

Huffman Coding

Huffman Coding

Consider coding a set of symbols by using binary strings, i.e. 0s and 1s.

Huffman Coding

Consider coding a set of symbols by using binary strings, i.e. 0s and 1s.

We have a set of symbols $\{s_i\}$
with corresponding probabilities $\{p_i\}$

Huffman Coding

Consider coding a set of symbols by using binary strings, i.e. 0s and 1s.

We have a set of symbols $\{s_i\}$
with corresponding probabilities $\{p_i\}$

We require an efficient code that is
instantaneously decipherable.

Huffman Coding

We require an efficient code that is **instantaneously decipherable**.

Huffman Coding

We require an efficient code that is **instantaneously decipherable**.

For example, suppose symbols s_1 and s_2 are coded as 0 and 00 respectively.

Huffman Coding

We require an efficient code that is **instantaneously decipherable**.

For example, suppose symbols s_1 and s_2 are coded as 0 and 00 respectively.

If we see 00, is this s_2 or $s_1 s_1$?

Huffman Coding

We require an efficient code that is **instantaneously decipherable**.

For example, suppose symbols s_1 and s_2 are coded as 0 and 00 respectively.

If we see 00, is this s_2 or $s_1 s_1$?

We avoid this by insisting that no code should be a precursor of another.

Huffman Coding

We avoid this by insisting that no code should be a precursor of another.

Huffman Coding

We avoid this by insisting that no code should be a precursor of another.

In our example, we could have symbols s_1 and s_2 coded as 0 and 10 respectively.

Huffman Coding

We avoid this by insisting that no code should be a precursor of another.

In our example, we could have symbols s_1 and s_2 coded as 0 and 10 respectively.

We also want our code to be efficient, so more probable symbols have shorter codes.

Huffman Coding

We also want our code to be efficient, so more probable symbols have shorter codes.

Huffman Coding

We also want our code to be efficient, so more probable symbols have shorter codes.

The length of the code is measured by the average code length per symbol:

Huffman Coding

We also want our code to be efficient, so more probable symbols have shorter codes.

The length of the code is measured by the average code length per symbol:

$$L = \sum_i l_i p_i$$

where l_i is the the code length of symbol s_i .

The Huffman Algorithm

1. Combine the two least probable symbols as a single composite symbol, with probability equal to the sum of the individual probabilities.

s_1	0.4
s_2	0.3
s_3	0.1
s_4	0.1
s_5	0.06
s_6	0.04

s_1 0.4

s_2 0.3

s_3 0.1

s_4 0.1

s_5 0.06

s_6 0.04

s_1	0.4	
s_2	0.3	
s_3	0.1	
s_4	0.1	
s_5	0.06	$\longrightarrow$ 0.1
s_6	0.04	$\uparrow$

The Huffman Algorithm

1. Combine the two least probable symbols as a single composite symbol, with probability equal to the sum of the individual probabilities.
2. Continue until we have just **two** symbols and arbitrarily code one as 0 and the other as 1.

s_1	0.4	
s_2	0.3	
s_3	0.1	
s_4	0.1	
s_5	0.06	$\longrightarrow$ 0.1
s_6	0.04	$\uparrow$

s_1	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1
s_4	0.1	$\longrightarrow$	0.1
s_5	0.06	$\longrightarrow$	0.1
s_6	0.04	$\uparrow$	

s_1	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1
s_4	0.1	$\longrightarrow$	0.1
s_5	0.06	$\longrightarrow$	0.1
s_6	0.04	$\uparrow$	

s_1	0.4	$\longrightarrow$	0.4	
s_2	0.3	$\longrightarrow$	0.3	
s_3	0.1	$\longrightarrow$	0.1	
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$ 0.2
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$
s_6	0.04	$\uparrow$		

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$	
s_6	0.04	$\uparrow$			

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$	
s_6	0.04	$\uparrow$			

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1	$\longrightarrow$ 0.3
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2	$\uparrow$
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$		
s_6	0.04	$\uparrow$				

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.3
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2	$\uparrow$	
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$			
s_6	0.04	$\uparrow$					

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.3
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2	$\uparrow$	
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$			
s_6	0.04	$\uparrow$					

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$ 0.6
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.3	$\uparrow$
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2	$\uparrow$		
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$				
s_6	0.04	$\uparrow$						

s_1	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4	$\longrightarrow$	0.4
s_2	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.3	$\longrightarrow$	0.6
s_3	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.3	$\uparrow$	
s_4	0.1	$\longrightarrow$	0.1	$\longrightarrow$	0.2	$\uparrow$			
s_5	0.06	$\longrightarrow$	0.1	$\uparrow$					
s_6	0.04	$\uparrow$							

