Example 7.11

Let X and Y be the input and output symbols in the lazy typist channel.

(a) By Exercise 7.5, if all s input symbols are equiprobable then $I(X;Y) = \log_2 s - 1$.

(b) Let s be even. Suppose that

$$p_\alpha = \begin{cases} \frac{2}{s} & \text{if } \alpha \text{ is even} \\ 0 & \text{if } \alpha \text{ is odd.} \end{cases}.$$

Then Y is uniformly distributed so $H(Y) = \log_2 s$ and $I(X;Y) = H(Y) - H(Y|X) = \log_2 s - 1$.

(c) Suppose that $s = 4$ and $p_0 = p_1 = \frac{1}{2}$. Then Y has probability distribution $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}, 0)$ and $H(Y) = \frac{3}{2}$. We have $I(X;Y) = \frac{3}{2} - 1 = \frac{1}{2}$. The maximum value of $I(X;Y)$ is 1; by (b) the maximum is attained for $p = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ and $p = \frac{1}{4}), (\frac{1}{2}, 0, \frac{1}{2}, 0)$. *Exercise:* find another probability measure on X that maximizes $I(X;Y)$.

Capacity

As usual let X be the input symbol and let Y be the output symbol

Definition 7.12

The *capacity* of a channel is $\max_p I(X;Y)$ where the maximum is taken over all probability measures p on the input symbol X.

Example 7.13

(a) Let $p \leq 1/2$. The capacity of the Binary Symmetric Channel with error probability p is $1 - H(p, 1-p)$.

(b) The capacity of the Binary Erasure Channel with erasure probability p is $1 - p$. You are asked to show this on Problem Sheet 5. *Exercise:* draw a graph comparing (a) and (b).

(c) We saw in Example 7.11 that for the Noisy Typist Channel with $s = 4$, the maximum of $I(X;Y)$ is 1. In general the maximum is $\log_2 s - 1$; the proof is almost the same, replacing 4 with s.

Theorem 7.14 (Shannon's Noisy Coding Theorem for Discrete Memoryless Channels)

Fix a discrete memoryless channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$ of capacity c.

(a) *Let $\epsilon > 0$ be given. For every $r < c$ there exists $n \in \mathbb{N}$ and a code $C \subseteq \mathcal{A}^n$ such that $|C| \geq 2^{rn}$ and the error probability when C is used to send codewords through the channel is less than ϵ.*

(b) *If $r > c$ then, when n is large, it is impossible to find a code as in (a).*

Theorem 7.14 (Shannon's Noisy Coding Theorem for Discrete Memoryless Channels)

Fix a discrete memoryless channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$ of capacity c.

(a) *Let $\epsilon > 0$ be given. For every $r < c$ there exists $n \in \mathbb{N}$ and a code $C \subseteq \mathcal{A}^n$ such that $|C| \geq 2^{rn}$ and the error probability when C is used to send codewords through the channel is less than ϵ.*

(b) *If $r > c$ then, when n is large, it is impossible to find a code as in (a).*

- In fact we will have $|C| \approx 2^{rn}$: the inequality is necessary only because 2^{rn} may not be an integer.
- The 'error probability' in (a) for a codeword $u \in C$ is the probability that when u is sent through the channel, and v is received, v is not decoded as u. The claim in (a) is that, by choosing the code and decoding rule suitably, we can make all these probabilities $< \epsilon$.

Theorem 7.14 (Shannon's Noisy Coding Theorem for Discrete Memoryless Channels)

Fix a discrete memoryless channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$ of capacity c.

(a) *Let $\epsilon > 0$ be given. For every $r < c$ there exists $n \in \mathbb{N}$ and a code $C \subseteq \mathcal{A}^n$ such that $|C| \geq 2^{rn}$ and the error probability when C is used to send codewords through the channel is less than ϵ.*

(b) *If $r > c$ then, when n is large, it is impossible to find a code as in (a).*

Example 7.15

Take the lazy typist channel on $\{0, 1, 2, 3\}$. The capacity of the channel is 1 by Example 7.11. In Example 7.3 we used the encoder

$$\mathtt{A} \mapsto 00, \mathtt{T} \mapsto 02, \mathtt{G} \mapsto 20, \mathtt{C} \mapsto 22.$$

and decoder $00, 01, 10, 11 \mapsto \mathtt{A}$, and so on. We will generalize this example to find a suitable code C *and decoding rule* proving that (a) in Shannon's Noisy Coding Theorem holds. In this special case we can even take $r = c$ and n does not need to be large.

§8 Nearest neighbour decoding

Question. What decoding rule minimizes the probability of decoding error?

Definition 8.1

Let $\mathcal{A}$ be an alphabet. Let $u, v \in \mathcal{A}^n$ be words of length n. The *Hamming distance* between u and v, denoted $d(u, v)$, is the number of positions in which u and v are different.

In mathematical notation, $d(u, v) = \left|\left\{i \in \{1, 2, \ldots, n\} : u_i \neq v_i\right\}\right|$.
We will often abbreviate 'Hamming distance' to '*distance*'.

Example 8.2

Working with binary words of length 4, we have $d(0011, 1101) = 3$ because the words 0011 and 1101 differ in their first three positions, and are the same in their final position. Working with words over the alphabet $\{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\}$, we have $d(\mathtt{tale}, \mathtt{take}) = 1$ and $d(\mathtt{tale}, \mathtt{tilt}) = 2$.

Properties of Hamming Distance

Theorem 8.3

Let $\mathcal{A}$ be an alphabet and let $u, v, w \in \mathcal{A}^n$.

(a) $d(u, v) = 0$ *if and only if* $u = v$;

(b) $d(u, v) = d(v, u)$;

(c) $d(u, w) \leq d(u, v) + d(v, w)$.

Properties of Hamming Distance

Theorem 8.3

Let $\mathcal{A}$ be an alphabet and let $u, v, w \in \mathcal{A}^n$.

(a) $d(u, v) = 0$ *if and only if* $u = v$;

(b) $d(u, v) = d(v, u)$;

(c) $d(u, w) \leq d(u, v) + d(v, w)$.

Part (c) is called the *triangle inequality*. As an exercise, find all English words v such that

$$d(\texttt{warm}, v) = d(\texttt{cold}, v) = 2.$$

Check that the triangle inequality holds when u, v, w are `warm`, `wall`, `cold`, respectively.

Properties of Hamming Distance

Theorem 8.3

Let $\mathcal{A}$ be an alphabet and let $u, v, w \in \mathcal{A}^n$.

(a) $d(u, v) = 0$ *if and only if* $u = v$;

(b) $d(u, v) = d(v, u)$;

(c) $d(u, w) \leq d(u, v) + d(v, w)$.

Part (c) is called the *triangle inequality*. As an exercise, find all English words v such that

$$d(\texttt{warm}, v) = d(\texttt{cold}, v) = 2.$$

Check that the triangle inequality holds when u, v, w are `warm`, `wall`, `cold`, respectively.

If you have seen metric spaces then you will probably have noticed that Theorem 8.3 says that $(\mathcal{A}^n, d)$ is a metric space.

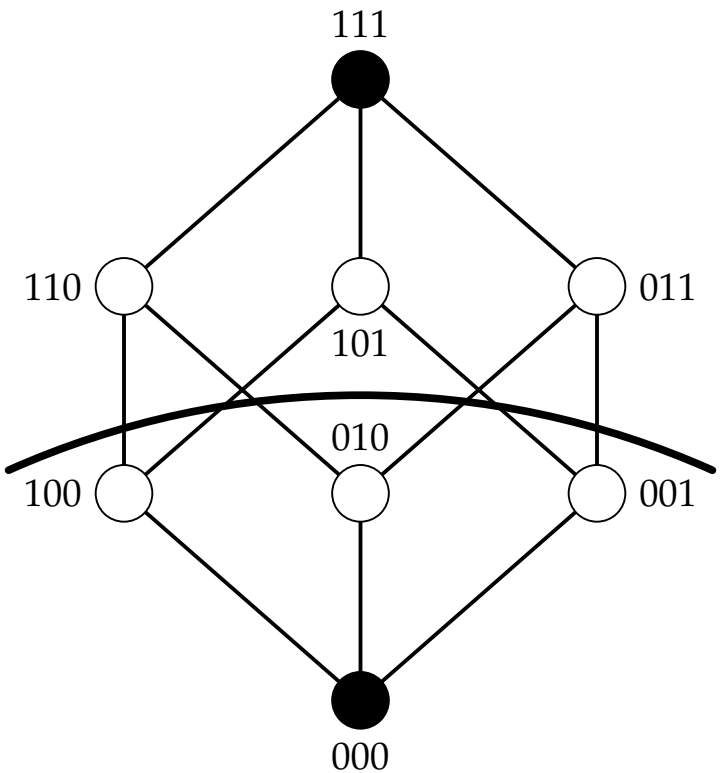
Error Probabilities for the Binary Symmetric Channel

In Exercise 1.9 Alice sent Bob a codeword $X \in \{000, 111\}$ across the Binary Symmetric Channel with error probability p, and Bob received $Y \in \{0, 1\}^3$. We saw that $\mathbb{P}[Y = 111|X = 000] = p^3$, $\mathbb{P}[Y = 110|X = 000] = p^2(1 - p)$, and so on.

Error Probabilities for the Binary Symmetric Channel

In Exercise 1.9 Alice sent Bob a codeword $X \in \{000, 111\}$ across the Binary Symmetric Channel with error probability p, and Bob received $Y \in \{0,1\}^3$. We saw that $\mathbb{P}[Y = 111|X = 000] = p^3$, $\mathbb{P}[Y = 110|X = 000] = p^2(1-p)$, and so on.

Generally the power of p in $\mathbb{P}[Y = v|X = u]$ is the number of bits flipped by the channel. This is the Hamming distance $d(u, v)$. It is also the number of edges between u and v in the graph seen in §1.



Error Probabilities for the Binary Symmetric Channel

In Exercise 1.9 Alice sent Bob a codeword $X \in \{000, 111\}$ across the Binary Symmetric Channel with error probability p, and Bob received $Y \in \{0,1\}^3$. We saw that $\mathbb{P}[Y = 111|X = 000] = p^3$, $\mathbb{P}[Y = 110|X = 000] = p^2(1-p)$, and so on.

Lemma 8.4

Suppose that $u \in \{0,1\}^n$ is sent through the $\mathrm{BSC}(p)$. The probability that $v \in \{0,1\}^n$ is received is $p^{d(u,v)}(1-p)^{n-d(u,v)}$.

Theorem 8.5

Suppose that we use a binary code C of length n to send messages through the $\mathrm{BSC}(p)$ with $p < 1/2$, and that each codeword in C is equally likely to be sent. Let X be the sent codeword and Y the received word. For each $u \in C$,

$$\mathbb{P}[X = u|Y = v] = p^{d(u,v)}(1-p)^{n-d(u,v)}c(v).$$

where $c(v)$ does not depend on u. Hence $\mathbb{P}[X = u|Y = v]$ is maximized by choosing u to be the nearest codeword to v.

Maximum Likelihood Decoding

In Theorem 8.5 we decode to maximize the *likelihood* that we are correct, so we choose X to maximize $\mathbb{P}[X = u | Y = v]$. Here

- $Y = v$ is the event we observe:
- $X = u$ is our inference.

Maximum Likelihood Decoding

In Theorem 8.5 we decode to maximize the *likelihood* that we are correct, so we choose X to maximize $\mathbb{P}[X = u | Y = v]$. Here

- $Y = v$ is the event we observe:
- $X = u$ is our inference.

The assumption that every codeword is equally likely to be sent was vital to Theorem 8.5.

For instance, in Question 5 on Sheet 1, we saw that if 000 has probability $\frac{1}{10}$ and 111 has probability $\frac{9}{10}$ then,

$$\frac{\mathbb{P}[X = 000 | Y = 000]}{\mathbb{P}[X = 111 | Y = 000]} = \frac{27}{77} \approx 0.351.$$

When 000 is received, it is about twice as likely 111 was sent than 000. Using maximum likelihood decoding it is never correct to decode as 000, and the channel is useless.

Nearest Neighbour Decoding

Definition 8.6 (Nearest neighbour decoding)

Let $C \subseteq \mathcal{A}^n$ be a code. Suppose that a codeword is sent through the channel and we receive the word v. To decide v using *nearest neighbour decoding* look at all the codewords of C and pick the one that is nearest, in Hamming distance to v, choosing arbitrarily if there are several equally close.

Exercise 8.7

Take the code $C = \{00000, 11100, 00111, 11011\}$ from Question 4 on Sheet 5.

(a) Using C on the Binary Symmetric Channel the alphabet is $\{0, 1\}$. Decode the received words 00000, 01111, 01010.

(b) Using C on the Binary Erasure Channel the alphabet is $\{0, \star, 1\}$. Here, as usual, the codewords are in $\{0, 1\}^5$, so the erasure symbol $\star$ appears only in received words. Decode the received words above and $0000\star$, $000\star\star$, $00\star\star\star$.

Connection with Source Coding

After source coding by a good code, such as a Huffman code, we could expect most binary words to occur roughly equally often. We saw at the end of §6 that encoding the first chapter of *Persuasion*, by the optimal Huffman code gives the sequence

$$001111000\,110101\,110111\,0101\,111\,101101\,011\,0000\,\ldots$$

corresponding to $\mathtt{S} \mapsto 001111000$, $\mathtt{u} \mapsto 110101$, $\mathtt{c} \mapsto 110111$, $\mathtt{h} \mapsto 0101$, $\mathtt{\sqcup} \mapsto 111$, and so on.

Quiz: suppose there are no errors in the channel, so Bob receives

$$00111100011010111011101011111011010110000.$$

- Q: How does Bob know how to decode?

- Q: Why is there a unique way to decode?

Connection with Source Coding

After source coding by a good code, such as a Huffman code, we could expect most binary words to occur roughly equally often. We saw at the end of §6 that encoding the first chapter of *Persuasion*, by the optimal Huffman code gives the sequence

$$001111000\,110101\,110111\,0101\,111\,101101\,011\,0000\,\ldots$$

corresponding to $\mathtt{S} \mapsto 001111000$, $\mathtt{u} \mapsto 110101$, $\mathtt{c} \mapsto 110111$, $\mathtt{h} \mapsto 0101$, $\mathtt{\sqcup} \mapsto 111$, and so on.

Quiz: suppose there are no errors in the channel, so Bob receives

$$00111100011010111011101011111011010110000.$$

- Q: How does Bob know how to decode?
  A: The code is not secret!
- Q: Why is there a unique way to decode?

Connection with Source Coding

After source coding by a good code, such as a Huffman code, we could expect most binary words to occur roughly equally often. We saw at the end of §6 that encoding the first chapter of *Persuasion*, by the optimal Huffman code gives the sequence

$$001111000\,110101\,110111\,0101\,111\,101101\,011\,0000\,\ldots$$

corresponding to $\mathtt{S} \mapsto 001111000$, $\mathtt{u} \mapsto 110101$, $\mathtt{c} \mapsto 110111$, $\mathtt{h} \mapsto 0101$, $\mathtt{\sqcup} \mapsto 111$, and so on.

Quiz: suppose there are no errors in the channel, so Bob receives

$$00111100011010111011101011111011010110000.$$

- Q: How does Bob know how to decode?
  A: The code is not secret!
- Q: Why is there a unique way to decode?
  A: The Huffman code is prefix free.

Equally Likely Codewords

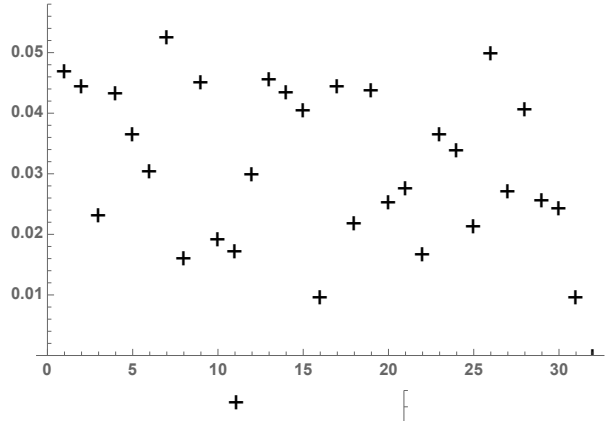
The graphs below show the probability of seeing each binary word of length 5 in the source coding of the first chapter of *Persuasion* by

(1) The 8-bit ASCII code;

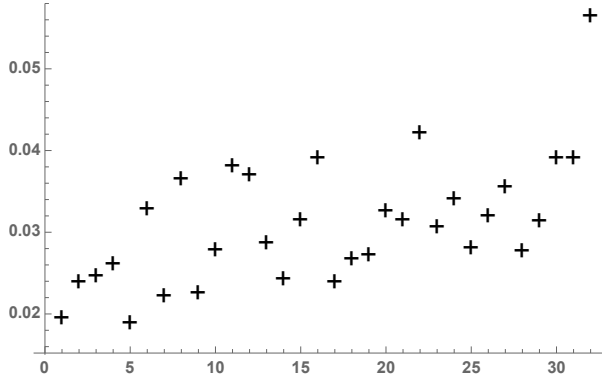
(2) The optimal Huffman character on characters above;

(3) The optimal Huffman code on pairs of characters.

