For a source with alphabet $A = \{a_1, a_2, \cdots, a_K\}$ and probability model: $\{P(a_1), P(a_2), \cdots, P(a_K)\}$, then the average codeword length:

(ACL) is given by: $$\bar{l} = \sum_{i=1}^{K} P(a_i)\, l_i$$

It can be shown that: $$H(S) \leq \boxed{\bar{l} \leq H(S) + 1}$$ ← Prove the upper bound

Define a code with $l_i = \left\lceil \log_2 \frac{1}{P(a_i)} \right\rceil$ ← ceiling function

Matlab:

Y = ceil(X) rounds each element of x to the nearest integer greater than or equal to that element.
\>> ceil(1.1)
ans =
  2 > 1.1 $\Rightarrow 1.1 \leq \text{ceil}(1.1) < 1.1 + 1$

Hence $$\log_2 \frac{1}{P(a_i)} \leq l_i < \log_2 \frac{1}{P(a_i)} + 1$$
$$\Downarrow$$
$$\log_2 P(a_i) - 1 < -l_i \leq \log_2 P(a_i)$$
$$2^{\log_2 P(a_i) - 1} = \frac{P(a_i)}{2} < 2^{-l_i} \leq 2^{\log_2 P(a_i)} = P(a_i)$$
$$\Downarrow$$
$$K(C) = \sum_{i=1}^{K} 2^{-l_i} \leq \sum_{i=1}^{K} P(a_i) = 1 \Rightarrow \text{K-M Inequality holds}$$

K-M Inequality:

(2) If K(C) ≤ 1, then we can always construct a prefix code with codeword lengths being $l_1, l_2, \cdots, l_K$.

$$\bar{l} = \sum_{i=1}^{K} p(a_i)\, l_i < \sum_{i=1}^{K} p(a_i) \left[\log_2 \frac{1}{p(a_i)} + 1\right] = H(S) + \underbrace{\sum_{i=1}^{K} p(a_i)}_{1} = H(S) + 1$$

Since $\log_2 \frac{1}{p(a_i)} \leq l_i < \log_2 \frac{1}{p(a_i)} + 1$

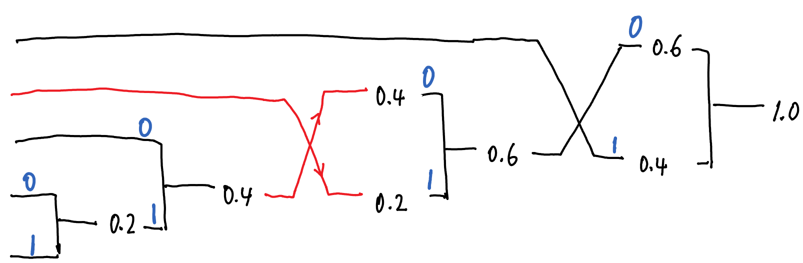
Thus, $\bar{l} < H(S) + 1$.

In summary, $H(S) \leq \bar{l} \leq H(S) + 1$.

$$l_i = \left\lceil \log_2 \frac{1}{p(a_i)} \right\rceil$$ ← ceiling function

Consider example:
Alphabet = $\{a_1, a_2, a_3, a_4, a_5\}$

	Codeword	Symbol	Prob
$l_1 = \lceil \log_2 \frac{1}{0.4} \rceil = 2$	1	a_1	0.4
$l_2 = \lceil \log_2 \frac{1}{0.2} \rceil = 3$	01	a_2	0.2
$l_3 = \lceil \ldots \rceil = 3$	000	a_3	0.2
$l_4 = \lceil \log_2 \frac{1}{0.1} \rceil = 4$	0010	a_4	0.1
$l_5 = \lceil \ldots \rceil = 4$	0011	a_5	0.1



Avg. Codeword Length (ACL)