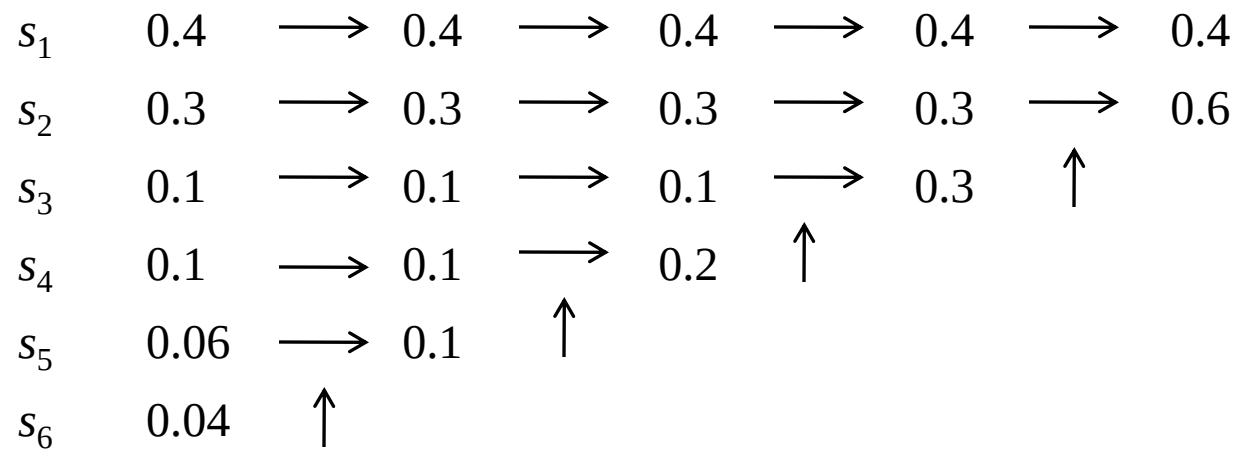
The Huffman Algorithm

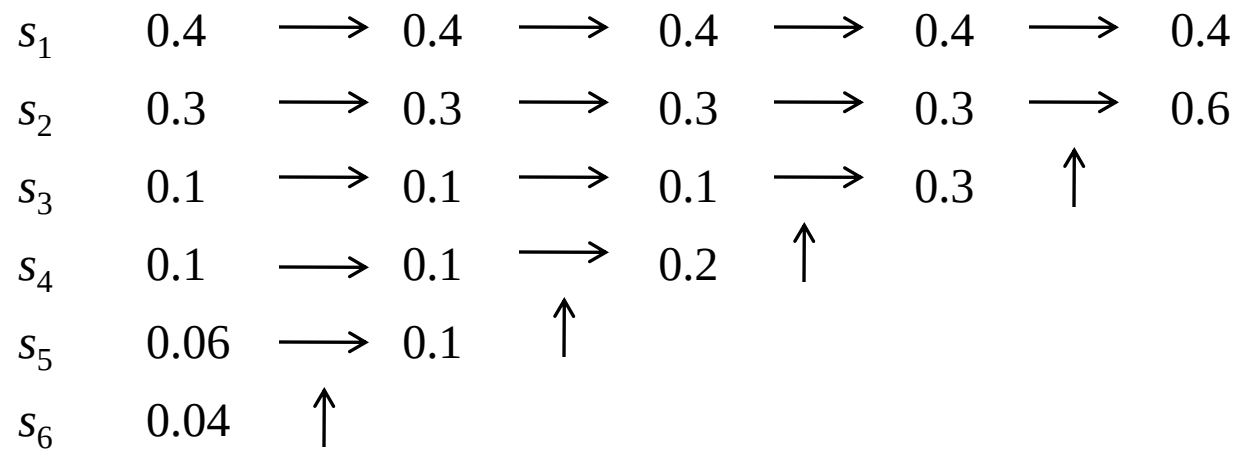
1. Combine the two least probable symbols as a single composite symbol, with probability equal to the sum of the individual probabilities.
2. Continue until we have just two symbols and arbitrarily code one a 0 and the other as 1.

