(bits) lenght

1: In base 2, $U(X)$ is the min. avg. ~~of bits~~ to encode a message. Optimized encoding.

in bits compressed

Example: We have 8 balls in a basket
- 4 red → 1/2
- 2 blue → 1/4
- 1 black → 1/8
- 1 yellow → 1/8

Then $U(X) = -\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{4}\log_2(\frac{1}{4}) - \frac{2}{8}\log_2(\frac{1}{8})$

$= \frac{1}{2} + \frac{1}{2} + \frac{3}{4} = 7/4$

The encoding is set as
- red → 0
- blue → 10
- black → 110
- yellow → 111

Taking into how often ~~this~~ each color can be draw, we have an avg. lenght of

$\underbrace{\frac{1}{2}}_{\text{red}}(1) + \underbrace{\frac{1}{4}}_{\text{blue}}(2) + \underbrace{\frac{1}{8}}_{\text{black}}(3) + \underbrace{\frac{1}{8}}_{\text{yellow}}(3) = \frac{1}{2} + \frac{1}{2} + \frac{6}{8} = 7/4$

So why not this encoding? ⇒
- red → 0
- blue → 1
- black → 01
- yellow → 10

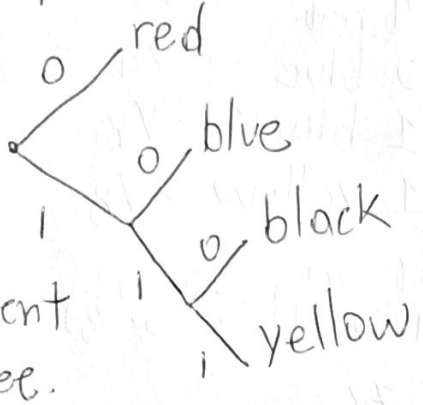
the avg. lenght will be smaller!

The reason is that this is NOT an effective encoding (it is ambiguous).

The sequence for example 01 is it a single black black ball or a red followed by a blue??

In our propose ending there is always a unique way to decrypt the message.

a way to find the optimized encoding in bits is to use a binary tree, assinging ~~an branch~~ end node to an event and using top branches for ~~even~~ the most probable events. and no event has a parent event in the tree.



2. In base 2, H corresponds to the avg. amount of binary questions to get the draw event.

In our case we will ask first the most probable color and continue with the next one.

is it a red ball? $1/2$ of the time is correct
is it a blue ball? $1/4$ of the time
is it a black ball? $2/8$ of the time

then the min. Avg. (remaining) of question binary is

$$\frac{1}{2}(1 \text{ question}) + \frac{1}{4}(2 \text{ questions}) + \frac{2}{8}(3 \text{ questions})$$

$$= 1 + \frac{6}{8} = \frac{1}{1} + \frac{3}{4} = 1.75$$