Coding of memoryless sources

Outline

1. Morse coding;
2. Definitions : encoding, encoding efficiency;
3. fixed length codes, encoding integers;
4. prefix condition;
5. Kraft and Mac Millan theorems;
6. First Shannon theorem for a discrete memoryless source source;
7. Simulating a random distribution with coin flips.

1. History

General idea: encoding the most frequent characters with short codewords as in the Morse code.

A	.-	N	-.	0	-----
B	-...	O	---	1	.----
C	-.-.	P	.--.	2	..---
D	-..	Q	--.-	3	...--
E	.	R	.-.	4	....-
F	..-.	S	...	5	.....
G	--.	T	-	6	-....
H	....	U	..-	7	--...
I	..	V	...-	8	---..
J	.---	W	.--	9	----.
K	-.-	X	-..-	.	.-.-.-
L	.-..	Y	-.--	,	--..--
M	--	Z	--..	?	..--..

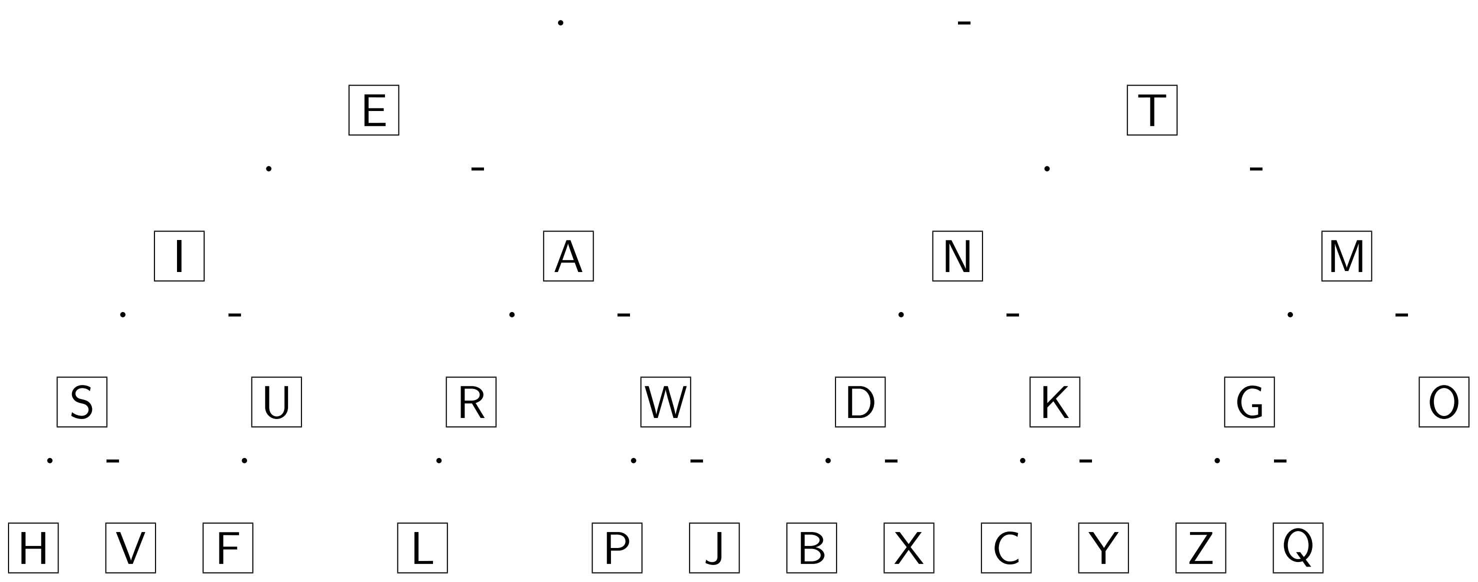
Actually ternary encoding (additional symbol to separate letters). Otherwise : impossible to distinguish

- "BAM" $\rightarrow$ "-....---"
- "NIJ" $\rightarrow$ "-....---"

Suited to humans, but unsuitable for modern communications techniques.

Morse Code

The Morse code can be represented by a binary tree. Each node, with the exception of the root is the encoding of a character.



2. Definitions

Code and Encoding

Let $\mathcal{X}$ be a finite alphabet.

Definition: A *code* over $\mathcal{X}$ is a mapping $\varphi : \mathcal{X} \to \{0,1\}^*$ (the set of all binary words).

Definition: A *codeword* is an element in $\varphi(\mathcal{X})$.