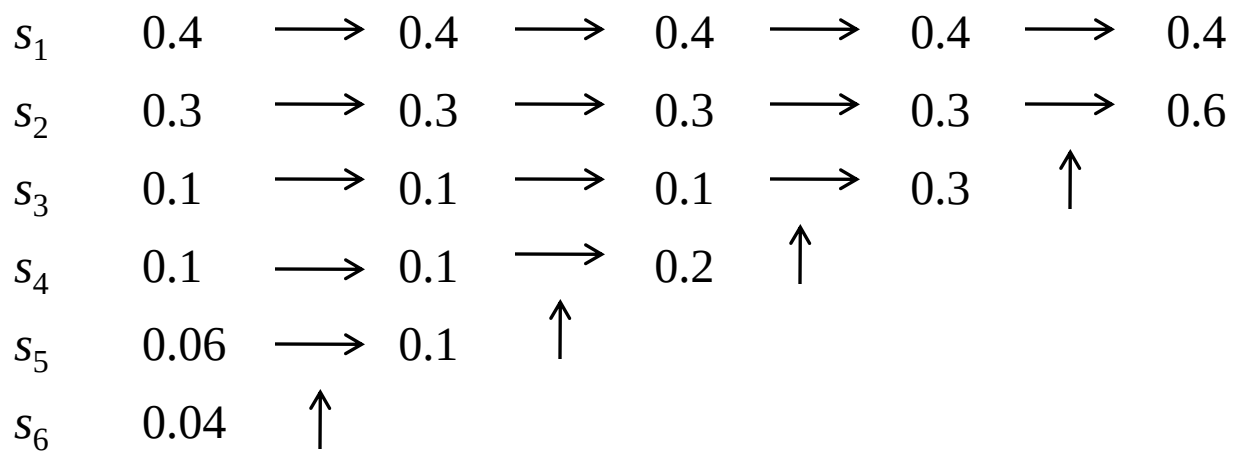
The Huffman Algorithm

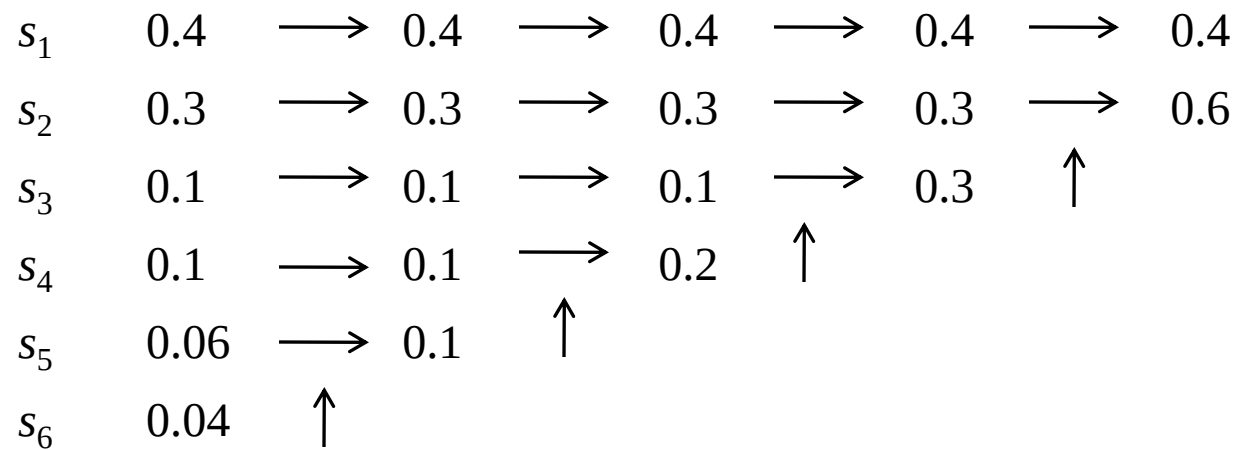
3. Trace the process back so if symbol s comes from just one symbol t then the code for t is the same as for s .