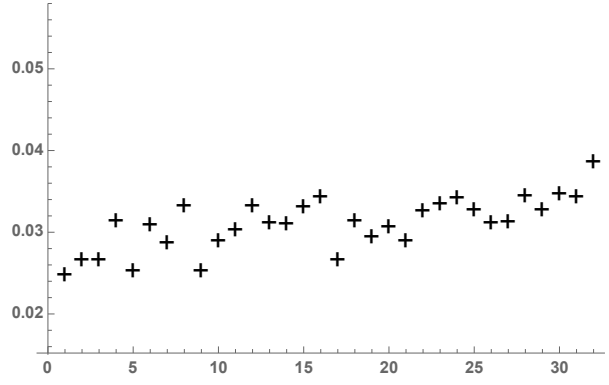
(1)



(2)



(3)



Hamming Balls

Definition 8.8

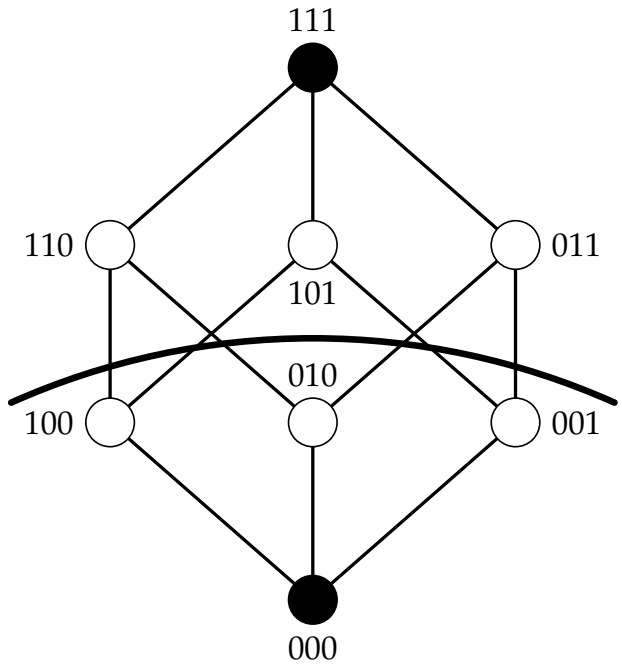
Let $\mathcal{A}$ be an alphabet and let $u \in \mathcal{A}^n$. The *Hamming ball* of *radius* r about u is the set

$$B_r(u) = \{v \in \mathcal{A}^n : d(u, v) \leq r\}.$$

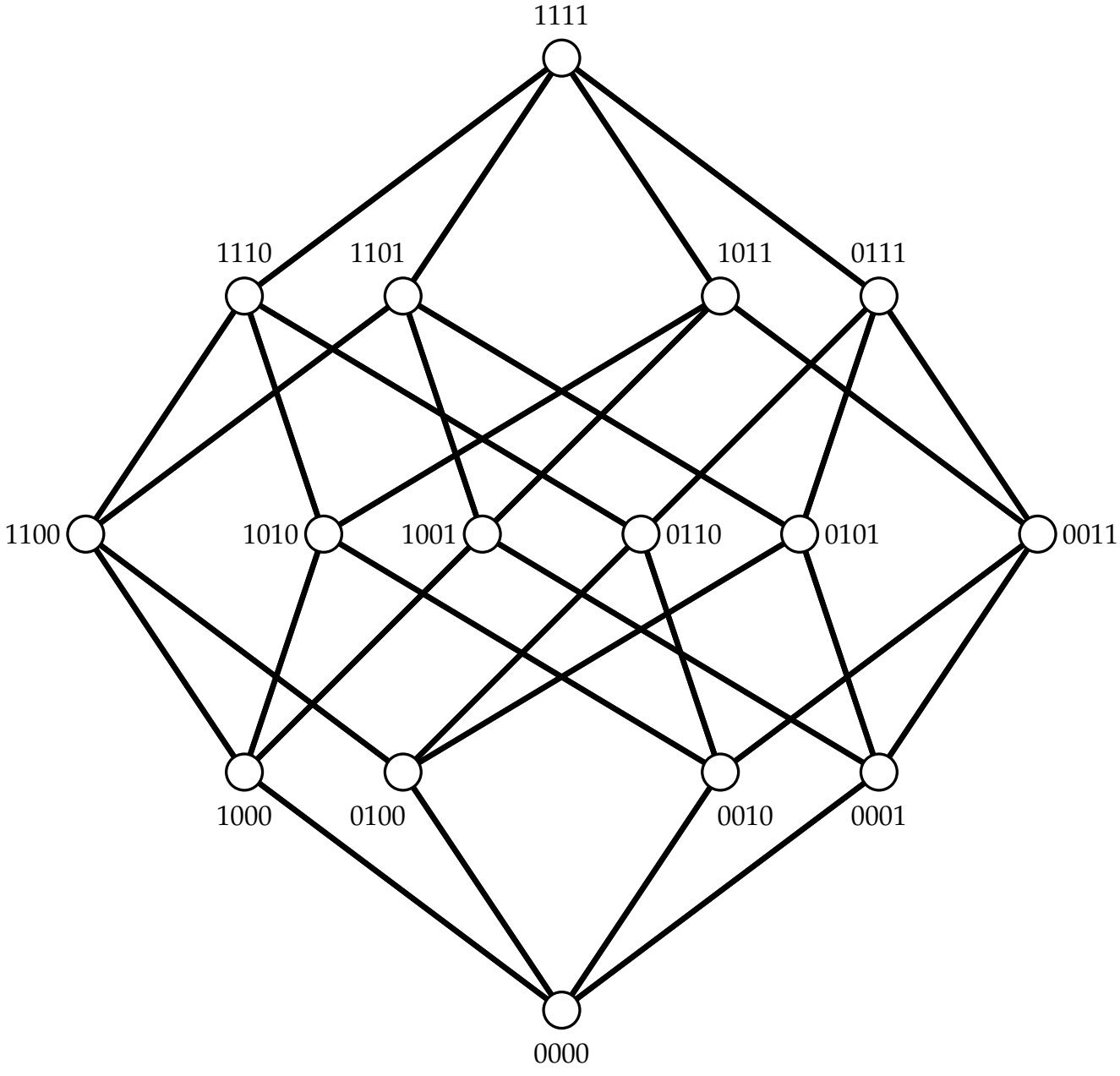
We saw earlier that when $\mathcal{A} = \{0, 1\}$ and $n = 3$, the words in $B_1(000) = \{000, 100, 010, 001\}$ decode to 000 using nearest neighbour decoding with the repetition code $\{000, 111\}$.

The repetition code can correct 1 error in the Binary Symmetric Channel because the Hamming balls of radius 1 about the codewords are disjoint.

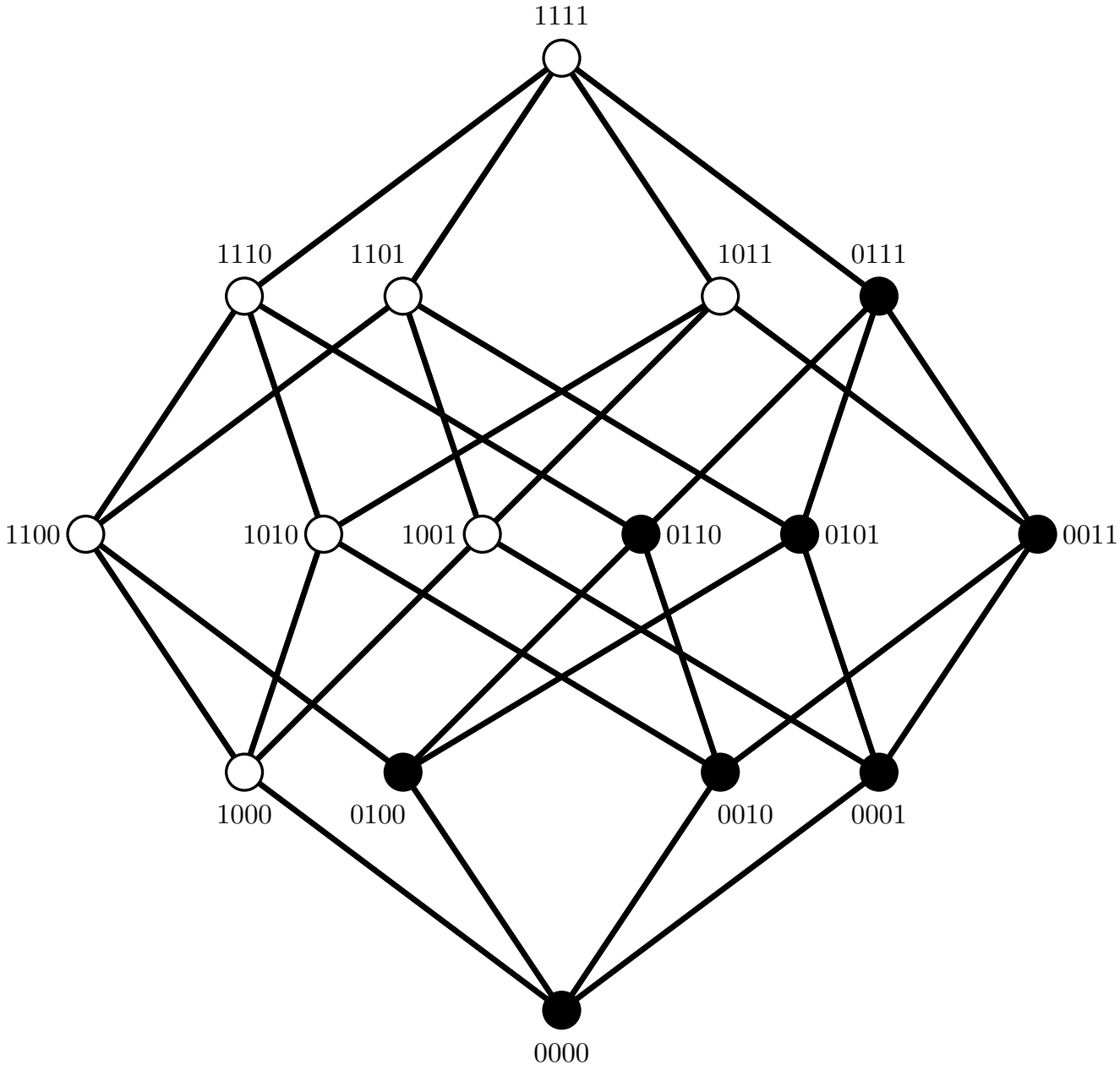
See Question 4 on Sheet 5 for a similar example.



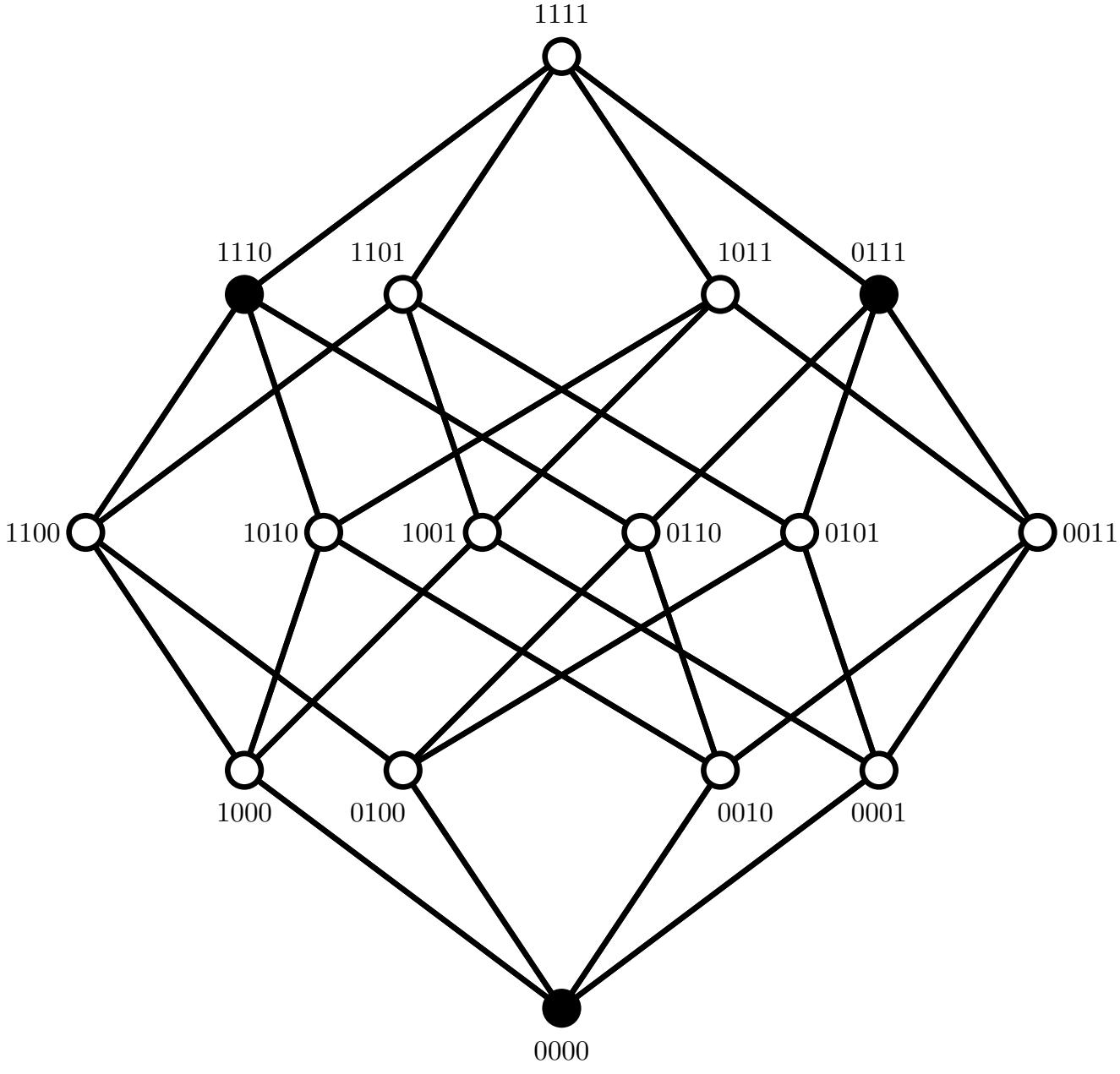
$\{0,1\}^4$ with Hamming Distance



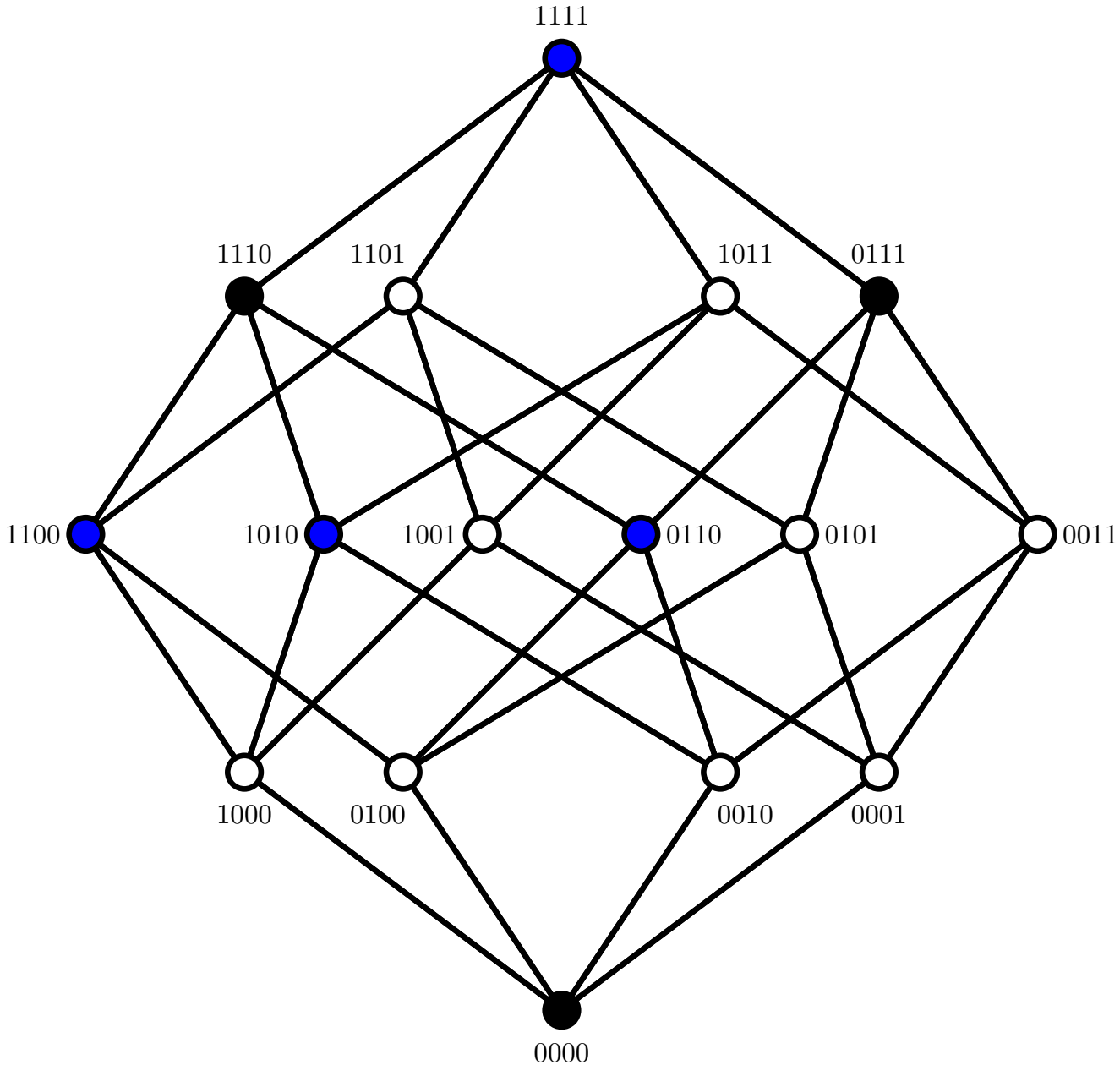
$$\left\{v \in \{0,1\}^4 : v_1 = 0\right\}$$