Definition: An *encoding* of $\mathcal{X}$ is a mapping $\psi : \mathcal{X}^* \to \{0,1\}^*$, that maps any finite sequence of letters in $\mathcal{X}$ to a binary sequence.

An encoding is associated to each code φ of $\mathcal{X}$ by

$$(x_1, x_2, \ldots, x_L) \to (\varphi(x_1) \parallel \varphi(x_2) \parallel \ldots \parallel \varphi(x_L))$$

Unambiguous encoding

Definition A code (resp. encoding) is *unambiguous* iff two distinct letters (resp. distinct sequences of characters) are encoded by two distinct binary words.

An ambiguous encoding yields information loss. Morse encoding:$\longrightarrow$ a separator : silence equal to a dash duration.

A code is *uniquely decodable* if the associated encoding is unambiguous.

	$\phi_1(a_k)$
a_1	0
a_2	010
a_3	01
a_4	10

Decoding 010 with a_2, a_1a_4, or a_3a_1 ?

Memoryless Source – Efficiency

A discrete source is a sequence $(X_i)_{i\in\mathbb{N}}$ of random variables over $\mathcal{X}$.

Definition : A source $X = (\mathcal{X}, p)$ is *memoryless* iff the X_i's are i.i.d.and their distribution is given by p. Its *entropy* is defined by $H(X) \overset{\text{def}}{=} -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x)$.

Definition : The *average codelength* of a code φ of a discrete memoryless source $X = (\mathcal{X}, p)$ is defined by

$$|\varphi| \overset{\text{def}}{=} \sum_{x\in\mathcal{X}} p(x)|\varphi(x)|$$

Definition: The *efficiency* of a code φ is defined by $E(\varphi) \overset{\text{def}}{=} \frac{H(X)}{|\varphi|}$.

Encoding efficiency

Let $(x_1, \dots, x_n)$ be a finite sequence of letters of $\mathcal{X}$, we denote by $p(x_1, \dots, x_n)$ its probability.

The *average codelength* of the encoding ψ is defined by the following limit (when it exists)

$$\mathcal{L}(\psi) \overset{\text{def}}{=} \lim_{n \to \infty} \frac{1}{n} \sum_{x_1, \dots, x_n} p(x_1, \dots, x_n) |\psi(x_1, \dots, x_n)|$$

Definition: Let X be a discrete memoryless source and ψ an encoding of X whose average codelength is well defined. The *efficiency* of ψ is equal to

$$E(\psi) \overset{\text{def}}{=} \frac{H(X)}{\mathcal{L}(\psi)}.$$

3. Fixed length codes

Fixed length codes: codes for which all codewords are of the same length (say n).

Proposition 1. *For any unambiguous code of codelength n of a finite alphabet $\mathcal{X}$ of cardinality K, we have*

$$\log_2 K \leq n$$

Proof

If $2^n < K$, the code can not be one-to-one. Therefore $n \geq \log K$.

The efficiency of such a code is limited by $H(X)/\log_2 K$ (which is equal to 1 when X uniformly distributed).

Upper Bound

Proposition 2. *For any source X of size K, there exists an unambiguous code of codelength n such that*

$$\log_2 K \leq n < 1 + \log_2 K$$

Proof

Let n be the smallest integer such that $2^n \geq K$, then for any symbol a_k one can associate the binary representation of k, over n bits. By definition of n we also have $2^{n-1} < K$, that is

$$n < 1 + \log_2 K$$

Efficiency of fixed length coding

Corollary 1. *There exists an unambiguous encoding of $\mathcal{X}$ whose efficiency is arbitrarily close to $H(X)/\log_2 K$.*

Proof The source is encoded by taking packets of L (in other words the new alphabet is $\mathcal{X}^L$). There exists a code of length n' for which

$$L \log_2 K \leq n' < L \log_2 K + 1$$

The length $\mathcal{L}$ per letter is equal to

$$\frac{n'}{L} \to \log_2(K)$$

Therefore

$$E = H/\mathcal{L} \to H/\log_2 K$$

Integer Encoding

Consider a source $\mathcal{X} = \{0, 1, \ldots, 9\}$ where the integers are uniformly distributed.