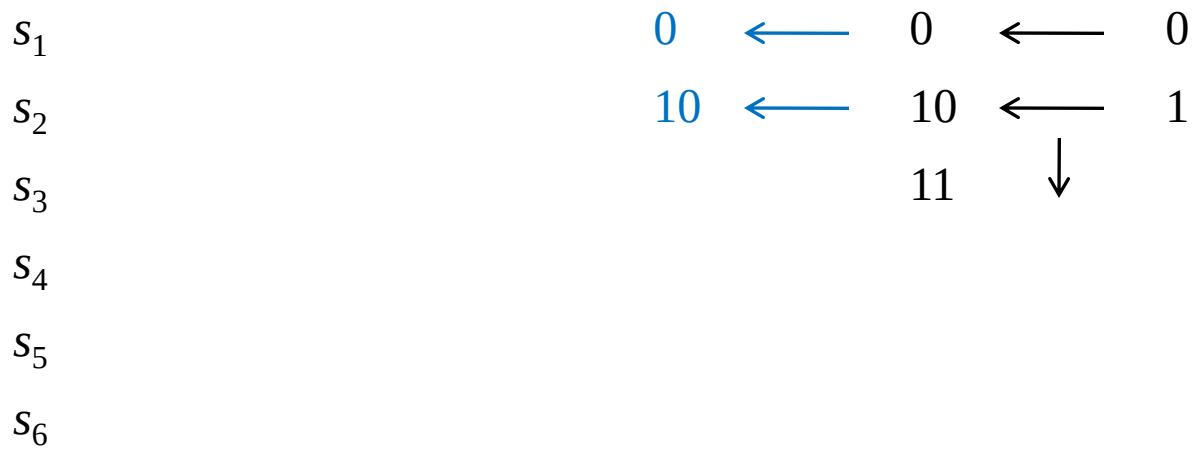
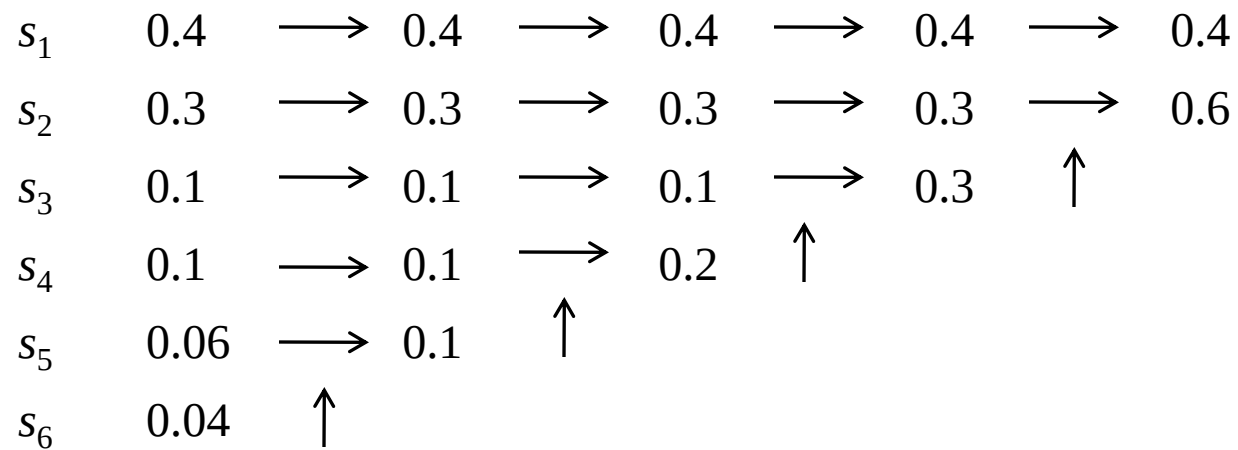
But if s comes from two symbols t and u , then arbitrarily add 0 at the **end** of the code for t and 1 at the **end** of the code for u .





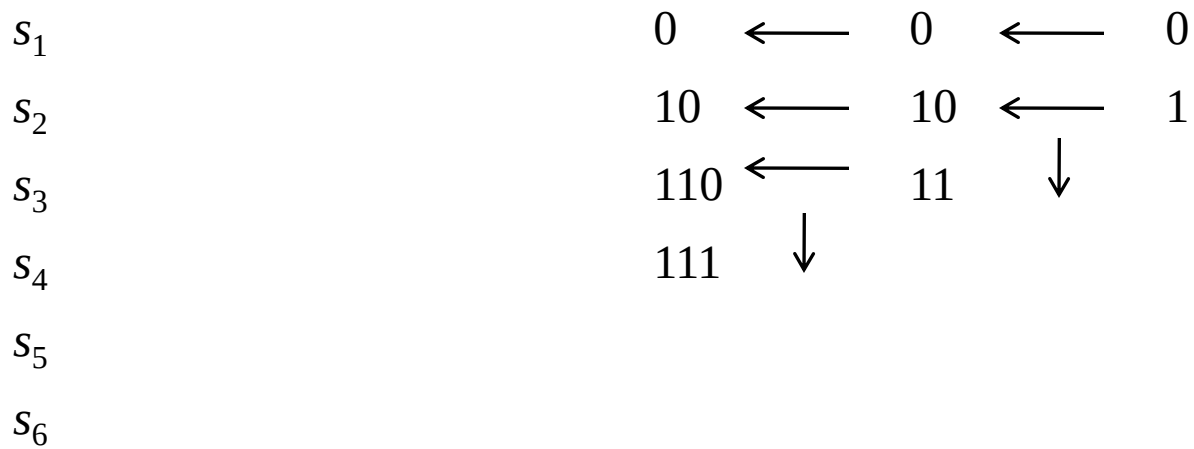
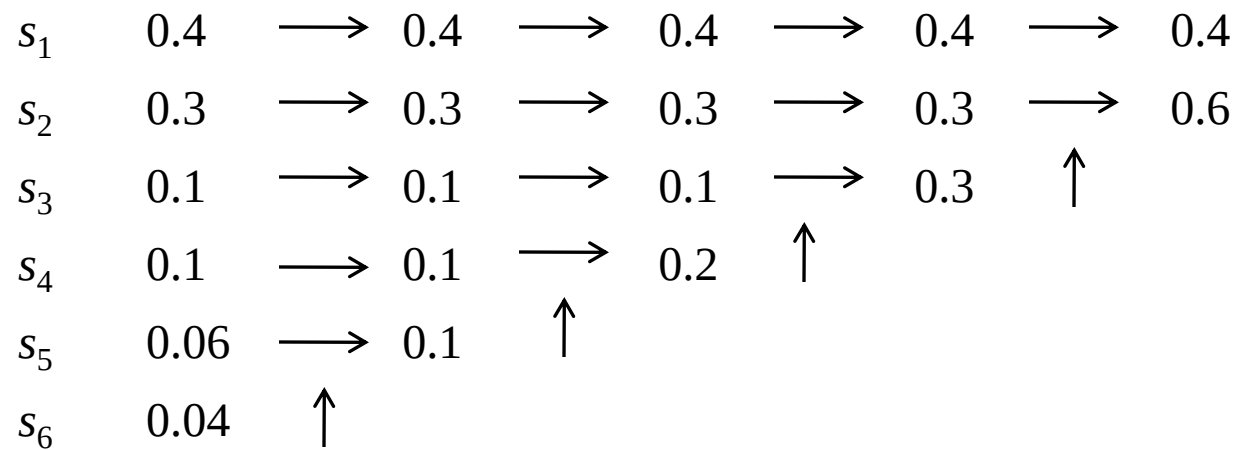