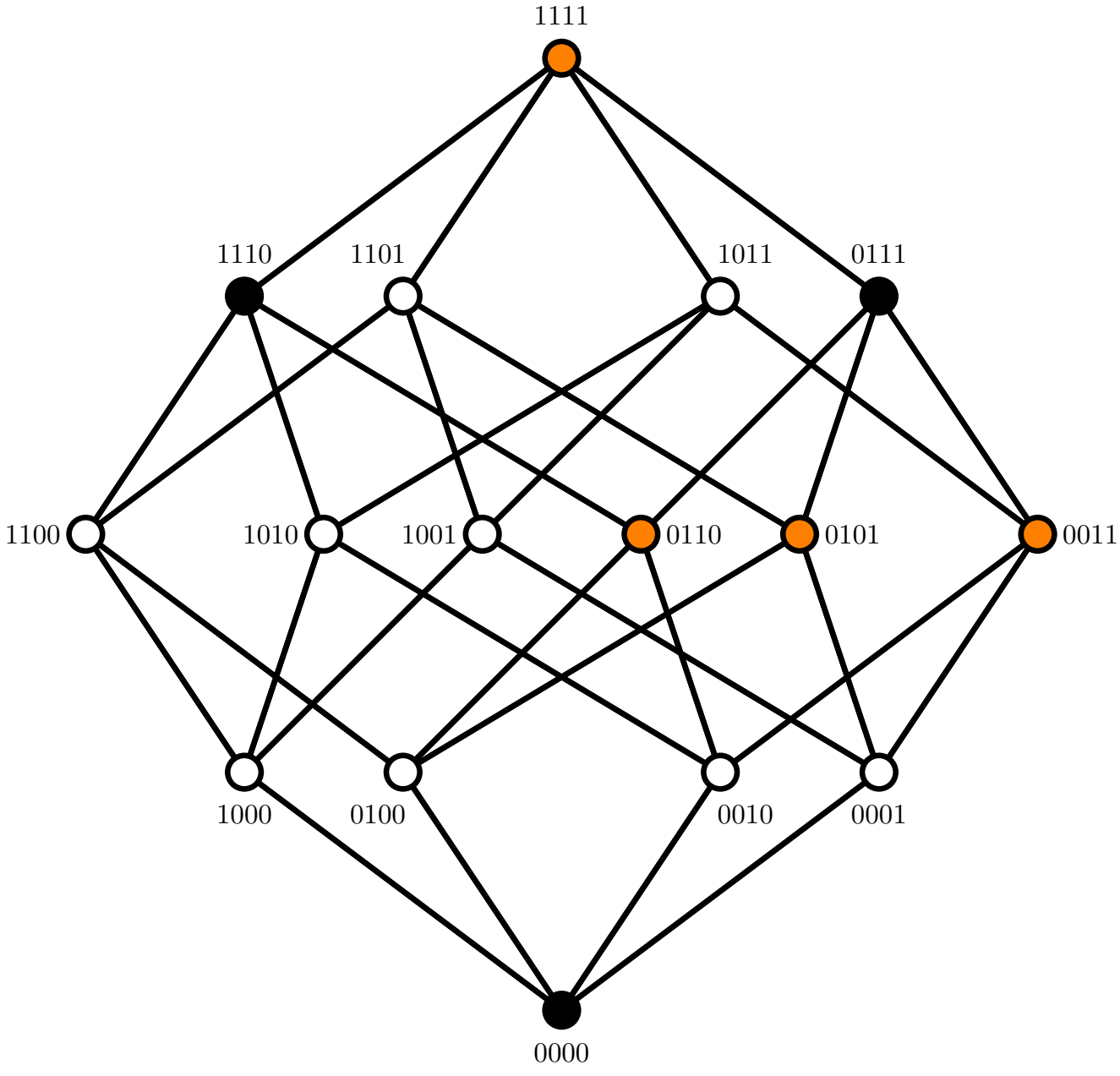
$C = \{0000, 1110, 0111\} \subseteq \{0,1\}^4$



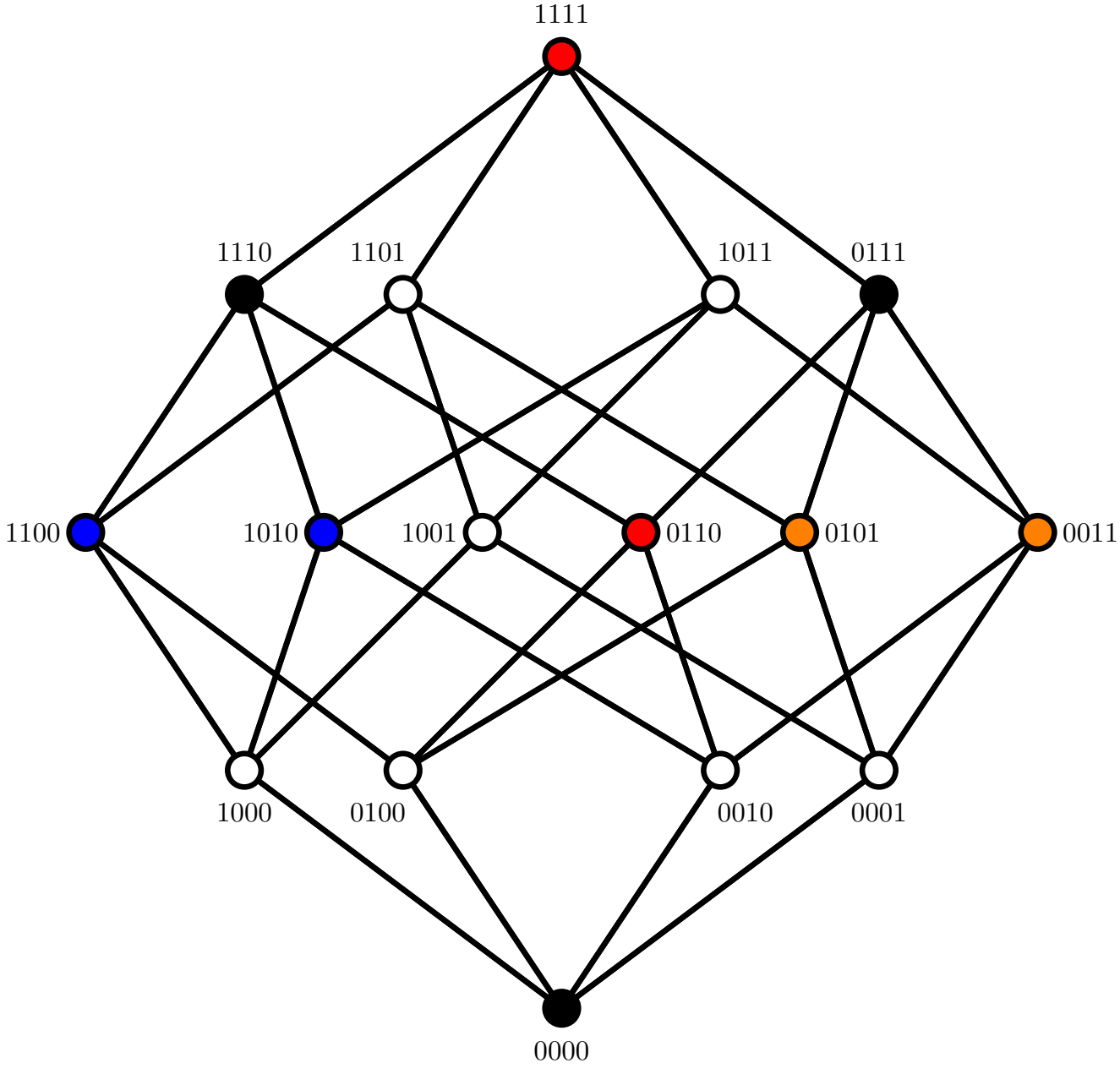
$B_1(1110) \subseteq \{0,1\}^4$



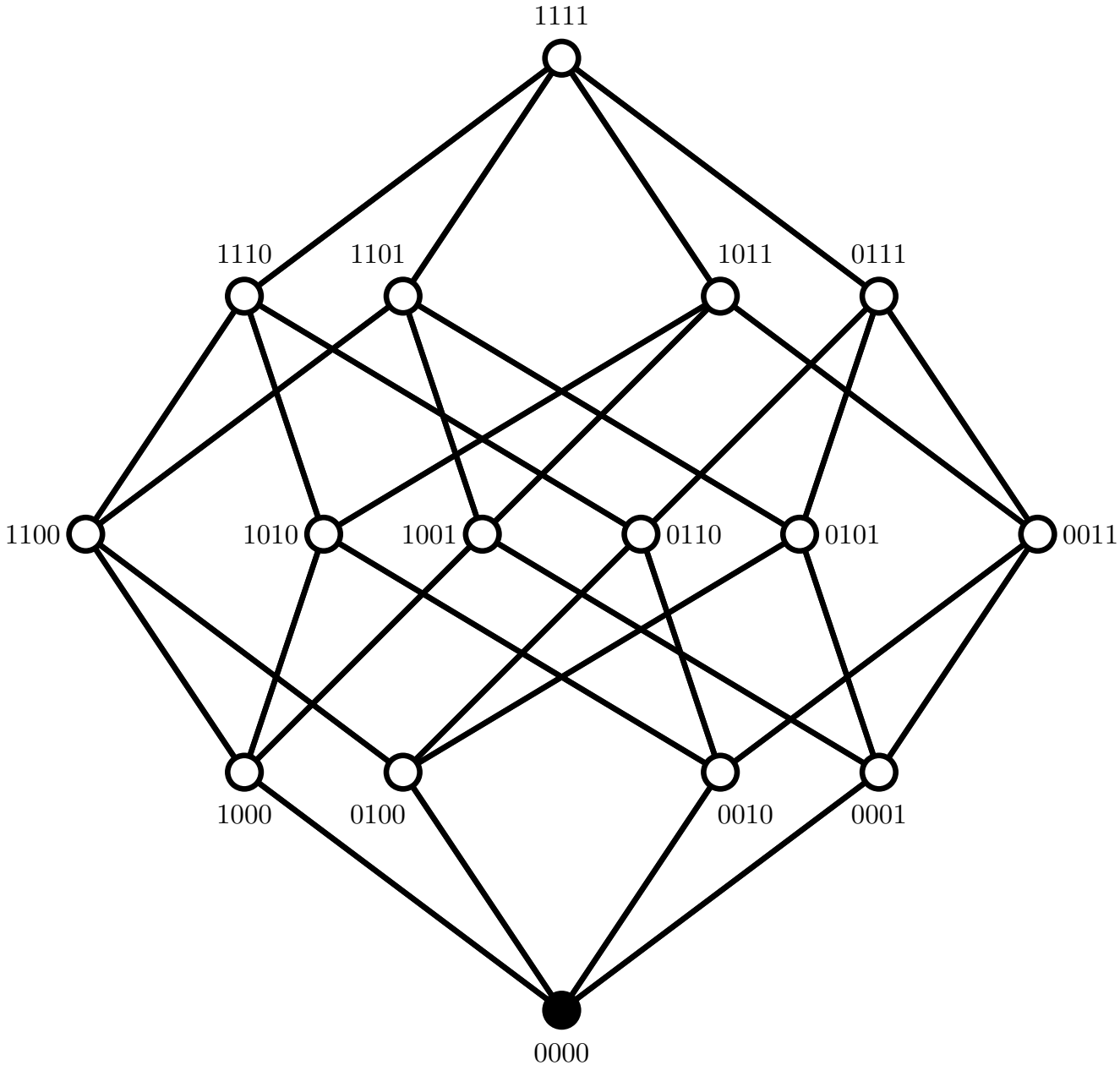
$B_1(0111) \subseteq \{0,1\}^4$



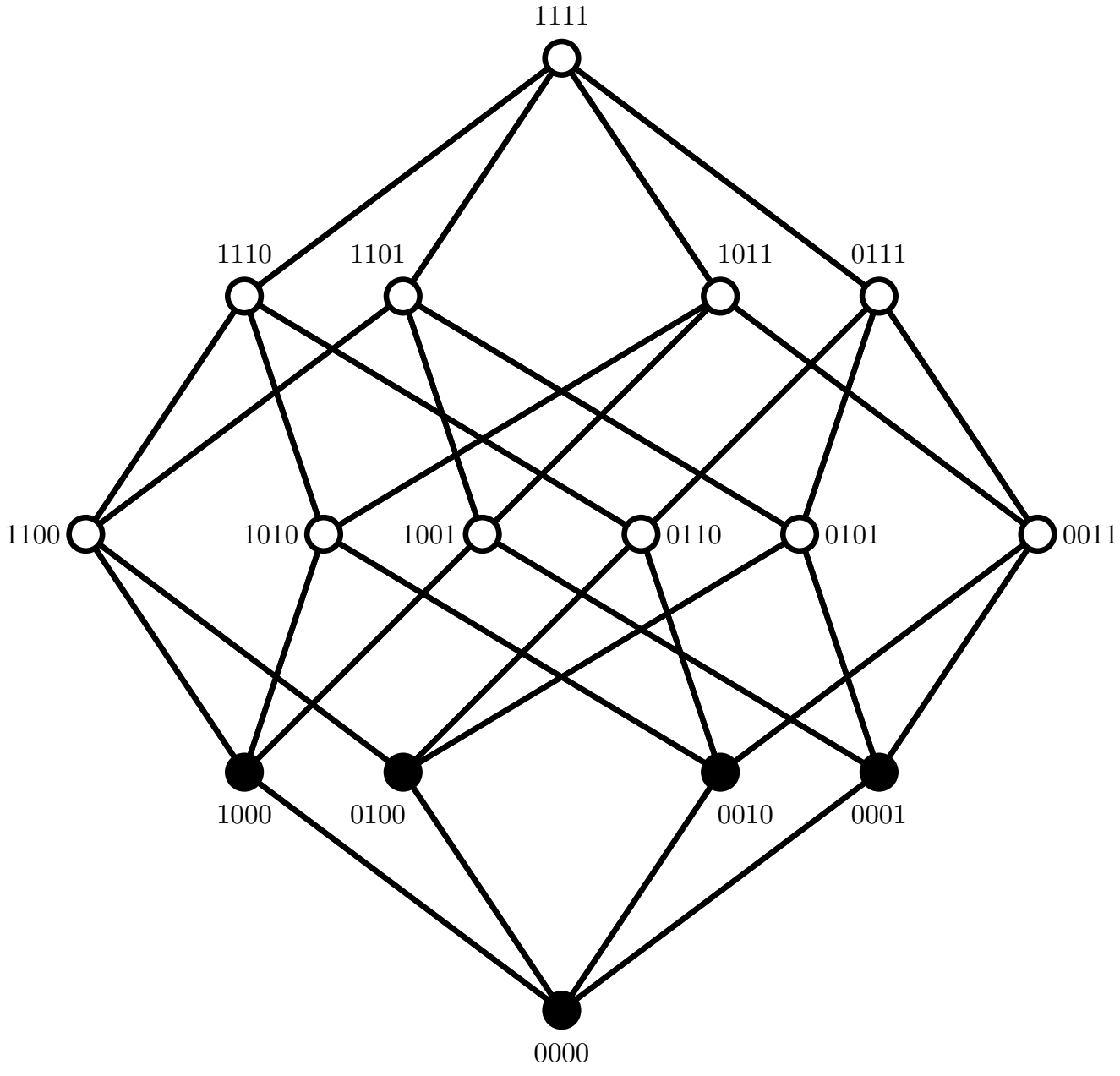
$B_1(1110) \cup B_1(0111) \subseteq \{0,1\}^4$



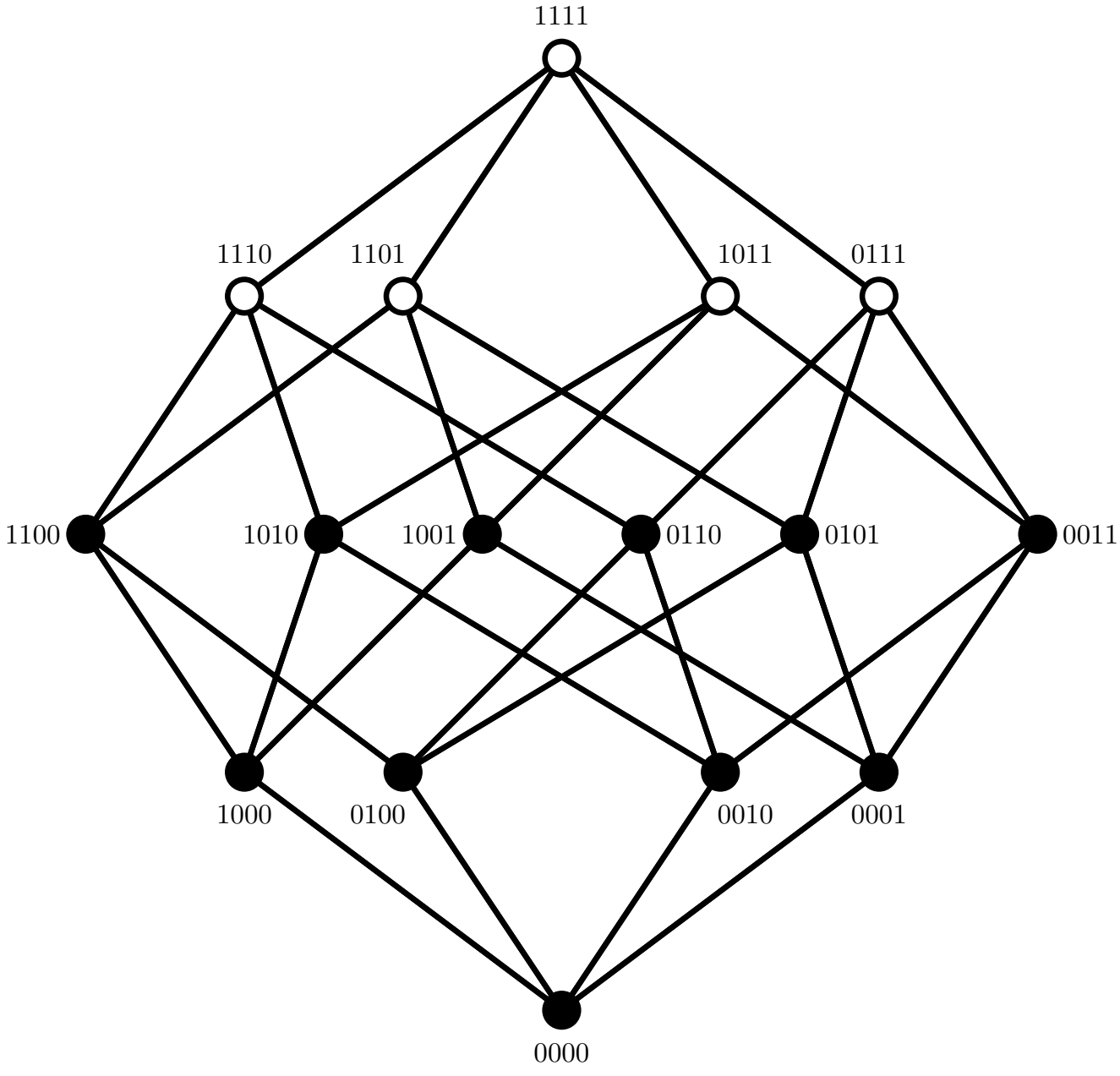
$B_0(0000)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