= 1 x 0.4 + 2 x 0.2 + 3 x 0.2 + 4 x 0.1 + 4 x 0.1 = 2.2 bits/symbol

Entropy of the source:
H(0.4, 0.2, 0.2, 0.1, 0.1) = 2.1219 bits/symbol $= H(S)$

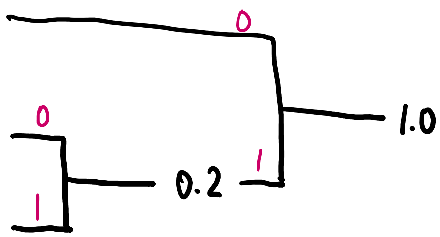
→ $\bar{l}$ based on $l_i = \left\lceil \log_2 \frac{1}{p(a_i)} \right\rceil$

$= 2 \times 0.4 + 2 \times (3 \times 0.2) + 2 \times (4 \times 0.1) = 2.8$ bits/symbol $< H(S) + 1 = 3.1219$ bits/symbol

Huffman Coding based on Extended Alphabet:

Example: Source $A = \{a_1, a_2, a_3\}$

Codeword		
0	a_1	0.8
10	a_2	0.18
11	a_3	0.02



$$\bar{l} = 1 \times 0.8 + 2 \times 0.18 + 2 \times 0.02 = 1.2 \text{ bits/symbol} < H(A)+1$$

$$H(A) = 0.8157 \text{ bit/symbol}$$

```
>> -0.8*log2(0.8)-0.18*log2(0.18)-0.02*log2(0.02)
ans =
   0.8157
```

Redundancy = $\bar{l} - H(A) = 0.3843$ bit/symbol

Consider symbol blocking (assuming that the symbols are independent):

Letters	Prob's
$a_1 a_1$	$P(a_1) \cdot P(a_1) = 0.8 \times 0.8 = 0.64$
$a_1 a_2$	$P(a_1) \cdot P(a_2) = 0.8 \times 0.18 = 0.144$
$\vdots$	$\vdots$
$a_3 a_3$	$P(a_3) \cdot P(a_3) = 0.02 \times 0.02 = 0.0004$

```
>> AA=[0.8*0.8,0.8*0.18,0.8*0.02,0.18*0.8,0.18*0.18,0.18*0.02,0.02*0.8,0.02*0.18,0.02*0.02];
>> sum(AA)
ans =
   1.0000
>> S = 0; for i = 1:9 S = S - AA(i)*log2(AA(i)); end;
>> S
S =
   1.6315
```

$$H(AA) = 1.6315 \text{ bits/two symbols} \overset{\Delta}{\approx} H(A) = 0.8157 \text{ bit/symbol}$$

Next, construct a Huffman code based on the extended alphabet:

```
>> symbols = 1:9;
>> [dict,avglen] = huffmandict(symbols,AA);
>> avglen

avglen =

   1.7228
```

bits / two symbols

```
>> 1.7228/2
ans =
   0.8614
```

$\Downarrow$

$$0.8614 \text{ bit/symbol} \ll \bar{l} = 1 \times 0.8 + 2 \times 0.18 + 2 \times 0.02 = \underbrace{1.2 \text{ bits/symbol}}_{\text{w/o blocking}}$$

Redundancy : $1.7228 - H(AA) = 1.7228 - 1.6315 = 0.0913$ bit/symbol

- In general, if we encode a sequence of symbols by generating one codeword for every n symbols, then there are m^n combination of n symbols, where m is the number of distinct symbols in the alphabet of the source.