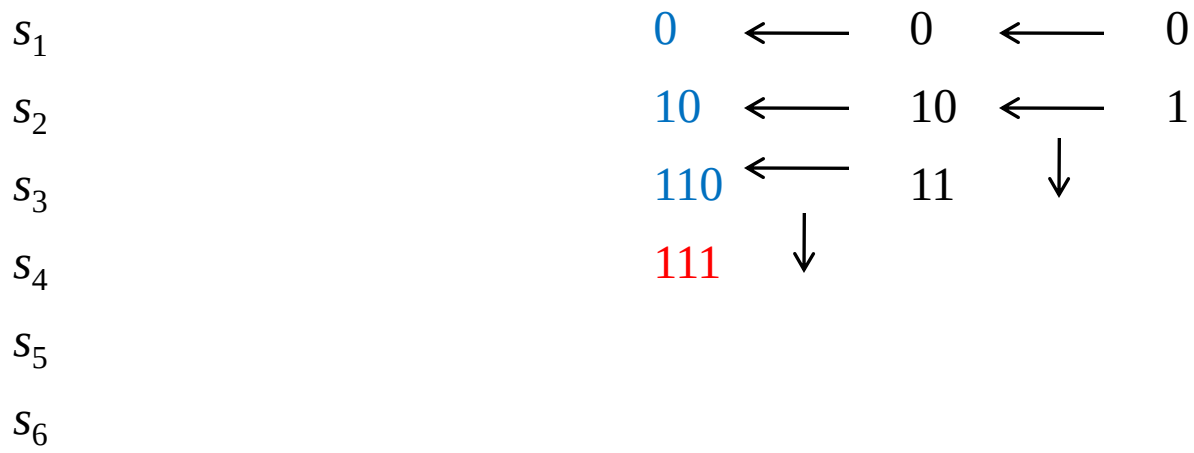
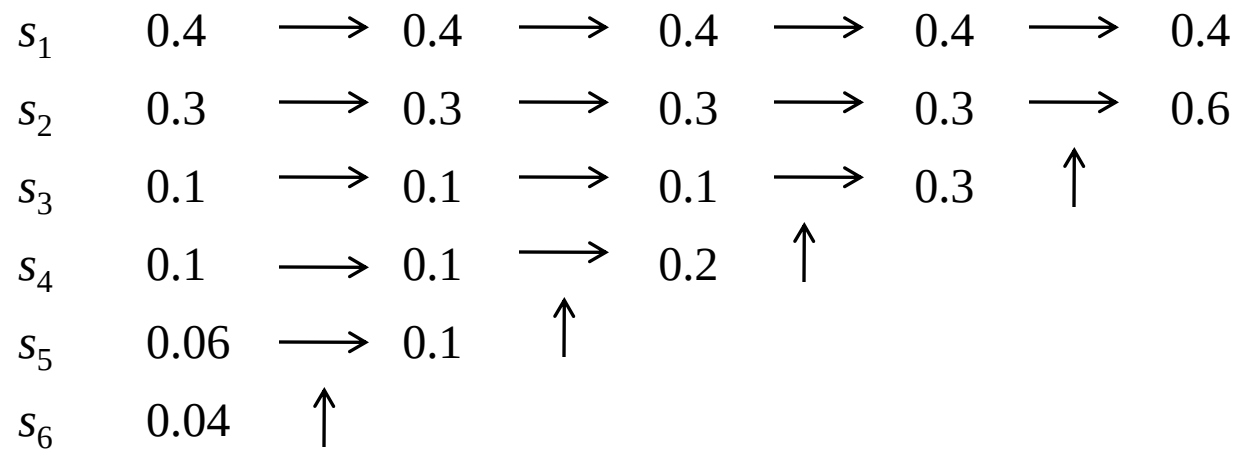