A fixed length code for such a source has length ≥ 4. For instance:

letter	codeword
0	0000
1	0001
2	0010
3	0011
4	0100

letter	codeword
5	0101
6	0110
7	0111
8	1000
9	1001

The efficiency of such a code is equal to $H(X)/4 = (\log_2 10)/4 = 0.83$

By taking groups of 3 integers, the cardinality of the alphabet becomes 1000 which can be encoded with 10 bits. The efficiency becomes

$$\frac{\log_2(1000)}{10} = 0.9965$$

4. Variable length codes

Recall that a code is *uniquely decodable* if the associated encoding is one-to-one.

Prefix condition
No codeword is the prefix of another codeword.

Definition A code is *prefix, or instantaneous* if the prefix condition is met.

Proposition 3. *Any prefix code is uniquely decodable.*

There exist uniquely decodable codes which are not prefix.

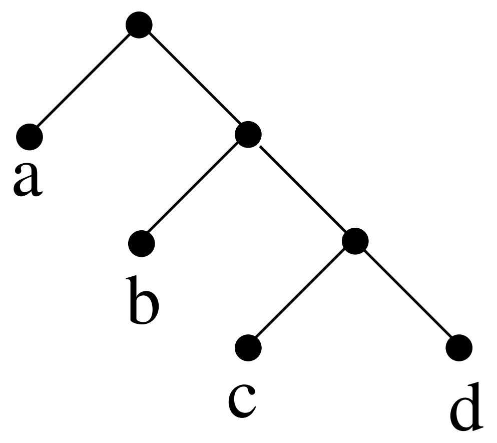
Exercise

$\mathcal{X}$	$\phi_0(a_k)$	$\phi_1(a_k)$	$\phi_2(a_k)$	$\phi_3(a_k)$	$\phi_4(a_k)$
a_1	0	0	1	0	10
a_2	11	010	10	10	00
a_3	11	01	100	110	11
a_4	10	10	1000	111	110

Which code are ambiguous, unambiguous, uniquely decodable, instantaneous ?

Tree associated to a prefix code

For every prefix prefix code there exists a binary tree whose codewords are the leaves of the tree (necessary and sufficient condition).

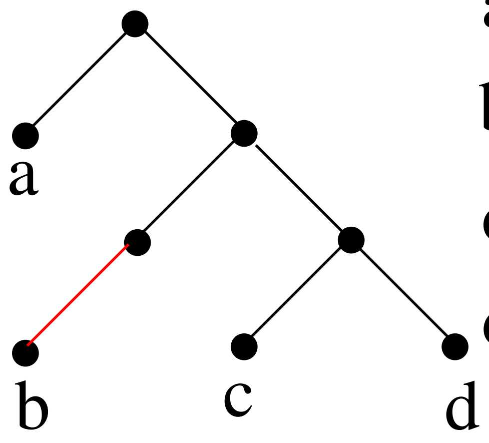


a : 0

b : 10

c : 110

d : 111



a : 0

b : 100

c : 110

d : 111

The code is irreducible when every tree node has either 0 or 2 descendants.

The irreducible codes are preferred: they are shorter.

5. Kraft inequality – Mac Millan's Theorem

Theorem 1. [(Kraft)] *There exists a prefix code with codewords of length* $n_1, n_2, \ldots, n_K$ *if and only if*

$$\sum_{k=1}^{K} \frac{1}{2^{n_k}} \leq 1.$$

Theorem 2. [(Mac Millan)] *There exists a uniquely decodable code with codewords of length r* $n_1, n_2, \ldots, n_K$ *if and only if*

$$\sum_{k=1}^{K} \frac{1}{2^{n_k}} \leq 1.$$

Proof of Kraft's theorem

$\Longrightarrow$ (If there exists a prefix code, then $\sum_{k=1}^{K} \frac{1}{2^{n_k}} \leq 1$)

Let $N = \max n_l$.

Consider the complete binary tree of depth N. By descending from the root, if codeword e_k is encountered all its descendants are removed.

If n_k is the depth of e_k, 2^{N-n_k} leaves of the complete binary tree are removed. The total number of leaves in this tree is 2^N.

We therefore have

$$\sum_k 2^{N-n_k} \leq 2^N$$

which leads to $\sum 2^{-n_k} \leq 1$.

Proof of Kraft's theorem (II)

$\Longleftarrow$ (If $\sum_{k=1}^{K} \frac{1}{2^{n_k}} \leq 1$ there exists a prefix code).

Sort the n_k's in increasing order. Let $N = \max n_k$. Consider the binary complete tree of depth N.