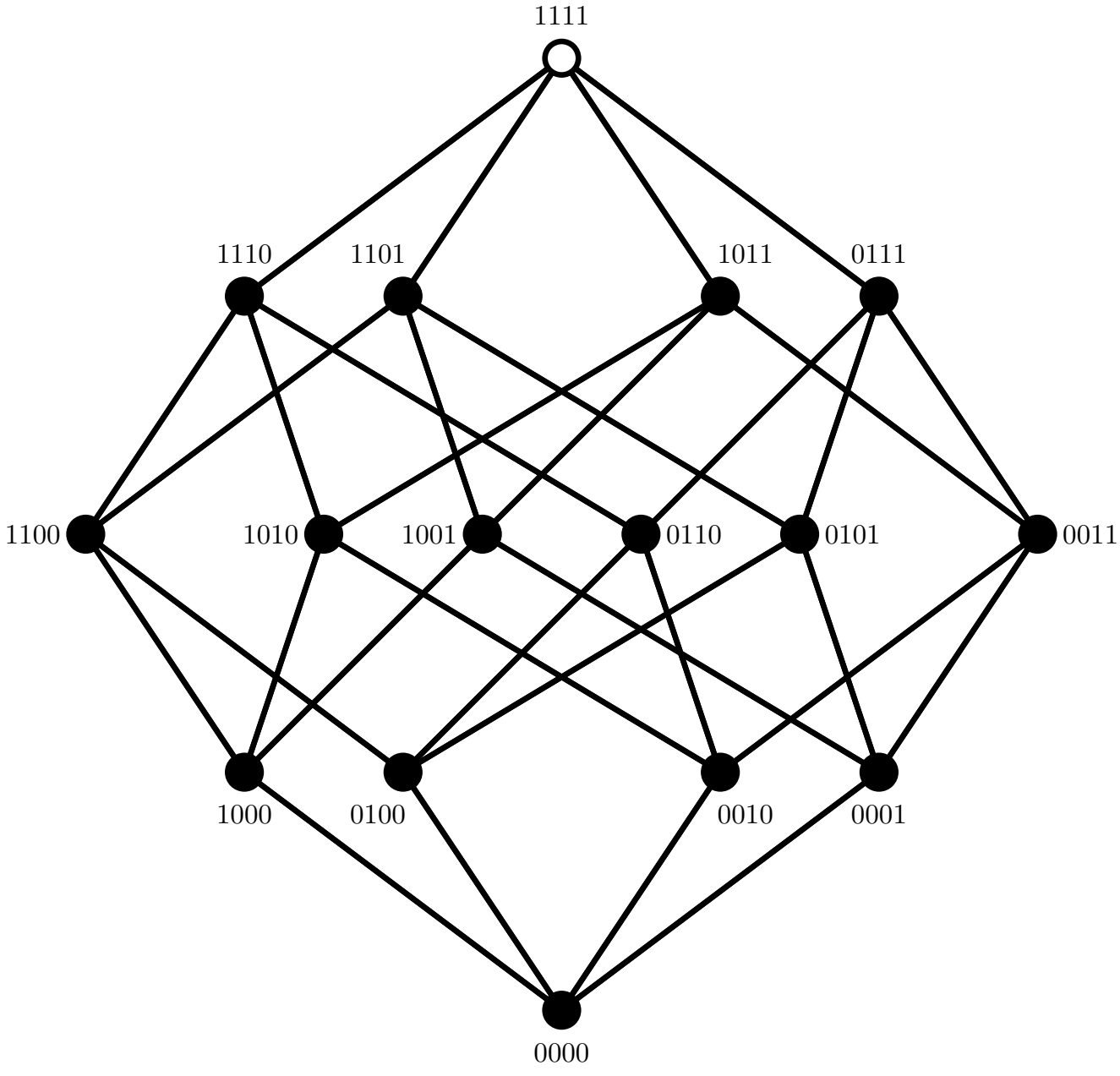
$B_1(0000)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

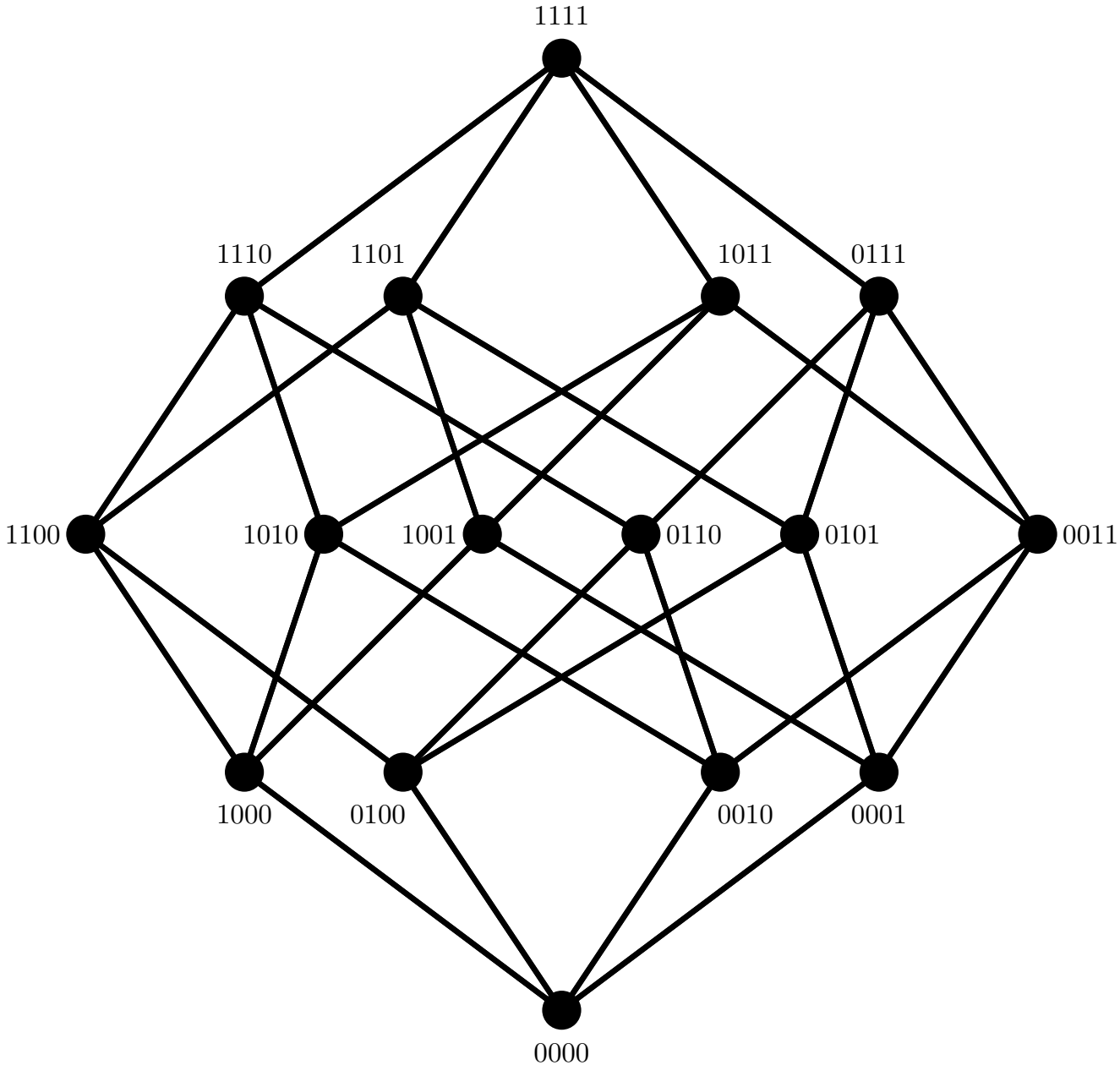
$B_2(0000)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

$B_3(0000)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

$B_4(0000)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

Sizes of Hamming Balls

Let $\mathbf{0}$ denote the all-zeros word $0 \ldots 0$. We just saw the Hamming balls with centre $\mathbf{0} = 0000$ in $\{0,1\}^4$.

r	0	1	2	3	4
$\left\lvert B_r(\mathbf{0})\right\rvert$	1	5	11	15	16
$\left\lvert B_r(\mathbf{0})\right\rvert - \left\lvert B_{r-1}(\mathbf{0})\right\rvert$	1	4	6	4	1

Sizes of Hamming Balls

Let $\mathbf{0}$ denote the all-zeros word $0 \ldots 0$. We just saw the Hamming balls with centre $\mathbf{0} = 0000$ in $\{0,1\}^4$.

r	0	1	2	3	4
$\lvert B_r(\mathbf{0})\rvert$	1	5	11	15	16
$\lvert B_r(\mathbf{0})\rvert - \lvert B_{r-1}(\mathbf{0})\rvert$	1	4	6	4	1

Lemma 8.9

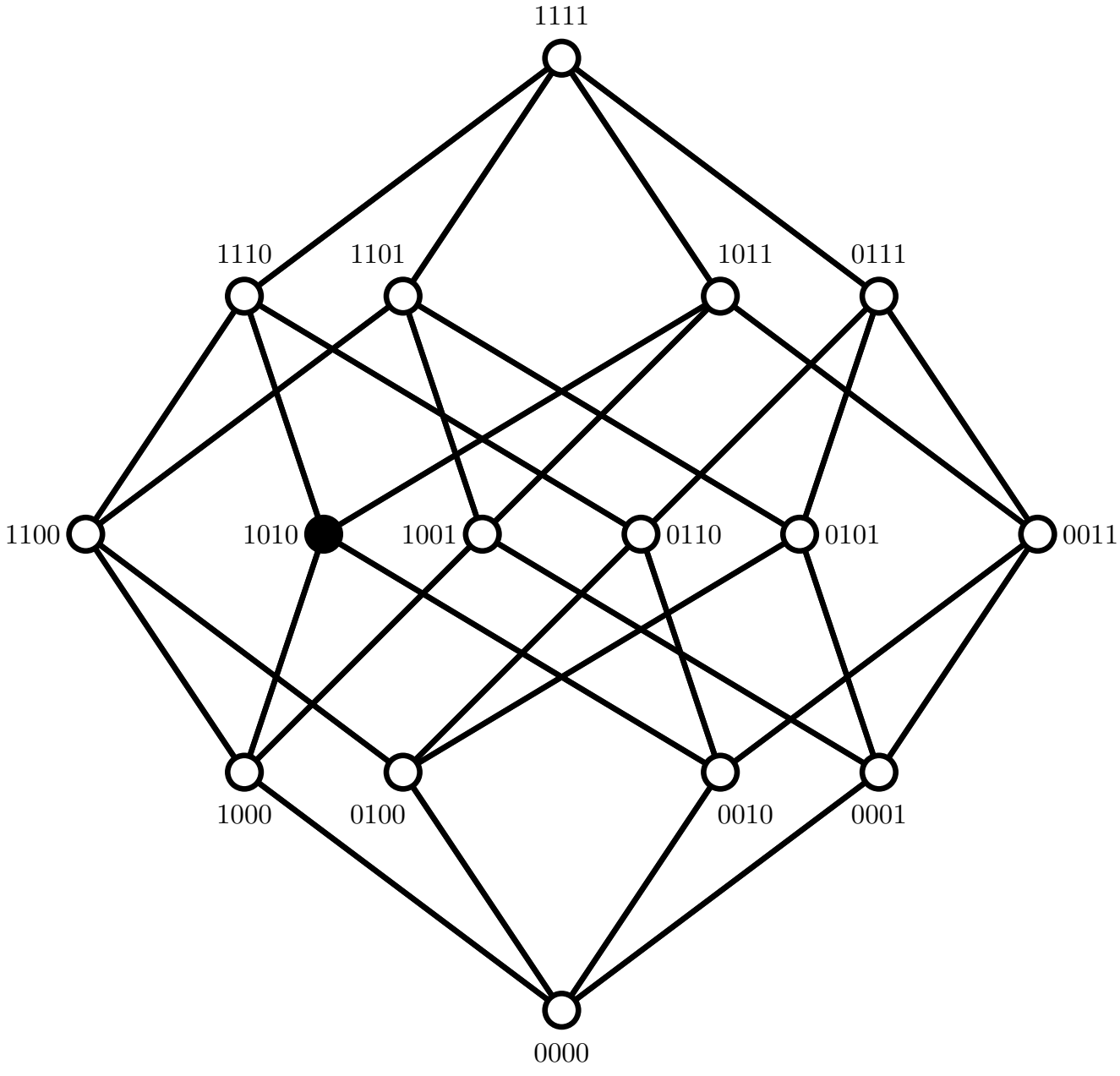
Let $n \in \mathbb{N}$.

(a) $\left|\{v \in \{0,1\}^n : d(u,v) = s\}\right| = \binom{n}{s}$.

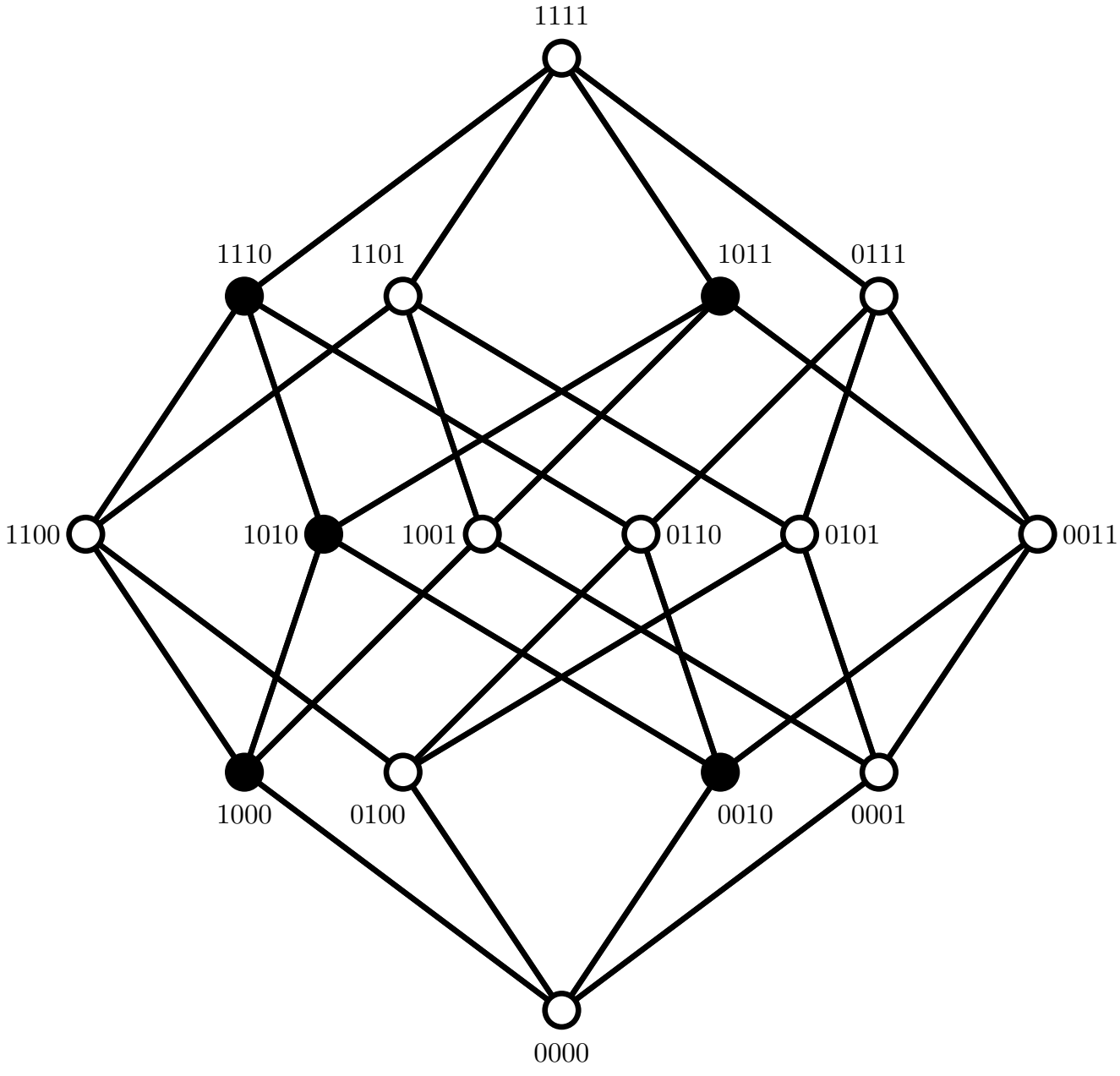
(b) $|B_r(\mathbf{0})| = \sum_{s=0}^{r} \binom{n}{s}$;

The size of a Hamming ball does not depend on its centre.