s_1	0.4	→	0.4	→	0.4	→	0.4	→	0.4
s_2	0.3	→	0.3	→	0.3	→	0.3	→	0.6
s_3	0.1	→	0.1	→	0.1	→	0.3	↑	
s_4	0.1	→	0.1	→	0.2	↑			
s_5	0.06	→	0.1	↑					
s_6	0.04	↑							

s_1	0	←	0	←	0
s_2	10	←	10	←	1
s_3	110	←	11	↓	
s_4	111	↓			
s_5					
s_6					



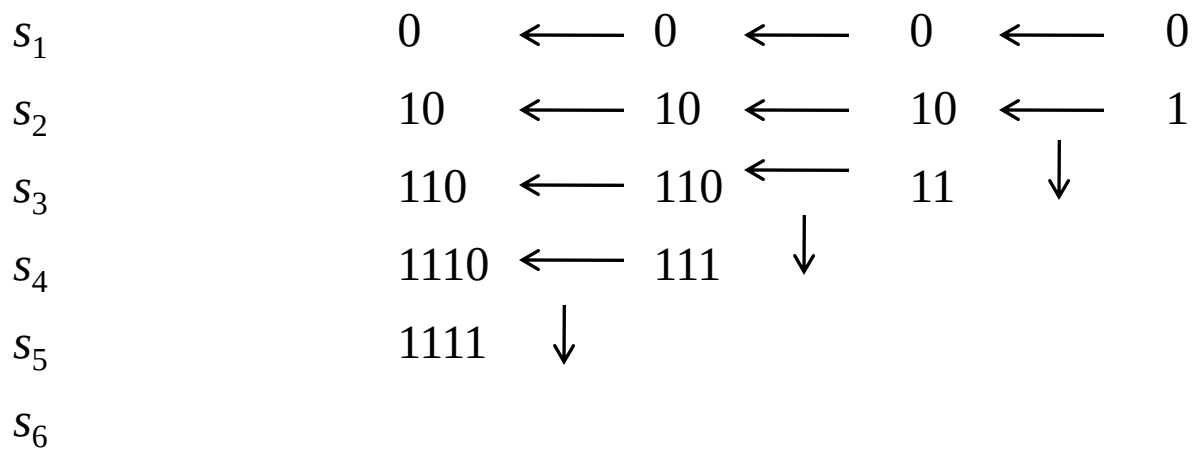
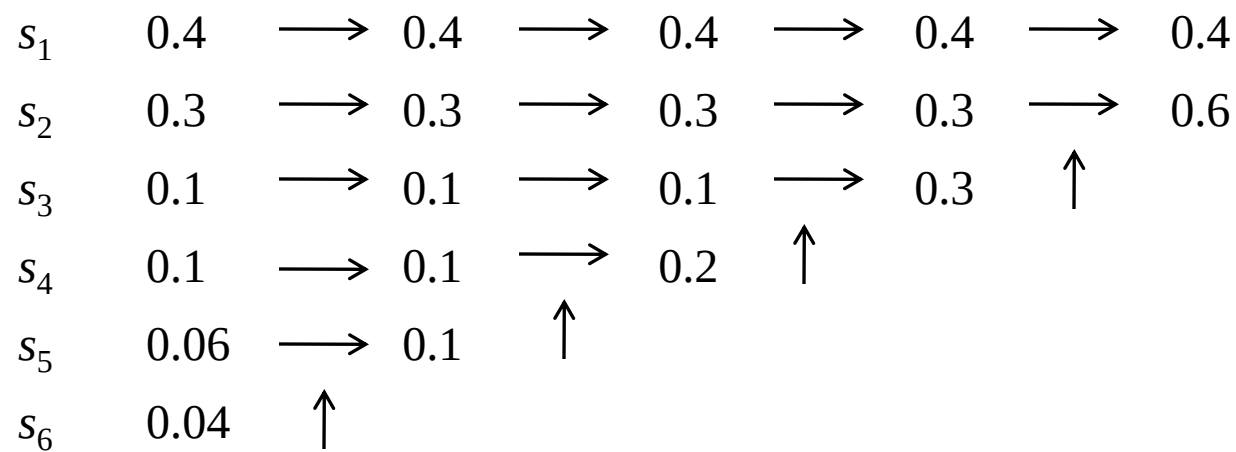