Algorithm : *At step k, take the first node (in lexicographic order) at depth n_k, and remove its descendants.*

At step k at most $\sum_{1 \leq i < k} 2^{n_k - n_i}$ nodes were removed at depth n_k. Note that

$$\sum_{1 \leq i < k} 2^{n_k - n_i} = 2^{n_k} \sum_{1 \leq i < k} 2^{-n_i} < 2^{n_k}.$$

This implies the correctness of the algorithm : it is always possible to find nodes at depth n_k at the k-th step.

Mac Millan's Theorem

Theorem 3. *(Mac Millan) If there exists a uniquely decodable code then* $\sum_{k=1}^{K} \frac{1}{2^{n_k}} \leq 1$.

Let ϕ be a uniquely decodable and Φ be its associated encoding. We have $\Phi(x_1, \dots, x_L) = \phi(x_1)||\dots||\phi(x_L)$. Let $m = \max_{x \in \mathcal{X}} l(x)$.

$$
\begin{aligned}
(\sum_{x} 2^{-|\phi(x)|})^L &= \sum_{x_1,\dots,x_L} 2^{-|\phi(x_1)|} \dots 2^{-|\phi(x_L)|} \\
&= \sum_{x_1,\dots,x_L} 2^{-|\Phi(x_1,\dots.x_L)|} \\
&= \sum_{i=1}^{mL} a_L(i) 2^{-i}
\end{aligned}
$$

with $a_L(i) = |\{(x_1, \dots, x_L); |\Phi(x_1, \dots, x_L)| = i\}|$.

$$\left(\sum_{x} 2^{-|\phi(x)|}\right)^{L} = \sum_{i=1}^{mL} a_L(i) 2^{-i}$$

We have $a_L(i) \leq 2^{i}$, otherwise two words would have the same encoding, which is impossible by assumption on the code.

This implies

$$\left(\sum_{x} 2^{-|\phi(x)|}\right)^{L} \leq mL$$

that is

$$\left(\sum_{x} 2^{-|\phi(x)|}\right) \leq (mL)^{1/L} \to 1, \text{ when } L \to \infty.$$

$\Rightarrow$ Uniquely decodable codes are not better than prefix codes.

6. 1st Shannon theorem (for discrete memoryless sources)

Theorem 4. *1. For any discrete memoryless source with entropy H encoded by means of a uniquely decodable code of average codelength $\bar{n}$, we have $\bar{n} \geq H$.*

2. For every discrete memoryless source with entropy H, there exists a prefix code of average codelength $\bar{n}$ such that $H \leq \bar{n} < H + 1$.

Proof of 1.

Let $X = (\mathcal{X}, p)$ be a discrete memoryless source with entropy H. Let φ be a uniquely decodable code for X. Let us show that $|\varphi| - H \geq 0$. Let $S \overset{\text{def}}{=} \sum_{x \in \mathcal{X}} 2^{-|\varphi(x)|}$ and q be the probability distribution over $\mathcal{X}$ given by $q(x) \overset{\text{def}}{=} \frac{2^{-|\varphi(x)|}}{S}$.

$$
\begin{aligned}
|\varphi| - H &= \sum_{x \in \mathcal{X}} p(x)|\varphi(x)| + \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \\
&= \sum_{x \in \mathcal{X}} p(x) \left(\log_2 p(x) + \log_2 2^{|\varphi(x)|} \right) \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{2^{-|\varphi(x)|}} \\
&= D(p||q) + \log_2 \frac{1}{S} \\
&\geq 0 \ \ \text{(non negativity of } D \text{ and } S \leq 1 \text{)}
\end{aligned}
$$

Proof of 2.

Let $\ell(x)$ be the unique integer such that $2^{-\ell(x)} \leq p(x) < 2^{-\ell(x)+1}$. We have

$$\sum_{x\in\mathcal{X}} 2^{-\ell(x)} \leq \sum_{x\in\mathcal{X}} p(x) = 1,$$

Kraft $\Longrightarrow$ there exists a prefix code φ with $|\varphi(x)| = \ell(x), x \in \mathcal{X}$.

Taking the logarithm gives $\log_2 p(x) < -\ell(x) + 1$, that is

$$|\varphi(x)| = \ell(x) < \log_2 \frac{1}{p(x)} + 1.$$

By averaging, we obtain

$$|\varphi| = \sum_x p(x)|\varphi(x)| < \sum_x p(x) \left(\log_2 \frac{1}{p(x)} + 1\right) = H + 1.$$

Dyadic case

Dyadic distribution : $p_i = 2^{-l_i}$.