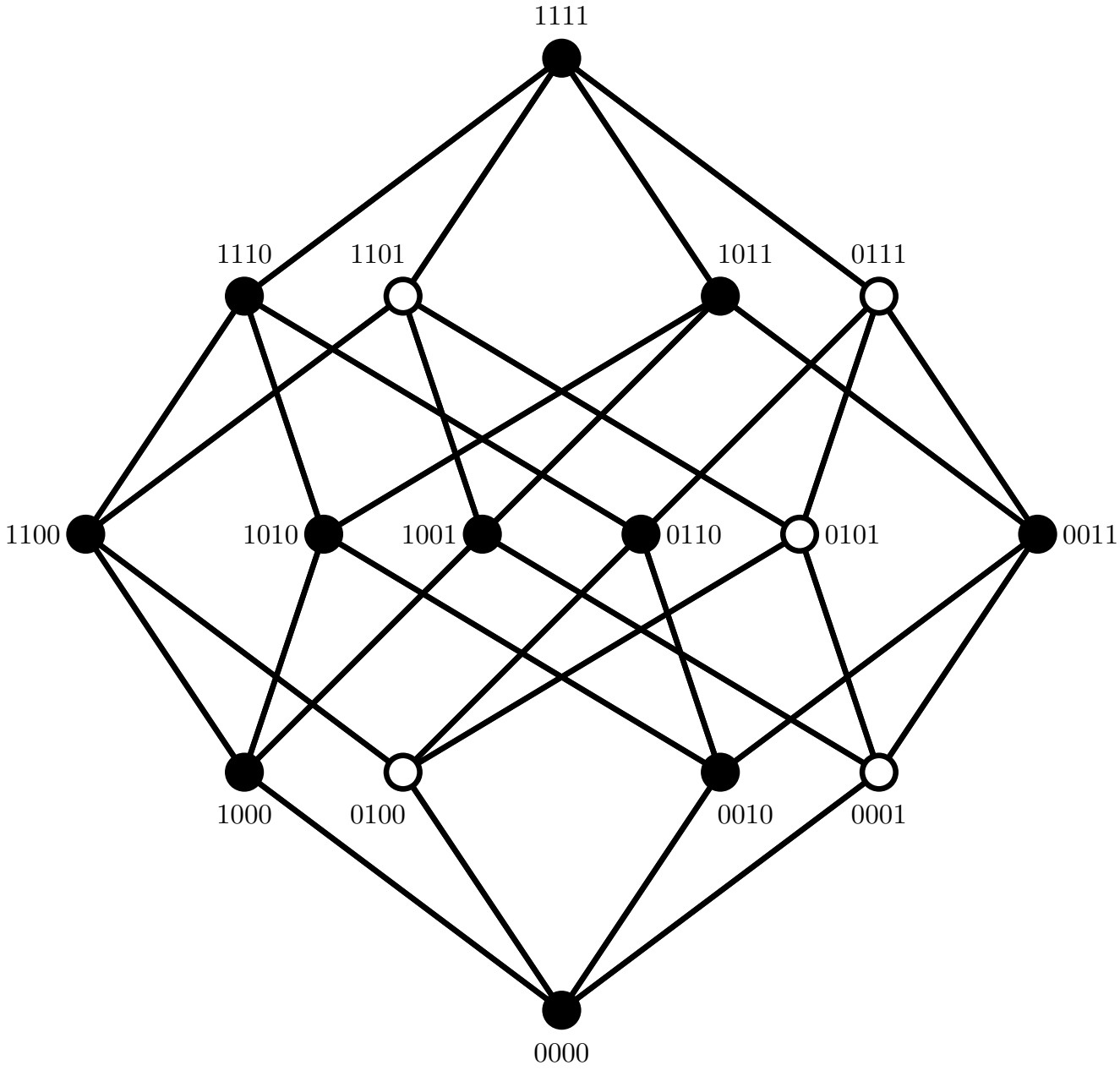
$B_0(1010)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

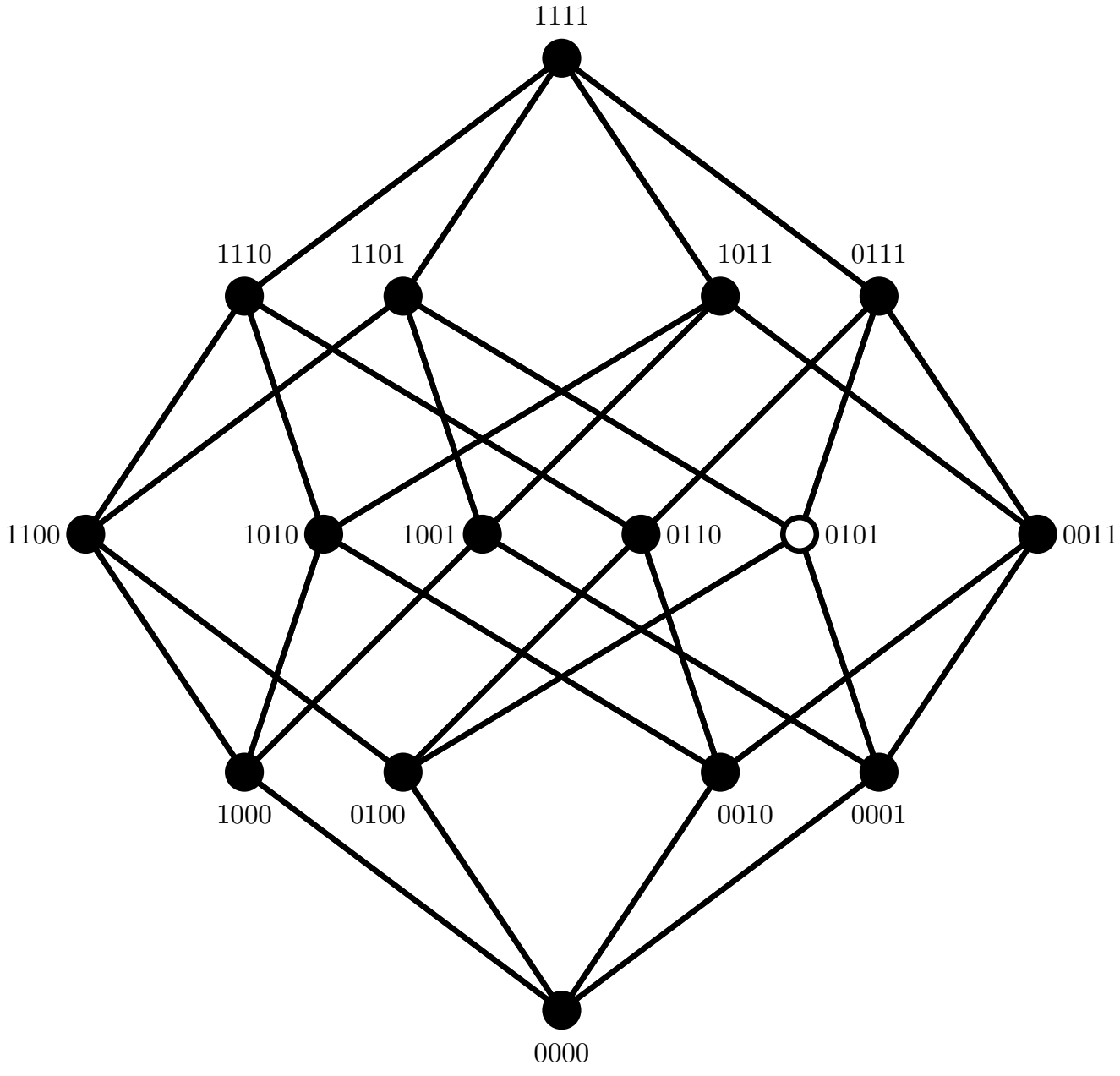
$B_1(1010)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

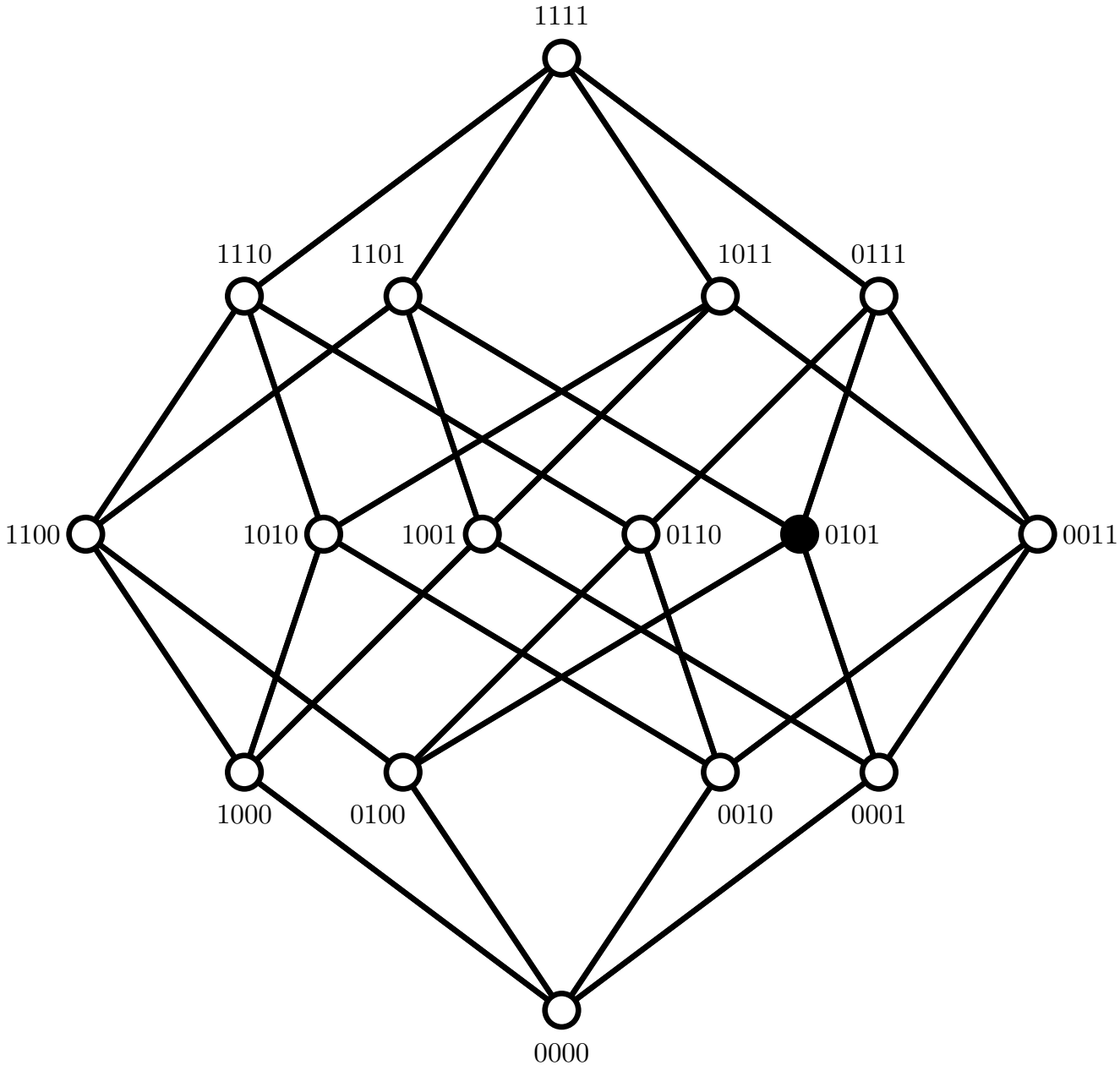
$B_2(1010)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

$B_3(1010)$


1111
1110
1101
1011
0111
1100
1010
1001
0110
0101
0011
1000
0100
0010
0001
0000

$$\{0,1\}^4 \setminus B_3(1010) = \{0101\}$$



Feedback for Problem Sheet 4

By definition, p is a probability measure on $\{1, \ldots, s\}$ if $p_i \geq 0$ for each i and $\sum_{i=1}^{s} p_i = 1$. This is all you needed to check in **(4)(a)**.

(2) Alice must guess Bob's secret number X by asking yes/no questions. She knows that X is distributed on $\{0, 1, \ldots, 2^r\}$ according to the probability measure $(\frac{1}{2}, \frac{1}{2^{r+1}}, \ldots, \frac{1}{2^{r+1}})$. Thus $\mathbb{P}[X = 0] = \frac{1}{2}$ and $\mathbb{P}[X = x] = \frac{1}{2^{r+1}}$ if $x \in \{1, \ldots, 2^r\}$.

(a) Find $H(X)$.

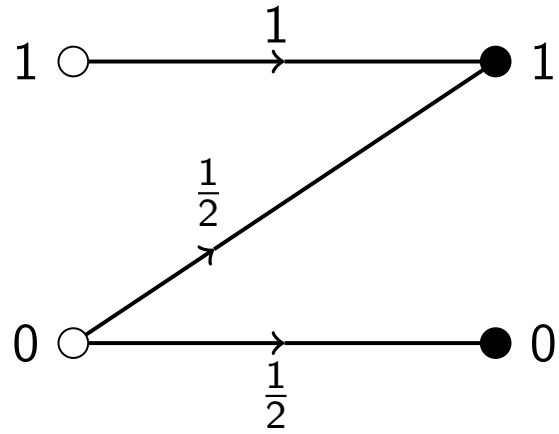
(b) Find, with proof, an optimal prefix-free code for this measure. [*Hint:* you could give a Huffman code, or use Corollary 3.10, or use Theorem 4.6(ii).]

(c) What is the corresponding questioning strategy for Alice?

(d) Let A be the answer to Alice's first question. Find $H(X|A = \text{'yes'})$, $H(X|A = \text{'no'})$ and $H(X|A)$.

(e) Comment on your answers in (d). Is it a surprise to you that the conditional entropy, given a particular answer by Bob, may be higher than $H(X)$?

(3) In the binary channel shown below, when 0 is sent, it flips to 1 with probability $\frac{1}{2}$, and when 1 is sent, 1 is always received.



Suppose that $\mathbb{P}[X = 0] = q$ and $\mathbb{P}[X = 1] = 1 - q$.

(a) Write down the matrix of channel probabilities, as in Example 7.2.

(b) Show that $\mathbb{P}[Y = 0] = \frac{1}{2}q$ and $\mathbb{P}[Y = 1] = 1 - \frac{1}{2}q$. Hence write down a formula for $H(Y)$.

(c) (i) Find $\mathbb{P}[Y = 0|X = 0]$, $\mathbb{P}[Y = 1|X = 0]$ and hence find $H(Y|X = 0)$.

(ii) Find $\mathbb{P}[Y = 0|X = 1], \mathbb{P}[Y = 1|X = 1]$ and hence find $H(Y|X = 1)$.

(iii) Find $H(Y|X)$ in terms of q.

§9 Shannon's Noisy Coding Theorem for the BSC

To prove Shannon's Noisy Coding Theorem for the Binary Symmetric Channel we need some good bounds on the sizes of Hamming balls.

See the optional question on Sheet 7 for some motivation for why the entropy function now appears.

Proposition 9.1

Let $n \in \mathbb{N}$ and let $0 \leq r \leq n/2$. Let $h = H(\frac{r}{n}, 1 - \frac{r}{n})$. Then

$$\frac{1}{n+1} 2^{hn} \leq \binom{n}{r} \leq |B_r(\mathbf{0})| \leq 2^{hn}.$$

Reminder of Linearity of Expectation

Exercise 9.2

(a) Let X, Y be independent rolls of a fair die. Let $Z = X$. Find $\mathbb{E}[X]$, $\mathbb{E}[X + Y]$, $\mathbb{E}[X + Z]$, $\mathbb{E}[X + Y + Z]$. [*Hint:* the hard way to compute $\mathbb{E}[X + Y]$ is to use its probability distribution on $\{2, \ldots, 12\}$, namely $(\frac{1}{36}, \frac{2}{36}, \ldots, \frac{6}{36}, \ldots, \frac{2}{36}, \frac{1}{36})$. The easy way is to use linearity of expectation.]

(b) Let F be the flip of a coin biased to lands heads with probability p and let

$$X = \begin{cases} 1 & \text{if } F = \text{heads} \\ 0 & \text{if } F = \text{tails.} \end{cases}$$

Then

$$\mathbb{E}[X] = 1 \times \mathbb{P}[F = \text{heads}] + 0 \times \mathbb{P}[F = \text{tails}] = p.$$

Thus the expectation of an 'indicator' random variable such as X is the probability of the event defining it. We use this to find $\mathbb{E}[g_i(v)]$ in (2) in the proof below.

Exercise 9.2 [continued]

(c) Suppose that 4 boys and 8 girls sit in a circle, choosing seats at random. On average, how many girls have a boy to their right? *Outline solution.* Number chairs from 0 to 11. Define

$$X_i = \begin{cases} 1 & \text{if chair } i \text{ has a girl and chair } i+1 \pmod{12} \text{ as a boy} \\ 0 & \text{otherwise.} \end{cases}.$$

Show, using the idea in (b) that $\mathbb{E}[X_i] = \frac{8}{12} \times \frac{4}{11}$ and hence that the expected number of GB pairs is $12 \times \frac{8}{12} \times \frac{4}{11} = \frac{32}{11}$.

- how many GG pairs are there?
- how many BG pairs are there?
- how many BB pairs are there?

Exercise 9.2 [continued]

(c) Suppose that 4 boys and 8 girls sit in a circle, choosing seats at random. On average, how many girls have a boy to their right? *Outline solution.* Number chairs from 0 to 11. Define

$$X_i = \begin{cases} 1 & \text{if chair } i \text{ has a girl and chair } i+1 \pmod{12} \text{ as a boy} \\ 0 & \text{otherwise.} \end{cases}.$$

Show, using the idea in (b) that $\mathbb{E}[X_i] = \frac{8}{12} \times \frac{4}{11}$ and hence that the expected number of GB pairs is $12 \times \frac{8}{12} \times \frac{4}{11} = \frac{32}{11}$.

- how many GG pairs are there? A: $\frac{56}{11}$
- how many BG pairs are there?
- how many BB pairs are there?

Exercise 9.2 [continued]

(c) Suppose that 4 boys and 8 girls sit in a circle, choosing seats at random. On average, how many girls have a boy to their right? *Outline solution.* Number chairs from 0 to 11. Define

$$X_i = \begin{cases} 1 & \text{if chair } i \text{ has a girl and chair } i+1 \pmod{12} \text{ as a boy} \\ 0 & \text{otherwise.} \end{cases}.$$

Show, using the idea in (b) that $\mathbb{E}[X_i] = \frac{8}{12} \times \frac{4}{11}$ and hence that the expected number of GB pairs is $12 \times \frac{8}{12} \times \frac{4}{11} = \frac{32}{11}$.

- how many GG pairs are there? A: $\frac{56}{11}$
- how many BG pairs are there? A: $\frac{32}{11}$
- how many BB pairs are there?

Exercise 9.2 [continued]

(c) Suppose that 4 boys and 8 girls sit in a circle, choosing seats at random. On average, how many girls have a boy to their right? *Outline solution.* Number chairs from 0 to 11. Define

$$X_i = \begin{cases} 1 & \text{if chair } i \text{ has a girl and chair } i+1 \pmod{12} \text{ as a boy} \\ 0 & \text{otherwise.} \end{cases}.$$

Show, using the idea in (b) that $\mathbb{E}[X_i] = \frac{8}{12} \times \frac{4}{11}$ and hence that the expected number of GB pairs is $12 \times \frac{8}{12} \times \frac{4}{11} = \frac{32}{11}$.