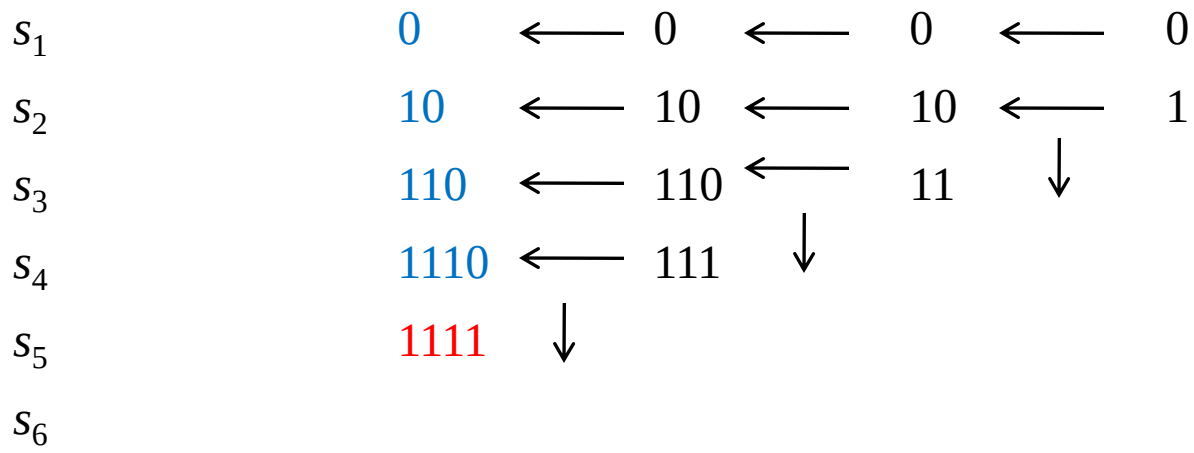
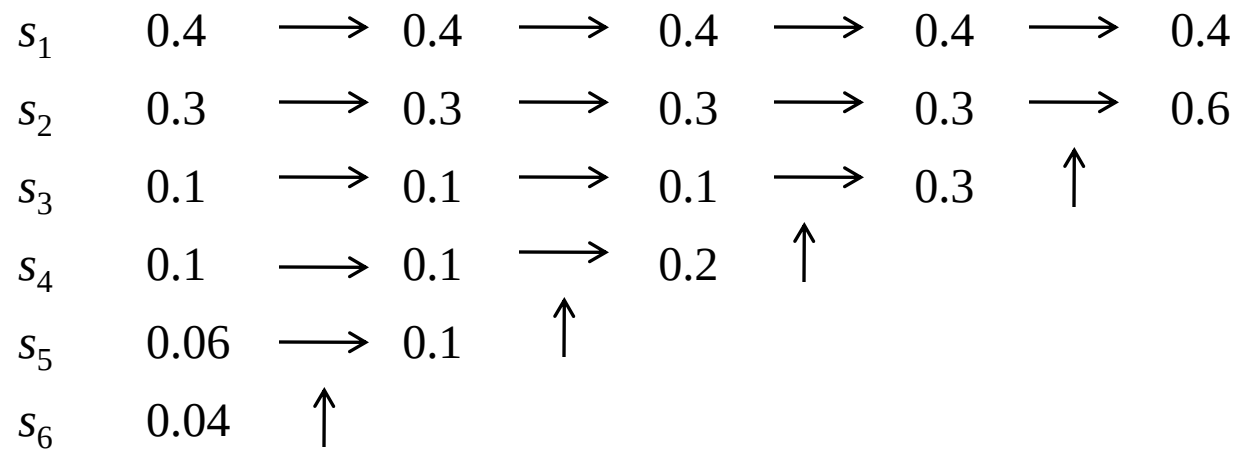
s_1	0.4	→	0.4	→	0.4	→	0.4	→	0.4
s_2	0.3	→	0.3	→	0.3	→	0.3	→	0.6
s_3	0.1	→	0.1	→	0.1	→	0.3	↑	
s_4	0.1	→	0.1	→	0.2	↑			
s_5	0.06	→	0.1	↑					
s_6	0.04	↑							

s_1		0	←	0	←	0	←	0
s_2		10	←	10	←	10	←	1
s_3		110	←	110	←	11	↓	
s_4				111	↓			
s_5								
s_6								

s_1	0.4	→	0.4	→	0.4	→	0.4	→	0.4
s_2	0.3	→	0.3	→	0.3	→	0.3	→	0.6
s_3	0.1	→	0.1	→	0.1	→	0.3	↑	
s_4	0.1	→	0.1	→	0.2	↑			
s_5	0.06	→	0.1	↑					
s_6	0.04	↑							

s_1		0	←	0	←	0	←	0
s_2		10	←	10	←	10	←	1
s_3		110	←	110	←	11	↓	
s_4		1110	←	111	↓			
s_5		1111	↓					
s_6								





s_1	0.4	→	0.4	→	0.4	→	0.4	→	0.4
s_2	0.3	→	0.3	→	0.3	→	0.3	→	0.6
s_3	0.1	→	0.1	→	0.1	→	0.3	↑	
s_4	0.1	→	0.1	→	0.2	↑			
s_5	0.06	→	0.1	↑					
s_6	0.04	↑							