Extended Alphabet:

$$A^{(n)} = \{ \underbrace{a_1 a_1 a_1 \cdots a_1}_{n \text{ times}}, \underbrace{a_1 a_1 \cdots a_2}_{n \text{ times}}, \cdots, \underbrace{a_m a_m \cdots a_m}_{n \text{ times}} \}$$

Construct the Huffman code on the extended alphabet:

$$H(A^{(n)}) \le R^{(n)} < H(A^{(n)}) + 1$$

$$\frac{H(A^{(n)})}{n} \le \frac{R^{(n)}}{n} < \frac{H(A^{(n)})}{n} + \frac{1}{n}$$

where

$$\underbrace{H(A^{(n)})}_{\text{Joint Entropy}} = - \sum_{i_1=1}^{m} \sum_{i_2=1}^{m} \cdots \sum_{i_n=1}^{m} \underbrace{p(a_{i_1}, a_{i_2}, \cdots, a_{i_n})}_{\text{Joint PMF}} \log_2 p(a_{i_1}, a_{i_2}, \cdots, a_{i_n})$$

If we assume that the symbols are independent, then

$$P(a_{i_1}, a_{i_2}, \cdots, a_{i_n}) = \prod_{k=1}^{n} P(a_{i_k})$$

$$\log_2 P(a_{i_1}, a_{i_2}, \cdots, a_{i_n}) = \sum_{k=1}^{n} \log_2 P(a_{i_k})$$

Thus,

$$\underbrace{H(A^{(n)})}_{\text{Joint Entropy}} = -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} \underbrace{P(a_{i_1}, a_{i_2}, \cdots, a_{i_n})}_{\text{Joint PMF}} \log_2 P(a_{i_1}, a_{i_2}, \cdots, a_{i_n})$$

$$= -\sum_{i=1}^{m} P(a_{i_1}) \log_2 P(a_{i_1}) - \sum_{i=1}^{m} P(a_{i_2}) \log_2 P(a_{i_2}) - \cdots - \sum_{i=1}^{m} P(a_{i_n}) \log_2 P(a_{i_n})$$

$$= n H(A)$$

Thus,

$$\frac{H(A^{(n)})}{n} \leq \frac{R^{(n)}}{n} < \frac{H(A^{(n)})}{n} + \frac{1}{n}$$

$$\Downarrow$$

$$\frac{H(A^{(n)})}{n} = \frac{n H(A)}{n} = H(A) \leq \underset{\substack{\uparrow \\ \text{(bits per symbol)}}}{R} < H(A) + \boxed{\frac{1}{n}}$$

benefit of using symbol blocking

$$n \uparrow \Rightarrow \frac{1}{n} \downarrow$$

Use m = 2, n = 2, as an example.

$$A = \{a_1, a_2\}, \text{ and } p(a_1) + p(a_2) = 1.$$

$$H(A^{(n)}) = -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} p(a_{i_1}, a_{i_2}, \cdots, a_{i_n}) \log_2 p(a_{i_1}, a_{i_2}, \cdots, a_{i_n})$$

$$= -\sum_{i_1=1}^{2}\sum_{i_2=1}^{2} p(a_{i_1}, a_{i_2}) \log_2 p(a_{i_1}, a_{i_2}), \text{ where}$$

$$p(a_{i_1}, a_{i_2}) = p(a_{i_1}) \cdot p(a_{i_2})$$

$$= -p(a_1, a_1)\log_2 p(a_1, a_1) - p(a_1, a_2)\log_2 p(a_1, a_2)$$

$$- p(a_2, a_1)\log_2 p(a_2, a_1) - p(a_2, a_2)\log_2 p(a_2, a_2)$$

$$p(a_1, a_1)\log_2 p(a_1, a_1) = p(a_1)\cdot p(a_1)\left[\log_2 p(a_1) + \log_2 p(a_1)\right]$$

$$= p(a_1)p(a_1)\log_2 p(a_1) + p(a_1)p(a_1)\log_2 p(a_1)$$

$$p(a_1, a_2)\log_2 p(a_1, a_2) = p(a_1)\cdot p(a_2)\left[\log_2 p(a_1) + \log_2 p(a_2)\right]$$

$$= p(a_1)p(a_2)\log_2 p(a_1) + p(a_1)p(a_2)\log_2 p(a_2)$$

$$\vdots$$