- how many GG pairs are there? A: $\frac{56}{11}$
- how many BG pairs are there? A: $\frac{32}{11}$
- how many BB pairs are there? A: $\frac{12}{11}$

The Toy $\mathrm{BSC}(p, n)$

Definition 9.3

Given $0 < p < 1/2$ and $n \in \mathbb{N}$ such that $pn \in \mathbb{N}$, the Toy $\mathrm{BSC}(p, n)$ is the channel with input and output alphabets $\{0,1\}^n$ such that when $u \in \{0,1\}^n$ is sent, exactly pn of the positions of u flip.

- This is a good approximation to what happens when a word of length n is sent using the $\mathrm{BSC}(p)$.
- More precisely, given $\epsilon > 0$, by Chebychev's Inequality (or the Central Limit Theorem), the chance that more than $(p+\epsilon)n$ errors, or fewer than $(p-\epsilon)n$ positions are flipped tends to 0 as $n \to \infty$.

The following lemma is proved on Question 5 of Problem Sheet 6.

Lemma 9.4

Let c_n be the capacity of the Toy $\mathrm{BSC}(p, n)$. We have

$$\frac{c_n}{n} \to 1 - H(p, 1-p) \quad \text{as } n \to \infty$$

Shannon's Noisy Coding Theorem(a) for Toy $\mathrm{BSC}(p, n)$

Since n is part of the specification of the channel, the statement changes slightly from Theorem 7.14(a).

Proposition 9.5

Let $h = H(p, 1-p)$. Let $r < 1 - h$. Let $\epsilon > 0$ be given. Provided n is sufficiently large, there exists a binary code C of size $\geq 2^{rn}$ such that when C is used to communicate on the Toy $\mathrm{BSC}(p, n)$ using nearest neighbour decoding, the error probability is $< \epsilon$.

Shannon's great insight was that a code chosen at random is likely to work. For technical reasons it is necessary to choose at least twice as many codewords as are eventually required.

Set $M = 2\lceil 2^{rn} \rceil$ [**Correction: not** $\lceil 2^{rn+1} \rceil$] and let $U(1), \ldots, U(M)$ be codewords, chosen independently and uniformly at random from $\{0, 1\}^n$.

Shannon's Noisy Coding Theorem(a) for Toy $\mathrm{BSC}(p, n)$

Since n is part of the specification of the channel, the statement changes slightly from Theorem 7.14(a).

Proposition 9.5

Let $h = H(p, 1-p)$. Let $r < 1 - h$. Let $\epsilon > 0$ be given. Provided n is sufficiently large, there exists a binary code C of size $\geq 2^{rn}$ such that when C is used to communicate on the Toy $\mathrm{BSC}(p, n)$ using nearest neighbour decoding, the error probability is $< \epsilon$.

Set $M = 2\lceil 2^{rn} \rceil$ [**Correction: not** $\lceil 2^{rn+1} \rceil$] and let $U(1), \ldots, U(M)$ be codewords, chosen independently and uniformly at random from $\{0,1\}^n$.

- We write $\mathbb{P}_{\mathrm{ch}}$ for probabilities in the channel, for example $\mathbb{P}_{\mathrm{ch}}[Y = y | X = x]$ is the probability that Y is received given that X is sent.
- We write $\mathbb{P}_{\mathrm{code}}$ for probabilities depending on the random choice of code: for instance, $\mathbb{P}[U(1) = u] = \frac{1}{2^n}$ for all $u \in \{0,1\}^n$.

Shannon's Noisy Coding Theorem(a) for Toy $\mathrm{BSC}(p, n)$

Since n is part of the specification of the channel, the statement changes slightly from Theorem 7.14(a).

Proposition 9.5

Let $h = H(p, 1-p)$. Let $r < 1 - h$. Let $\epsilon > 0$ be given. Provided n is sufficiently large, there exists a binary code C of size $\geq 2^{rn}$ such that when C is used to communicate on the Toy $\mathrm{BSC}(p, n)$ using nearest neighbour decoding, the error probability is $< \epsilon$.

Set $M = 2\lceil 2^{rn} \rceil$ [**Correction: not** $\lceil 2^{rn+1} \rceil$] and let $U(1), \ldots, U(M)$ be codewords, chosen independently and uniformly at random from $\{0, 1\}^n$.

Lemma 9.6

If $X : \Omega \to \mathbb{R}$ is a random variable then there is an outcome $\omega \in \Omega$ such that $X(\omega) \leq \mathbb{E}[X]$. $\square$

Recap of Step 1 and Step 2

- We defined P_i to be the probability that when $X = U(i) \in \{0,1\}^n$ is sent through the Toy BSC(p, n), the received word $Y \in \{0,1\}^n$ is not decoded as $U(i)$, using nearest neighbour decoding.
- For $v \in \{0,1\}^n$ we defined

$$g_i(v) = \left|\{j : j \neq i, U(j) \in B_{pn}(v)\right|.$$

  - If no codeword $U(j)$ is in $B_{pn}(v)$ then, under nearest neighbour decoding, X is always decoded correctly as $U(i)$.
  - Otherwise we assume that nearest neighbour decoding *never* works.
  - Therefore

$$P_i \leq \sum_{v \in \{0,1\}^n} \mathbb{P}[Y = v | X = U(i)] g_i(v).$$

- In Step (2) we compute the expectation of P_i *in the probability space of the random code*.
  - We use that if S and T are independent random variables then $\mathbb{E}[ST] = \mathbb{E}[S]\mathbb{E}[T]$.
  - Correction: $\eta = 1 - r - h$, not $1 - r + h$.

Notes on Proof

This proof will take a lot of thinking about.

- Question 1 on Problem Sheet 7 asks you to fill in the details in the argument at the start of Step 2. This should clarify the role played by the two different probability spaces.
- Question 2 then asks you to adapt the proof to the Toy Binary Erasure Channel, in which exactly pn bits are erased.

§10 Converse in Shannon's Noisy Coding Theorem

The proof depends on two inequalities, both of interests in their own right.

- The *Data-Processing Inequality* states that if X, Y are random variables, taking values in sets $\mathcal{X}$ and $\mathcal{Y}$, and $d : \mathcal{Y} \to \mathcal{Z}$ is a function, then $I(X;Y) \geq I\big(X; d(Y)\big)$. In words: processing Y by the function d cannot increase the amount of information Y has about X.
- *Fano's Inequality* states that if X and Y are random variables taking values in a set of size M, and $\mathbb{P}[X = Y] \geq 1 - \epsilon$ then $H(X|Y) \leq H(\epsilon, 1-\epsilon) + \epsilon \log_2(M-1)$.

Data-Processing Inequality

Lemma 10.1

Let X, Y and Z be random variables. Then

$$H\big(X|(Y,Z)\big) \leq H(X|Z).$$

[Typo in printed notes: $H\big(X;(Y,Z)\big)$ should be $H\big(X|(Y,Z)\big)$ and similarly $H(X;Z)$ should be $H(|Z)$.]

Lemma 10.2 (Data-Processing Inequality)

If X, Y are random variables, taking values in sets $\mathcal{X}$ and $\mathcal{Y}$ respectively, and $d : \mathcal{Y} \to \mathcal{Z}$ is a function, then

$$I(X;Y) \geq I\big(X;d(Y)\big).$$

Motivation for Fano's Inequality

Example 10.3

Alice and Bob may go to the cinema, theatre or stay at home, each with equal probability. With probability $1 - p$ where we imagine p is small, their decisions X and Z agree. In the 'error' case they differ. A nice way to find the joint entropy (X, Z) is to condition on the event that $X \neq Z$. For this we introduce the 'indicator' random variable, as in Example 9.2(b),

$$F = \begin{cases} 1 & \text{if } X \neq Z \\ 0 & \text{if } X = Z. \end{cases}$$

Example 10.3 [continued]

Since F is determined by (X, Z) we have

$$H(X, Z) = H(X, Z, F) = H(X, Z|F) + H(F)$$

using the Chaining Rule (Lemma 7.6) for the second equality. Now $H(F) = H(p, 1-p)$,

$$H(X, Z|F = 0) = H(X, X) = H(X) = \log_2 3$$

and $H(X, Z|F = 1) = \log_2 6 = 1 + \log_2 3$ since there are 6 equally likely pairs of destinations when $X \neq Z$. Therefore $H(X, Y|F) = (1-p)\log_2 3 + p(1 + \log_2 3) = 1 - p$ and

$$H(X, Z) = p + \log_2 3 + H(p, 1-p).$$

Fano's Inequality

To prove Fano's Inequality we need that if X is a random variable taking m different values then $H(X) \leq \log_2 m$. By Question 6 on Problem Sheet 3, this follows easily from Gibbs' Inequality.

Lemma 10.4 (Fano's Inequality)

Let X and Z be random variables taking values in a set of size M. Let $\epsilon < \frac{1}{2}$. If $\mathbb{P}[X = Z] \geq 1 - \epsilon$ then

$$H(X|Z) \leq H(\epsilon, 1-\epsilon) + \epsilon \log_2(M-1).$$

Fano's Inequality

To prove Fano's Inequality we need that if X is a random variable taking m different values then $H(X) \leq \log_2 m$. By Question 6 on Problem Sheet 3, this follows easily from Gibbs' Inequality.

Lemma 10.4 (Fano's Inequality)

Let X and Z be random variables taking values in a set of size M. Let $\epsilon < \frac{1}{2}$. If $\mathbb{P}[X = Z] \geq 1 - \epsilon$ then

$$H(X|Z) \leq H(\epsilon, 1-\epsilon) + \epsilon \log_2(M-1).$$

The final result we need to prove the converse in Shannon's Noisy Coding Theorem is Question 4 on Problem Sheet 7: when a memoryless channel of capacity c is used to send words of n symbols, its capacity is nc.

This should be quite intuitive: since the channel transmits each symbol independently, the amount of information about the input we can (at best) learn from each of the n received symbols is the original capacity c.

Proof of Shannon's Noisy Coding Theorem (b)

The final result we need to prove the converse in Shannon's Noisy Coding Theorem is Question 3 on Problem Sheet 7: when a memoryless channel of capacity c is used to send words of n symbols, its capacity is nc.

Theorem 7.14 (Shannon's Noisy Coding Theorem for Discrete Memoryless Channels)

Fix a discrete memoryless channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$ of capacity c.

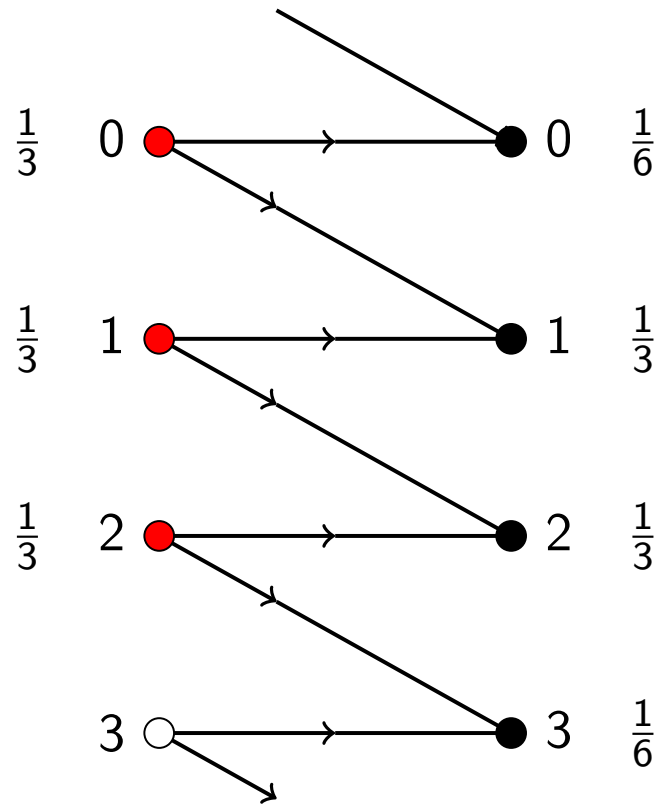
(a) *Let $\epsilon > 0$ be given. For every $r < c$ there exists $n \in \mathbb{N}$ and a code $C \subseteq \mathcal{A}^n$ such that $|C| \geq 2^{rn}$ and the error probability when C is used to send codewords through the channel is less than ϵ.*

(b) *If $r > c$ then, when n is large, it is impossible to find a code as in (a).*

Example 7.15

Take the lazy typist channel on $2t$ symbol and use it send words of length n from $\{0, 1, \ldots, 2t-1\}^n$. (This is the n-extension of the channel, as in Question 4 on Sheet 7.) By Question 3 on Sheet 5, the capacity of the lazy typist channel is $\log_2 t$.

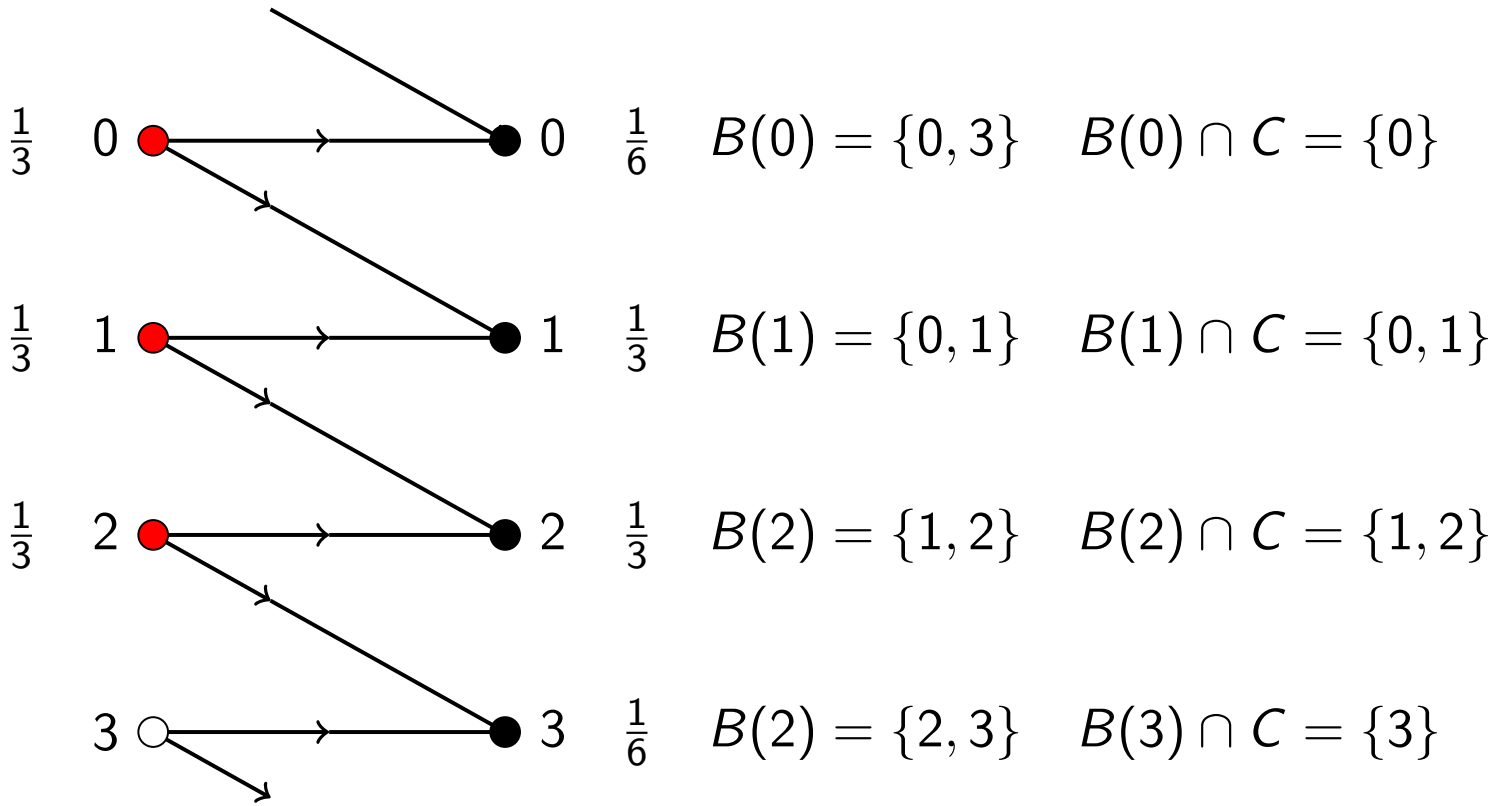
All arrows have probability $\frac{1}{2}$. As in Shannon's Noisy Coding Theorem, all codewords have equal probability.



Example 7.15

Take the lazy typist channel on $2t$ symbol and use it send words of length n from $\{0, 1, \ldots, 2t-1\}^n$. (This is the n-extension of the channel, as in Question 4 on Sheet 7.) By Question 3 on Sheet 5, the capacity of the lazy typist channel is $\log_2 t$.

All arrows have probability $\frac{1}{2}$. As in Shannon's Noisy Coding Theorem, all codewords have equal probability.



Example 7.15 [continued]

To make things more concrete, take $\epsilon = \frac{1}{10}$. Shannon's Noisy Coding Theorem (b) then says that if $r > \log_2 t$, when n is large, it is impossible to find a code $C \subseteq \{0, 1, \ldots, 2t-1\}^n$ such that $|C| \geq 2^{nr}$, and the error probability for each codeword is $< \frac{1}{10}$.

Suppose that C is such a code. For each $v \in \{0, 1, \ldots, 2t-1\}^n$ let

$$B(v) = \left\{u \in \{0, 1, \ldots, 2t-1\}^n : u_i = v_i \text{ or } u_i = v_i - 1 \bmod 2t \text{ for all } i\right\}.$$

This is the set of sent words that may be received as v. Let $M(v) = |B(v) \cap C|$. When v is received, the decoder must choose arbitrarily between the $M(v)$ equally likely codewords in $B(v)$. (Remember that each codeword is equally likely to be sent.) So the probability of decoding incorrectly is $1 - 1/M(v)$. Let P be the average probability of incorrect decoding; by assumption $P < \frac{1}{10}$. We have

$$P = \sum_{v \in \{0,1,\ldots,2s-1\}^n} \mathbb{P}[\text{decode wrongly} | Y = v]\mathbb{P}[Y = v]. \quad (\dagger)$$

By the usual conditioning argument

Part (C): Ergodic sources and the Asymptotic Equipartition Property

§11 Typical words from memoryless sources

Question. What is a typical word from a source?