Proposition 4. *There exists a code such that $|\phi| = H$ iff the distribution is dyadic.*

First point in the previous proof (when do we have equality?).

Shannon's 1st theorem

Theorem *(Shannon) For any discrete memoryless source X, there exists an unambiguous encoding whose efficiency is arbitrarily close to 1.*

Proof *Consider ϕ_i a code for the source X^i obtained by grouping symbols of X in packets of size i. We have $H(X^i) = iH$ and therefore :*

$$iH \leq |\phi_i| < iH + 1.$$

The packet length is fixed to l, and the word $x_1, \ldots, x_m, x_r$, of length $L = ml + r$, is encoded as follows

$$\phi_l(x_1)|| \ldots ||\phi_l(x_m)||\phi_r(x_r)$$

Proof (cont'd)

$L = ml + r$

$$\phi_l(x_1)||\ldots||\phi_l(x_m)||\phi_r(x_r)$$

The average codelength is given by

$$\begin{aligned} m|\phi_l| + |\phi_r| &\leq m(Hl+1) + rH + 1 \\ &= (ml+r)H + m + 1 = LH + m + 1 \end{aligned}$$

The average codelength per symbol is therefore

$$\mathcal{L}(\Phi) = \frac{m|\phi_l| + |\phi_r|}{L} \leq H + \frac{m+1}{ml+r} \leq H + \frac{1}{l} + \frac{1}{ml}$$

Therefore

$$\overline{\lim_{L\to\infty}}\, \mathcal{L}(\Phi) \leq H + \frac{1}{l}$$

The upper bound goes to H, when $l \to \infty$.

Exercise : Huffman Questions

Consider a random number X between 1 and n. Let p_i be the probability that $X_i = i$. We are asked to determine the value of X as quickly as possible by asking questions. Any yes-no question you can think of is admissible.

1. *Give a lower bound on the minimum average number of questions required.*
2. *Give an upper bound (within 1 question) on the minimum average number of questions required.*
3. *Find an optimal set of questions in the following case.*

p_1	p_2	p_3	p_4	p_5	p_6
$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

7. Simulating a random distribution with coin flips

Let $\begin{array}{c|c} a_1 & 1/2 \\ a_2 & 1/4 \\ a_3 & 1/4 \end{array}$ *and a fair coin* $p(Z=0) = p(Z=1) = 1/2$.

Two ways to proceed

1. *flip the coin twice*
   - *if we obtain* 00 *or* 11 *output* a_1,
   - *for* 01 *output* a_2, *and for* 10 *output* a_3

     average number of coin flips : 2
2. *flip the coin once*
   - *if* 0 *then* a_1,
   - *otherwise, flip the coin again, and if* 01 *output* a_2 *and* a_3 *otherwise.*

     average number of coin flips: $1/2 + 1/4 \cdot 2 + 1/4 \cdot 2 = 1,5$.

The last method is more efficient, we recover the entropy $H = 1/2 \log(2) + 1/4 \log(4) + 1/4 \log(4) = 3/2$.

The problem

We have $X = (\mathcal{X}, p(x))$, *and a sequence of r.v.* Z_i, $p(Z_i = 0) = p(Z_i = 1) = 1/2$.

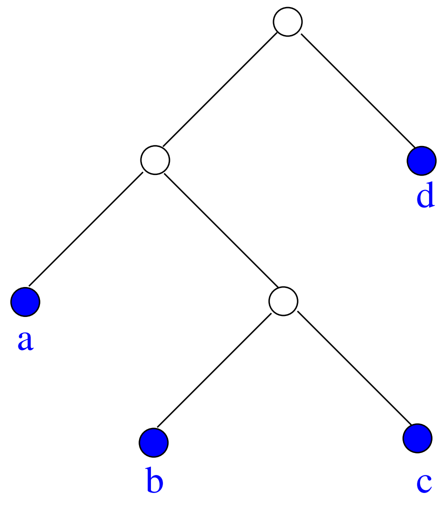
We want to draw $x \in \mathcal{X}$ *with probability* $p(x)$ *by using the* Z_i*'s.*

Drawing $z_1, \dots, z_m, \dots$ *can be viewed as a walk in a binary tree where the leaves have label* $x \in \mathcal{X}$.

When a leaf is encountered, output the corresponding letter.

