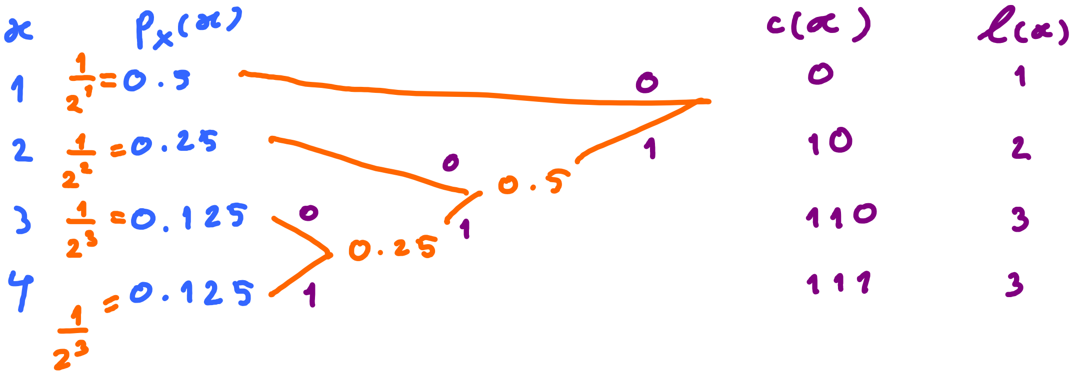
# 11.2 Huffman Coding

Wednesday, September 26, 2012
10:08 AM

Steps : Create a (binary) tree by repeatedly combining the probabilities of the two least likely (source) symbols.



| $x$ | $p_X(x)$ | $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| 1 | $\frac{1}{2^1} = 0.5$ | 0 | 1 |
| 2 | $\frac{1}{2^2} = 0.25$ | 10 | 2 |
| 3 | $\frac{1}{2^3} = 0.125$ | 110 | 3 |
| 4 | $\frac{1}{2^3} = 0.125$ | 111 | 3 |

$$\mathbb{E}[\ell(X)] = \underbrace{\frac{1}{2}\times 1 + \frac{1}{4}\times 2}_{1} + \underbrace{\frac{1}{8}\times 3 + \frac{1}{8}\times 3}_{3/4} = 1.75 \text{ bits/symbol}$$

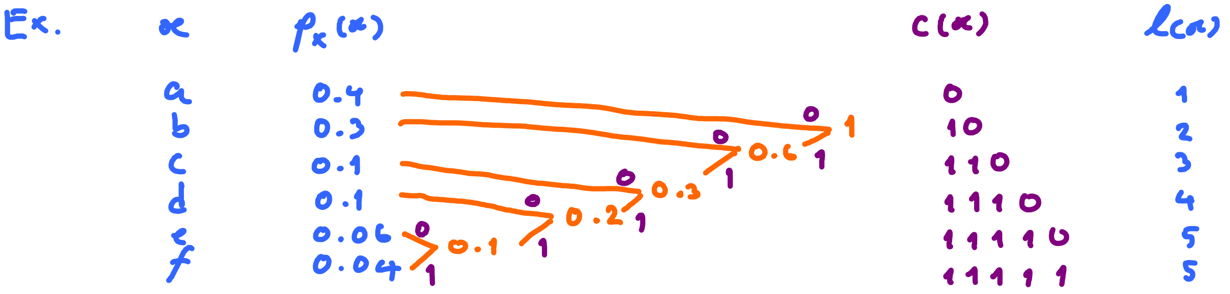
Observation : $p_X(x) = \frac{1}{2^{\ell(x)}}$

$$\ell(x) = \log_2 \frac{1}{p_X(x)} = -\log_2 p_X(x)$$

$$\mathbb{E}[\ell(X)] = \underbrace{\mathbb{E}[-\log_2 p_X(X)] = \sum_x p_X(x)\left(-\log_2 p_X(x)\right)}$$

$H(X)$ ← entropy of RV $X$.

Ex.



| $x$ | $p_X(x)$ | $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| a | 0.4 | 0 | 1 |
| b | 0.3 | 10 | 2 |
| c | 0.1 | 110 | 3 |
| d | 0.1 | 1110 | 4 |
| e | 0.06 | 11110 | 5 |
| f | 0.04 | 11111 | 5 |

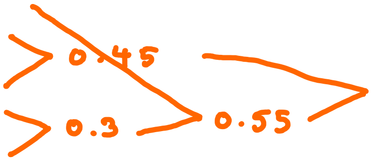
$\mathbb{E}[\ell(X)] = 0.4 + 0.6 + 0.3 + 0.4 + 0.3 + 0.2 = 2.2$ bits/symbol

✓

$H(X) = \sum_{x} p_X(x)\left(-\log_2 p_X(x)\right) = 2.1435$

Ex.

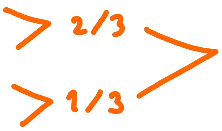
| $x$ | $p_X(x)$ | $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| 1 | .25 | | 2 |
| 2 | .25 | | 2 |
| 3 | .2 | | 2 |
| 4 | .15 | | 3 |
| 5 | .15 | | 3 |



$\mathbb{E}[\ell(X)] = 2.3$

Ex.

| $x$ | $p_X(x)$ | $c(x)$ | $\ell(x)$ |
|---|---|---|---|
| ก | 1/3 | | 2 |
| ข | 1/3 | | 2 |
| ค | 1/4 | | 2 |
| ง | 1/12 | | 2 |



$\mathbb{E}[\ell(X)] = 2$

Defn. A code is optimal for a given source if $\mathbb{E}[\ell(X)]$ (resulted from using that code) is minimal.

Thm: Huffman code is optimal among UD codes.