In the final part of the course we make sense of this question and use the answer to give a new proof of Shannon's Source Coding Theorem and, in outline only, our first proof of the constructive part of Shannon's Noisy Coding Theorem in full generality. We end by considering some practical solutions to the problems of source and channel coding.

Part (C): Ergodic sources and the Asymptotic Equipartition Property

§11 Typical words from memoryless sources

Question. What is a typical word from a source?

In the final part of the course we make sense of this question and use the answer to give a new proof of Shannon's Source Coding Theorem and, in outline only, our first proof of the constructive part of Shannon's Noisy Coding Theorem in full generality. We end by considering some practical solutions to the problems of source and channel coding.

It is hard to give an example of a 'typical word'. It is a bit like asking 'what is a typical pine tree?': the trees that stand out are all, for one reason or another, atypical. The best answer is probably: 'go into a pine wood and choose one at random — then it's probably fairly typical'.

Binomial memoryless source

Exercise 11.1

Let $p < \frac{1}{2}$. A memoryless source emits 0 with probability $1 - p$ and 1 with probability p. Let $n \in \mathbb{N}$.

(i) What is the most common message of length n? Can one reasonably say it is typical?

(ii) How many 1s are there in a typical word of length r?

(iii) What is the probability of each word of length r with the average number of 0s and 1s? (Suppose that $pr \in \mathbb{N}$.)

(iv) How is this related to the entropy of the source?

(v) What does Shannon's Source Coding Theorem have to say about efficiently coding messages from this source?

(vi) In what sense are words with about pr 1s typical?

Weak Law of Large Numbers

A good answer to (vi) is given by the Weak Law of Large Numbers. Despite its name, it is very powerful and useful! We shall prove it using Chebyshev's Inequality: see Question 6(b) on Problem Sheet 1.

Proposition 11.2 (Weak Law of Large Numbers)

Let $X_1, \ldots, X_r$ be independent real-valued random variables each with expectation μ and variance σ^2. Then

$$\mathbb{P}\Big[\mu - \epsilon < \frac{X_1 + \cdots + X_r}{r} < \mu + \epsilon\Big] \to 1 \quad \text{as } r \to \infty.$$

Logs of Probabilities

Exercise 11.1(iv) suggests that the random variable $\log_2 \mathbb{P}[S_1 \dots S_n]$ is of interest, where $S_1 \dots S_n$ is a random word of length n emitted by a source. Note that a probability appears 'inside' the random variable: this is not so unusual in information theory, but rarely seen in other fields using probability.

Example 11.3

Take the source from Example 11.1.

(a) The random variable $\log_2 \mathbb{P}[S_1]$ takes value $\log_2 p$ with probability p and $\log_2(1-p)$ with probability $1-p$. (Since $p < \frac{1}{2}$, these values are distinct.)

(b) The random variable $\log_2 \mathbb{P}[S_1 S_2 S_3]$ takes distinct values

$$\log_2(1-p)^3,\ \log_2 p(1-p)^2,\ \log_2(1-p)p^2,\ \log_2 p^3$$

with probabilities $(1-p)^3, 3p(1-p)^2, 3p(1-p)p^2, p^3$, respectively. *Exercise:* What is its expectation?

Logs of Probabilities: General (Memoryless) Case

Exercise 11.4

Let $S_1, S_2, \ldots$ be a memoryless source and let

$$h = H(S_1) = H(S_2) = \ldots.$$

(a) Express $\mathbb{E}\big[\log \mathbb{P}[S_1]\big]$ in terms of h.

(b) What is $\mathbb{E}\big[\log \mathbb{P}[S_1 \ldots S_n]\big]$ in terms of h?

Logs of Probabilities: General (Memoryless) Case

Exercise 11.4

Let $S_1, S_2, \ldots$ be a memoryless source and let

$$h = H(S_1) = H(S_2) = \ldots.$$

(a) Express $\mathbb{E}\big[\log \mathbb{P}[S_1]\big]$ in terms of h.

(b) What is $\mathbb{E}\big[\log \mathbb{P}[S_1 \ldots S_n]\big]$ in terms of h?

Lemma 11.5

Let $S_1, S_2, \ldots$ be the output of a memoryless source producing symbols in an alphabet $\mathcal{S}$. Let $h = H(S_1)$ be the per-symbol entropy. Given $\epsilon > 0$, there exists $r \in \mathbb{N}$ and a subset $\mathcal{T}^{(r)}$ of $\mathcal{A}^r$ such that

(i) $\mathbb{P}[S_1 \ldots S_r \in \mathcal{T}^{(r)}] > 1 - \epsilon$;

(ii) $2^{-r(h+\epsilon)} \leq \mathbb{P}[S_1 \ldots S_r = s_1 \ldots s_r] \leq 2^{-r(h-\epsilon)}$ *for all words* $s_1 \ldots s_r \in \mathcal{T}^{(r)}$.

§12 The Asymptotic Equipartition Property

Definition 12.1

Let $S_1, S_2, \ldots$ be the symbols in an alphabet $\mathcal{S}$ output by a source. We say the source satisfies the *Asymptotic Equipartition Property (AEP)* if there exists $h \geq 0$ such that for all $\epsilon > 0$ there exists $r \in \mathbb{N}$ and a subset $\mathcal{T}^{(r)}$ of $\mathcal{A}^r$ such that

(i) $\mathbb{P}[S_1 \ldots S_r \in \mathcal{T}^{(r)}] > 1 - \epsilon$;

(ii) $2^{-r(h+\epsilon)} \leq \mathbb{P}[S_1 \ldots S_r = s_1 \ldots s_r] \leq 2^{-r(h-\epsilon)}$ for all $s_1 \ldots s_r \in \mathcal{T}^{(r)}$.

By Lemma 11.5, a memoryless source $S_1, S_2, \ldots$ satisfies the AEP, with $h = H(S_1)$. In the remainder of this section we prove some corollaries of this result.

As a warm up, we prove a special case of the constructive part of Shannon's Source Coding Theorem. (See Problem Sheet 8 for the general version: this special case should be helpful.)

Shannon's Source Coding Theorem and the AEP

Example 12.2

Let $S_1, S_2, \ldots$ be a memoryless source emitting the bits 0 and 1 each with probability $1-p$ and p. Let

$$h = H(p, 1-p) = -p \log_2 p - (1-p)\log_2(1-p).$$

Then $H(S_1) = H(S_2) = \ldots = h$. Fix $\epsilon > 0$, to be chosen by the end of the proof. By the AEP, there exists a subset $\mathcal{T}$ of $\{0,1\}^n$ such that $\mathbb{P}[S_1 \ldots S_r \in \mathcal{T}] \geq 1-\epsilon$ and

$$2^{-n(h+\epsilon)} \leq \mathbb{P}[S_1 \ldots S_r = s_1 \ldots s_r] \leq 2^{-n(h-\epsilon)}$$

for all $s_1 \ldots s_r \in \mathcal{T}$. By the lower bound above,

$$\sum_{s_1 \ldots s_r \in \mathcal{T}} \mathbb{P}[S_1 \ldots S_r = s_1 \ldots s_r] \geq |\mathcal{T}| 2^{-n(h+\epsilon)}.$$

Therefore $|\mathcal{T}| \leq 2^{r(h+\epsilon)}$. We can encode all the typical words from the source using the binary words of a fixed length $\geq r(h+\epsilon)$.

Example 12.2: Injective Encoder

To turn this idea into a well-defined injective encoder $f^{(r)}$, let $m = 1 + \lceil r(h+\epsilon) \rceil$ and let $u(0), \ldots, u(M-1)$ be a list of all the words in $\mathcal{T}$. Note that $M \leq 2^{m-1}$, and so the binary form of each $j < M$ has at most $m-1$ bits. We define $f^{(r)}$ as follows:

- if $s_1 \ldots s_r \in \mathcal{T}$, with $s_1 \ldots s_r = u(j)$, then

$$f^{(r)}(s_1 \ldots s_r) = 1 j_0 \ldots j_{m-1} \in \{0,1\}^m$$

  where $j_0 \ldots j_{m-1}$ is the m-bit binary form of j.
- if $s_1 \ldots s_r \notin \mathcal{T}$, then

$$f^{(r)}(s_1 \ldots s_r) = 0 s_1 \ldots s_r \in \{0,1\}^{r+1}.$$

Exercise: check that the code

$$C = \{f^{(r)}(s_1 \ldots s_r) : s_1 \ldots s_r \in \{0,1\}^r\}$$

is prefix-free.

Example 12.2: Expected Length for the Code

The length of the codeword $f^{(r)}(s_1 \ldots s_r)$ depends only on whether or not $s_1 \ldots s_r$ is typical. We have

$$\begin{aligned}\overline{f}^{(r)} &= \mathbb{P}[S_1 \ldots S_r \in \mathcal{T}]m + \mathbb{P}[S_1 \ldots S_r \notin \mathcal{T}](r+1) \\ &\leq m + \epsilon(r+1) \\ &\leq 2 + r(h+\epsilon) + \epsilon(r+1)\end{aligned}$$

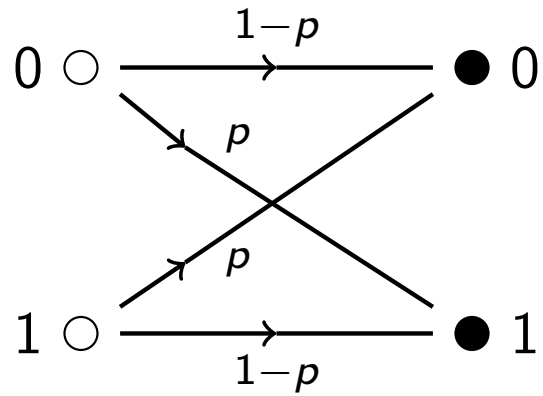
and so

$$\frac{\overline{f}^{(r)}}{r} = h + 2\epsilon + \frac{2+\epsilon}{r}.$$

By choosing r sufficiently large and ϵ sufficiently small, we may make the right-hand side arbitrarily close to h, as required.

Shannon's Noisy Coding Theorem for the BSC(p) and the AEP

We now use the AEP to prove Shannon's Noisy Coding Theorem (Theorem 7.4) in the special case of the Binary Symmetric Channel with error probability p. Recall that the input and output alphabets are $\{0, 1\}$ and that each sent bit flips, independently, with probability p.



Let $h = H(p, 1-p)$. By Example 7.13(a), the capacity of this channel is $1 - H(p, 1-p)$. Let $r < c$ be given and, as in the proof for the toy version of the channel seen in §9, let $M = 2\lceil 2^{rn}$, where n will be chosen by the end of the proof.

Shannon's Noisy Coding Theorem for BSC(p)

As in the proof for the toy version, we choose a code $C = \{U(1), \ldots, U(M)\} \subseteq \{0,1\}^n$ by picking each codeword independently and uniformly at random from $\{0,1\}^n$.

As usual, let $X \in \{0,1\}^n$ denote the sent codeword and $Y \in \{0,1\}^n$ denote the received word. The main idea is to apply the AEP to find the typical set for the random variable (X, Y). For this we need to know its entropy.

Lemma 12.3

Let $X \in \{0,1\}^n$ be distributed uniformly and let $Y \in \{0,1\}^n$ be the received word when X is sent through the BSC(p). The pairs (X_i, Y_i) are independent random variables and

$$H(X_1, Y_1) = \ldots = H(X_n, Y_n) = 1 + h.$$

Application of the AEP

Thus $(X_1, Y_1), (X_2, Y_2), \ldots$ is the sequence of symbols in $\{(0,0), (0,1), (1,0), (1,1)\}$ emitted by a memoryless source of entropy $1 + h$. By the AEP, given $\epsilon > 0$, provided n is sufficiently large, there is a subset $\mathcal{T}$ of $\{0,1\}^{2n}$ such that if X and Y are as in the lemma, then

$$\mathbb{P}[(X, Y) \in \mathcal{T}] \geq 1 - \epsilon$$

and

$$2^{-n(1+h+\epsilon)} \leq \mathbb{P}[X = u, Y = v] \leq 2^{-n(1+h-\epsilon)}$$

for all $(u, v) \in \mathcal{T}$. We use $\mathcal{T}$ to define the following decoding rule:

- Suppose that $v \in \{0,1\}^n$ is received. If there exists a unique i such that $(U(i), v) \in \mathcal{T}$ then decode v as i. Otherwise decode as $U(1)$.

Using the Typical Set to Decode

Since $\mathcal{T}$ is a typical set, we expect that most of the time when we send $U(i)$, we receive a v such that $(U(i), v) \in \mathcal{T}$. Therefore decoding should succeed most of the time. To make this idea precise, we need two further properties of $\mathcal{T}$.

(a) Since $2^{-n(1+h+\epsilon)} \leq \mathbb{P}[X = u, Y = v]$ for all $(u, v) \in \mathcal{T}$, the same argument as Example 12.2 shows that $|\mathcal{T}| \leq 2^{n(1+h+\epsilon)}$.

(b) Suppose that $\widetilde{X}$ and $\widetilde{Y}$ are independently and uniformly distributed on $\{0,1\}^n$, so $\mathbb{P}[\widetilde{X} = u, \widetilde{Y} = v] = \frac{1}{2^n} \times \frac{1}{2^n} = \frac{1}{2^{2n}}$ for all $(u, v) \in \mathbb{F}_2^n$. Hence, by (a),

$$\mathbb{P}[(\widetilde{X}, \widetilde{Y}) \in \mathcal{T}] = \sum_{(u,v)\in\mathcal{T}} \mathbb{P}[\widetilde{X} = u, \widetilde{Y} = v]$$

$$\leq 2^{-2n} 2^{n(1+h+\epsilon)} = 2^{-n(1-h-\epsilon)}.$$

Bound on Error Probability

Let P_i be the probability that when $U(i)$ is sent it is incorrectly decoded. If this happens then either $(U(i), Y) \notin \mathcal{T}$, or there is some other codeword $U(j)$ such that $(u(j), Y) \in \mathcal{T}$. Therefore

$$P_i \leq \mathbb{P}[(U(i), Y) \notin \mathcal{T}] + \sum_{j \neq i} \mathbb{P}[(U(j), Y) \in \mathcal{T}].$$

- Since $U(i)$ is distributed uniformly at random, it is the same random variable as the X used in the AEP and so

$$\mathbb{P}[(U(i), Y) \notin \mathcal{T}] = \mathbb{P}[(X, Y) \notin \mathcal{T}] < \epsilon.$$

  (Note here that we are using both random models: channel *and* code.)
- For the second summand note that Y is independent of $U(j)$, and that, since $U(i)$ is distributed uniformly on $\{0,1\}^n$, so is Y. (This is the BSC(p)-version of Question 1 on Sheet 7.) Therefore (X, Y) is the same random variable as the $(\widetilde{X}, \widetilde{Y})$ in (b) and so $\mathbb{P}[(U(j), Y) \in \mathcal{T}] \leq 2^{-n(1-h-\epsilon)}$ for each j.

End of proof

Since $M = 2\lceil 2^{rn} \rceil \leq 2\big(2^{rn} + 1\big) = 2^{rn+1} + 2 \leq 2^{rn+2}$ we have

$$\begin{aligned}
\mathbb{E}_{\text{code}}[P_i] &< \epsilon + (M-1)2^{-n(1-h-\epsilon)} \\
&< \epsilon + 2^{rn+2-n(1-h-\epsilon)} \\
&= \epsilon + 2^{-n(1-h-r-\epsilon)+2}.
\end{aligned}$$

Since $r < 1 - h$, by choosing ϵ sufficiently small, we have $1 - h - r - \epsilon > 0$. Therefore by choosing r sufficiently large, we have $\mathbb{E}[P_i] < 2\epsilon$ for all i.

The remainder of the proof is as in the proof of Proposition 9.5: there is a particular code $C^\star$ of size M such that the average error probability P is at most 2ϵ. Choosing the best half of the codewords then gives a code of size $M/2 = \lceil 2^{rn} \rceil$ for which all the error probabilities are at most 2ϵ.

Source Coding with Errors

A memoryless source emits the bits 0 and 1 each with equal probability $\frac{1}{2}$. Thus

$$\mathbb{P}[S_1S_2S_3 = 000] = \mathbb{P}[S_1S_2S_3 = 001] = \ldots = \mathbb{P}[S_1S_2S_3 = 111] = \tfrac{1}{8}.$$

The per-symbol entropy is $H(\frac{1}{2}, \frac{1}{2}) = 1$, so Shannon's Noiseless Coding Theorem (Theorem 4.7(b)) says that the average length of any injective encoder for words of length r is at least r.

In this section we allow non-injective encoders. These lose some information about the source; correspondingly, the source can be compressed beyond the bound in Shannon's Noiseless Coding Theorem.

Useful Inequality

In the proof of Proposition 12.5 outlined in the printed notes, we need the inequality $1 - t \leq \mathrm{e}^{-t}$ and so $(1 - \frac{1}{M})^M \leq \mathrm{e}^{-1}$. In fact, when M is large, the two sides are very close.

Example 12.4

A lottery sells tickets numbered from $\{1, \ldots, T\}$. On the day of the draw, a random number is generated in this set: everyone whose ticket matches wins a prize. Let p_M be the probability that no-one wins when M people buy tickets. Then $p_M \leq \mathrm{e}^{-M/T}$. Moreover, $p_{\alpha T} \to \mathrm{e}^{-\alpha}$ as $T \to \infty$ for any $\alpha > 0$.

Exercise. Let W be the number of winning tickets. What is $\mathbb{E}[W]$?

Useful Inequality

In the proof of Proposition 12.5 outlined in the printed notes, we need the inequality $1 - t \leq \mathrm{e}^{-t}$ and so $(1 - \frac{1}{M})^M \leq \mathrm{e}^{-1}$. In fact, when M is large, the two sides are very close.