The problem is to construct the optimal tree such that the average walk length is the smallest possible and such that the leaves are output with the right probability.

An example



This tree enables to simulate the source $\mathcal{X} = \{a, b, c, d\}$ *with probabilities*

x	p(x)
a	1/4
b	1/8
c	1/8
d	1/2

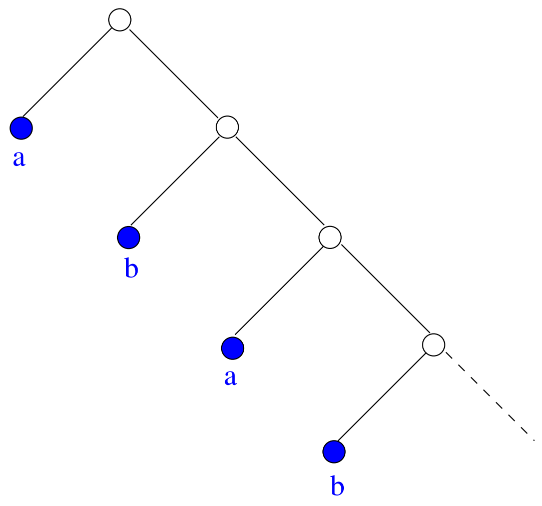
By source simulation we mean such a binary tree.

The average number of coin flips is equal to

$$1/2 \cdot 1 + 1/4 \cdot 2 + 1/8 \cdot 3 + 1/8 \cdot 3 = 1.75 = H(X)$$

Infinite tree

For the source $\mathcal{X} = \{a, b\}$, with $p(a) = 2/3$ and $p(b) = 1/3$, we have an infinite tree



It is readily checked that

$$\begin{aligned} p(a) &= \frac{1}{2} + \frac{1}{8} + \cdots + \frac{1}{2^{2i+1}} + \cdots = \frac{2}{3} \\ p(b) &= \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{2^{2i}} + \cdots = \frac{1}{3} \end{aligned}$$

The average number of coin flips is equal to $\sum \frac{i}{2^i} = 2$. *Notice that* $H(X) = h_2(1/3) = 0.918$.

Average depth

***Lemma* 1.** *Let T be a binary tree with a probability distribution p on the leaves such that*

each leaf y at depth $k(y)$ has probability $2^{-k(y)}$.

Then the average number of coin flips to simulate such a distribution is equal to $H(Y)$ where Y is distributed according to p.

Proof

$$\bar{T} = \sum_{y \in \mathcal{Y}} k(y) 2^{-k(y)}$$

On the other hand

$$H(Y) = -\sum_{y \in \mathcal{Y}} \frac{1}{2^{k(y)}} \log_2 \frac{1}{2^{k(y)}} = \sum_{y \in \mathcal{Y}} k(y) 2^{-k(y)}$$

A first inequality

Proposition 5. *The average number of coin tosses for generating a random variable X is greater than or equal to $H(X)$.*

Proof

Such an algorithm is represented by a binary tree : to each leaf y a symbol x is associated. Let Y be a random variable taking its values among the set of leaves and $\mathbf{Prob}(Y = y) = 2^{-k(y)}$ where k is the depth of leaf y.

Notice that

$$\bar{T} = H(Y)$$

where $\bar{T}$ is the average number of coin tosses.

On the other hand, X is a function of Y: $X = f(Y)$. Each leaf determines the value that X should take. Applying a function to a random variable can only reduce its entropy, therefore

$$H(X) \leq H(Y) = \bar{T}.$$

Dyadic case

Dyadic distribution : the probabilities satisfy $p_i = 2^{-l_i}$.

In this case, there exists a prefix code ϕ such that $l(\phi(x)) = -\log_2(p(x))$, and $|\phi| = H$.

Theorem 5. *Let X be a r.v. with a dyadic distribution. There exists a way to toss fair coins such that the average number of coin tosses $\bar{T}$ satisfies $\bar{T} = H(X)$.*

Reminder: *There exists a prefix code such that $|\varphi(x)| = l(x) = -\log_2 p(x)$.*

The depth of leaf x is exactly $-\log_2(p(x))$, therefore the average depth is equal to

$$\sum_x p(x)\,(-\log_2(p(x))) = H(X)$$

General case

Theorem 6. *The average number of coin tosses $\bar{T}$ to simulate a r.v. X verifies*

$$H(X) \leq \bar{T} < H(X) + 2.$$