Example 12.4

A lottery sells tickets numbered from $\{1, \ldots, T\}$. On the day of the draw, a random number is generated in this set: everyone whose ticket matches wins a prize. Let p_M be the probability that no-one wins when M people buy tickets. Then $p_M \leq \mathrm{e}^{-M/T}$. Moreover, $p_{\alpha T} \to \mathrm{e}^{-\alpha}$ as $T \to \infty$ for any $\alpha > 0$.

Exercise. Let W be the number of winning tickets. What is $\mathbb{E}[W]$? *Answer:* each ticket has probability $1/T$ of winning, so by linearity of expectation, $\mathbb{E}[W] = \sum_{i=1}^{M} 1/T = M/T$.

Compression

Proposition 12.5

Let $D < \frac{1}{2}$ be given and let $h = H(D, 1-D)$. Let $\epsilon > 0$ be given. Provided n is sufficiently large, there is a binary code $C \subseteq \{0,1\}^n$ of size $2^{n(1-h)}$ and an encoder $f : \{0,1\}^n \to C$ such that

$$\mathbb{P}\big[d(f(S_1 \dots S_n), S_1 \dots S_n) \geq (D+\epsilon)\big] \leq \epsilon.$$

Above d denotes Hamming distance. Thus it is very likely that the codeword $e(S_1 \dots S_n)$ chosen to encode $S_1 \dots S_n$ differs from $S_1 \dots S_n$ in at most $(D+\epsilon)n$ bits. Allowing a probability D of error on each bit allows us to compress n bits into $n(1-h)$ bits.

Example of Compression

Example 12.6

A source emits 120 bits per second, each equally likely to be 0 and 1. If a noiseless channel can only send 80 bits per second then we must compress by a factor of $\frac{2}{3}$.

- Therefore D, the least possible bit error probability, must satisfy $1 - H(D, 1-D) = \frac{2}{3}$, or equivalently, $H(D, 1-D) = \frac{1}{3}$. Solving numerically we find that $D \approx 0.0615$. So this compression is feasible provided a bit error probability of about 6.1% is acceptable.

Example of Compression

Example 12.6

A source emits 120 bits per second, each equally likely to be 0 and 1. If a noiseless channel can only send 80 bits per second then we must compress by a factor of $\frac{2}{3}$.

- Therefore D, the least possible bit error probability, must satisfy $1 - H(D, 1-D) = \frac{2}{3}$, or equivalently, $H(D, 1-D) = \frac{1}{3}$. Solving numerically we find that $D \approx 0.0615$. So this compression is feasible provided a bit error probability of about 6.1% is acceptable.
- Compare this with the very naive encoder that simply forgets the final 40 bits each second. For each forgotten bit, there is a $\frac{1}{2}$ chance that the decoder makes the correct choice. So the average bit error probability is $(80 \times 0 + 40 \times \frac{1}{2})/120 = \frac{1}{6}$, or about 16.7%.

Equality and Diversity Survey

The Mathematics and ISG Equality and Diversity Committee wants to hear from you!

All students in the Mathematics Department and ISG are warmly encouraged to complete this survey. Go to:

- https://rhul.onlinesurveys.ac.uk/athena-swan-student-survey-mathematics-2019
- tinyurl.com/vof7lso
- use QR code below, or read your email for the link

The survey is linked to the Athena SWAN scheme. Its purpose is to promote gender equality in higher education for staff and students. Your responses will inform our Athena SWAN actions.

Athena SWAN GENDER CHARTER

§13 General sources and Lempel–Ziv encoding

So far we have only considered memoryless sources. We end by briefly considering general sources.

Definition 13.1

The *entropy* of a source S_1, $S_2 \ldots$ is

$$\lim_{r\to\infty} \frac{H(S_1, \ldots, S_r)}{r}$$

when this limit exists.

Example 13.2

(1) The memoryless binary source in Exercise 11.1 which emits 0 with probability $1-p$ and 1 with probability p has entropy $H(p, 1-p)$.

(2) Consider the binary source for which $\mathbb{P}[S_1 = 0] = \mathbb{P}[S_1 = 1] = \frac{1}{2}$ and $S_t = S_1$ for all $t \in \mathbb{N}$. Thus the source emits either $000\ldots$ or $111\ldots$ with equal probability. Its entropy is 0.

§13 General sources and Lempel–Ziv encoding

So far we have only considered memoryless sources. We end by briefly considering general sources.

Definition 13.1

The *entropy* of a source S_1, $S_2 \ldots$ is

$$\lim_{r\to\infty} \frac{H(S_1, \ldots, S_r)}{r}$$

when this limit exists.

Example 13.2

(3) Consider the binary source which starts by flipping a coin biased to lands heads with probability p. If the coin lands heads, it emits $111\ldots$. Otherwise it behaves as the source in (1). You are asked to show on Problem Sheet 9 that the entropy of this source is $(1-p)H(p, 1-p)$, and that it is not memoryless.

§13 General sources and Lempel–Ziv encoding

So far we have only considered memoryless sources. We end by briefly considering general sources.

Definition 13.1

The *entropy* of a source S_1, $S_2 \ldots$ is

$$\lim_{r\to\infty} \frac{H(S_1, \ldots, S_r)}{r}$$

when this limit exists.

The following lemma generalizes Example 13.2(1). A proof is outlined on Problem Sheet 9.

Lemma 13.3

The entropy of a memoryless source $S_1, S_2, \ldots$ exists and is $H(S_1)$.

Stationary Sources

Definition 13.4

Let $S_1, S_2, \ldots$ be a source emitting symbols in an alphabet $\mathcal{A}$. The source is *stationary* if for all $\alpha_1, \ldots, \alpha_\ell \in \mathcal{A}$ and distinct times $t_1, \ldots, t_\ell$ we have

$$\mathbb{P}[S_{t_1} = \alpha_1, \ldots, S_{t_\ell} = \alpha_\ell] = \mathbb{P}[S_{t_1+r} = \alpha_1, \ldots, S_{t_\ell+r} = \alpha_\ell]$$

for all $r \in \mathbb{N}_0$.

For example, memoryless sources (see Definition 3.1) are stationary since the symbols are independent and identically distributed.

Stationary Sources

Definition 13.4

Let $S_1, S_2, \ldots$ be a source emitting symbols in an alphabet $\mathcal{A}$. The source is *stationary* if for all $\alpha_1, \ldots, \alpha_\ell \in \mathcal{A}$ and distinct times $t_1, \ldots, t_\ell$ we have

$$\mathbb{P}[S_{t_1} = \alpha_1, \ldots, S_{t_\ell} = \alpha_\ell] = \mathbb{P}[S_{t_1+r} = \alpha_1, \ldots, S_{t_\ell+r} = \alpha_\ell]$$

for all $r \in \mathbb{N}_0$.

For example, memoryless sources (see Definition 3.1) are stationary since the symbols are independent and identically distributed. The sources in Example 13.2(2) and (3) are also stationary. Example 13.7 gives a source that may not be stationary.

Ergodic Sources

Definition 13.5

Fix a source $S_1, S_2, \ldots$ emitting symbols in an alphabet $\mathcal{A}$.

(i) The *frequency* of a word $\alpha_1 \ldots \alpha_\ell$, in the first r symbols, denoted $F_{\alpha_1 \ldots \alpha_\ell}(r)$ is the number of times $t \in \{1, \ldots, r - \ell + 1\}$ such that $S_t = \alpha_1, \ldots, S_{t+\ell-1} = \alpha_\ell$.

(ii) The source is *ergodic* if for all words $\alpha_1 \ldots \alpha_\ell$,

$$\lim_{r \to \infty} \frac{F_{\alpha_1 \ldots \alpha_\ell}(r)}{r} = \mathbb{P}[S_1 = \alpha_1, \ldots, S_\ell = \alpha_\ell].$$

Example 13.6

The source in Example 13.2(2) is not ergodic. The frequency $F_1(r)$ of the word 1 in the first r bits $S_1 \ldots S_r$ is either r or 0, with equal probability. Therefore

$$\frac{F_1(r)}{r} = \begin{cases} 1 & \text{with probability } \frac{1}{2} \\ 0 & \text{with probability } \frac{1}{2}. \end{cases}$$

Example 13.7

A source $S_1, S_2, \ldots$ has alphabet $\{0, 1\}$. If $S_t = 0$ then $S_{t+1} = 0$ with probability $\frac{3}{4}$ and otherwise 1; if $S_t = 1$ then S_{t+1} is equally likely to be 0 and 1. These 'transition probabilities' can be recorded in a matrix

$$T = \begin{pmatrix} \frac{3}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

By diagonalizing T one finds that [**typo in printed notes** t **not** s]

$$T^t = \frac{1}{3} \begin{pmatrix} 2 + \frac{1}{4^t} & 1 - \frac{1}{4^t} \\ 2 - \frac{2}{4^t} & 1 + \frac{2}{4^t} \end{pmatrix} \to \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix} \quad \text{as } t \to \infty.$$

Let h_t be the entropy of the first t bits emitted by the source. By the Chaining Rule (Lemma 7.6)

$$h_t = H(S_1, \ldots, S_{t-1}, S_t) = H(S_1, \ldots, S_{t-1}) + H(S_t | S_1, \ldots, S_{t-1}).$$

Since S_t depends on $S_1, \ldots, S_{t-1}$ only through S_{t-1}, this equation implies that $h_t = h_{t-1} + H(S_t | S_{t-1})$.

AEP for Stationary Ergodic Sources

Theorem 13.8

A stationary ergodic source satisfies the Asymptotic Equipartition Property, as stated in Definition 12.1, taking h to be the entropy of the source.

The proof of this theorem is beyond the scope of the course. You were asked to show in Question 2 on Sheet 8 that the AEP for memoryless sources implies Shannon's Noiseless Coding Theorem: it is routine to generalize this proof using Theorem 13.8 to prove Shannon's Noiseless Coding Theorem for a general stationary ergodic source.

Correction: Lossy Coding

For a memoryless source emitting 0 with probability $1 - p$ and 1 with probability p, and any $D < p$ one can encode with bitwise error probability D using $r(H(p) - H(D))$ bits to encode every r bits emitted by source.

In the lecture I argued that by Shannon's Noiseless Coding Theorem one could encode using $rH(p)$ bits, with no error, and then encode the codewords with bitwise error probability D compressing by a further $1 - H(D)$, to get $rH(p)(1 - H(D))$.

- This is a weaker result than the one above.
- Problem: bit errors change the codewords, so the error probability for the original source may be worse than D.

Lempel–Ziv Encoding: Example 13.9

Take

$$x = 10110\,10100\,010 = x_1 x_2 \ldots x_{13}.$$

We initialize the dictionary with the empty word $\varnothing$, to which we assign $0 \in \mathbb{N}_0$, so $\varnothing \mapsto 0$. (As a notational aid, values are written in red.) We then read the word from position 1.

At Step s, reading the word from position t, we take the longest subword $x_t \ldots x_{t+\ell-1}$ that is in the dictionary. (This could be the empty word when $\ell = 0$.) We then extend the dictionary by $x_t \ldots x_{t+\ell-1} x_{t+\ell} \mapsto s$ and continue from position $t + \ell + 1$.

s	From	Subword	New word
1	1	$\varnothing$	$1 \mapsto 1$
2	2	$\varnothing$	$0 \mapsto 2$
3	3	1	$11 \mapsto 3$
4	5	0	$01 \mapsto 4$
5	7	01	$010 \mapsto 5$
6	10	0	$00 \mapsto 6$
7	12	1	$10 \mapsto 7$

Example 13.9: Efficient Encoding of Dictionary

The final dictionary determines the word: just concatenate its elements in value order. To represent the dictionary efficiently, note that each new word is obtained by appending a bit to a word already in the dictionary.

For example at Step 5, 010 was obtained by appending 0 to 01, which had value 4. We may therefore replace 010 with $(4, 0)$. We then encode 4 in binary, as 100.

In general, at Step s we append to a word with value in $\{0, 1, \ldots, s-1\}$, so we need $\lceil \log_2 s \rceil$ bits to distinguish all the values.

Example 13.9: Efficient Encoding of Dictionary

s	New Word	(Value, New Bit)	$\lceil \log_2 s \rceil$	(Binary Value, New Bit)	Encoding
1	$1 \mapsto 1$	$(0, 1)$	0	$(\varnothing, 1)$	1
2	$0 \mapsto 2$	$(0, 0)$	1	$(0, 0)$	00
3	$11 \mapsto 3$	$(1, 1)$	2	$(01, 1)$	011
4	$01 \mapsto 4$	$(2, 1)$	2	$(10, 1)$	101
5	$010 \mapsto 5$	$(4, 0)$	3	$(100, 0)$	1000
6	$00 \mapsto 6$	$(2, 0)$	3	$(010, 0)$	0100
7	$10 \mapsto 7$	$(1, 0)$	3	$(001, 0)$	0011

Example 13.9: Efficient Encoding of Dictionary

s	New Word	(Value, New Bit)	$\lceil \log_2 s \rceil$	(Binary Value, New Bit)	Encoding
1	$1 \mapsto 1$	$(0, 1)$	0	$(\varnothing, 1)$	1
2	$0 \mapsto 2$	$(0, 0)$	1	$(0, 0)$	00
3	$11 \mapsto 3$	$(1, 1)$	2	$(01, 1)$	011
4	$01 \mapsto 4$	$(2, 1)$	2	$(10, 1)$	101
5	$010 \mapsto 5$	$(4, 0)$	3	$(100, 0)$	1000
6	$00 \mapsto 6$	$(2, 0)$	3	$(010, 0)$	0100
7	$10 \mapsto 7$	$(1, 0)$	3	$(001, 0)$	0011

The final encoding is therefore 100011101100001000010, or

$$10001110110000100010$$

as it should be written using just the uncoloured binary alphabet. Note that without the colour coding, the convention that $\lceil \log_2 s \rceil$ bits are used for the value at step s is essential to decode unambiguously. [**Corrected 2 April 2020: the binary encoding for the final step was wrong in the final bit (which I then omitted to put in the final encoding).**]

Algorithm 13.9: Lempel–Ziv

The input is a word $x_1 \dots x_r$ and the output is a binary word.

Initialize the dictionary: define $\varnothing \mapsto 0$ and set $t = 1$. Go to Step 1.

Step s: if $t > r$ then terminate.

- Read the word from position t. Choose ℓ maximal such that the dictionary contains $x_t \dots x_{t+\ell-1}$: suppose that $x_t \dots x_{t+\ell-1} \mapsto v$.
- Append the binary form of v of length $\lceil \log_2 s \rceil$ to the output word.
- If $t + \ell - 1 = r$ then terminate. Otherwise append $x_{t+\ell}$ and add $x_t \dots x_{t+\ell} \mapsto s$ to the dictionary. Go to Step $s + 1$.

Algorithm 13.9: Lempel–Ziv

The input is a word $x_1 \dots x_r$ and the output is a binary word.

Initialize the dictionary: define $\varnothing \mapsto 0$ and set $t = 1$. Go to Step 1.

Step s: if $t > r$ then terminate.

- Read the word from position t. Choose ℓ maximal such that the dictionary contains $x_t \dots x_{t+\ell-1}$: suppose that $x_t \dots x_{t+\ell-1} \mapsto v$.
- Append the binary form of v of length $\lceil \log_2 s \rceil$ to the output word.
- If $t + \ell - 1 = r$ then terminate. Otherwise append $x_{t+\ell}$ and add $x_t \dots x_{t+\ell} \mapsto s$ to the dictionary. Go to Step $s + 1$.

The only extra feature is that if x ends with a subword in the dictionary, we do not need to extend the dictionary, and we simply output the binary form of the value of the subword.

Algorithm 13.9: Lempel–Ziv

The input is a word $x_1 \dots x_r$ and the output is a binary word.

Initialize the dictionary: define $\varnothing \mapsto 0$ and set $t = 1$. Go to Step 1.

Step s: if $t > r$ then terminate.

- Read the word from position t. Choose ℓ maximal such that the dictionary contains $x_t \dots x_{t+\ell-1}$: suppose that $x_t \dots x_{t+\ell-1} \mapsto v$.
- Append the binary form of v of length $\lceil \log_2 s \rceil$ to the output word.
- If $t + \ell - 1 = r$ then terminate. Otherwise append $x_{t+\ell}$ and add $x_t \dots x_{t+\ell} \mapsto s$ to the dictionary. Go to Step $s + 1$.

The only extra feature is that if x ends with a subword in the dictionary, we do not need to extend the dictionary, and we simply output the binary form of the value of the subword.

All this can be done using the MATHEMATICA notebook `LempelZiv.nb` on Moodle.

Lempel–Ziv on a Highly Structured Word

In any example you are likely to have the patience to do by hand, the Lempel–Ziv encoding will almost certainly be longer than the original word.

Example 13.11

Consider the word $w^{(s)}$ of length $2s$ formed by repeating 01. For example,

$$w^{(10)} = 0101010101\,0101010101$$

shown split into two blocks of size 10 for readability. Let ℓ_s be the length of the Lempel–Ziv encoding $w^{(s)}$. Even for this highly regular word, it is not until $s = 28$, so length 56, that the Lempel–Ziv encoding is shorter. The table below, shows the ratio $\ell_s/2s$.

s	1	2	5	10	20	27	28	29	30	40	50	60	120
$\ell_s/2s$	1.5	1.5	1.6	1.2	1.1	1	0.982	1.017	0.983	0.875	0.830	0.783	0.621

Lempel–Ziv Theory and Practice

For a worked example of decoding, see Problem Sheet 9.

The Lempel–Ziv Algorithm was published in 1977. Later in 1991 it was proved that, for suitable sources, the algorithm achieves the bound in Shannon's Noiseless Coding Theorem, and so is optimal. Very, very roughly, one might expect this to be true because, by the AEP, a source of entropy h has a set of about 2^{hr} typical messages of length r, all of which enter the Lempel–Ziv dictionary; at this point in the algorithm the dictionary also has size about 2^{hr}, so $\lceil 2^{hr} \rceil + 1 \approx hr + 1$ bits are output for each subword of length r emitted by the source.

So the Lempel–Ziv Algorithm is a practical solution to the problem of source coding.