s_1	0	←	0	←	0	←	0	←	0
s_2	10	←	10	←	10	←	10	←	1
s_3	110	←	110	←	110	←	11	↓	
s_4	1110	←	1110	←	111	↓			
s_5	11110	←	1111	↓					
s_6	11111	↓							

s_1	0.4	→	0.4	→	0.4	→	0.4	→	0.4
s_2	0.3	→	0.3	→	0.3	→	0.3	→	0.6
s_3	0.1	→	0.1	→	0.1	→	0.3	↑	
s_4	0.1	→	0.1	→	0.2	↑			
s_5	0.06	→	0.1	↑					
s_6	0.04	↑							

s_1	0	←	0	←	0	←	0	←	0
s_2	10	←	10	←	10	←	10	←	1
s_3	110	←	110	←	110	←	11	↓	
s_4	1110	←	1110	←	111	↓			
s_5	11110	←	1111	↓					
s_6	11111	↓							

We can now find the average code length per symbol:

We can now find the average code length per symbol:

symbol	code
s_1	0
s_2	10
s_3	110
s_4	1110
s_5	11110
s_6	11111

We can now find the average code length per symbol:

symbol	code	length l_i
s_1	0	1
s_2	10	2
s_3	110	3
s_4	1110	4
s_5	11110	5
s_6	11111	5

We can now find the average code length per symbol:

symbol	code	length l_i	probability p_i
s_1	0	1	0.4
s_2	10	2	0.3
s_3	110	3	0.1
s_4	1110	4	0.1
s_5	11110	5	0.06
s_6	11111	5	0.04

We can now find the average code length per symbol:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.4	0.4
s_2	10	2	0.3	0.6
s_3	110	3	0.1	0.3
s_4	1110	4	0.1	0.4
s_5	11110	5	0.06	0.3
s_6	11111	5	0.04	0.2

We can now find the average code length per symbol:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.4	0.4
s_2	10	2	0.3	0.6
s_3	110	3	0.1	0.3
s_4	1110	4	0.1	0.4
s_5	11110	5	0.06	0.3
s_6	11111	5	0.04	0.2

$$L = \sum l_i p_i = 2.2$$

Now consider the following simpler example:

symbol	code	length l_i	probability p_i
s_1	0	1	0.9
s_2	1	1	0.1

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

To **improve** this, we can use the second extension, where **pairs** of symbols are used:

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

To **improve** this, we can use the second extension, where **pairs** of symbols are used:

symbol	probability
--------	-------------

$s_1 s_1$

$s_1 s_2$

$s_2 s_1$

$s_2 s_2$

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

To **improve** this, we can use the second extension, where **pairs** of symbols are used:

symbol	probability
$s_1 s_1$	$0.9 \times 0.9 = 0.81$
$s_1 s_2$	
$s_2 s_1$	
$s_2 s_2$	

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

To **improve** this, we can use the second extension, where **pairs** of symbols are used:

symbol	probability
$s_1 s_1$	$0.9 \times 0.9 = 0.81$
$s_1 s_2$	$0.9 \times 0.1 = 0.09$
$s_2 s_1$	$0.1 \times 0.9 = 0.09$
$s_2 s_2$	$0.1 \times 0.1 = 0.01$

Now consider the following simpler example:

symbol	code	length l_i	probability p_i	$l_i p_i$
s_1	0	1	0.9	0.9
s_2	1	1	0.1	0.1

$$L = \sum l_i p_i = 1$$

To **improve** this, we can use the second extension, where **pairs** of symbols are used:

symbol	probability
$s_1 s_1$	$0.9 \times 0.9 = 0.81$
$s_1 s_2$	$0.9 \times 0.1 = 0.09$
$s_2 s_1$	$0.1 \times 0.9 = 0.09$
$s_2 s_2$	$0.1 \times 0.1 = 0.01$

This assumes symbols occur independently, so
 $P(s_i s_j) = P(s_i) \times P(s_j)$

$$s_1s_1 \quad 0.81$$

$$s_1s_2 \quad 0.09$$

$$s_2s_1 \quad 0.09$$

$$s_2s_2 \quad 0.01$$

s_1s_1	0.81	$\longrightarrow$	0.81
s_1s_2	0.09	$\longrightarrow$	0.09
s_2s_1	0.09	$\longrightarrow$	0.1
s_2s_2	0.01	$\uparrow$	

s_1s_1	0.81	$\longrightarrow$	0.81	$\longrightarrow$	0.81
s_1s_2	0.09	$\longrightarrow$	0.09	$\longrightarrow$	0.19
s_2s_1	0.09	$\longrightarrow$	0.1	$\uparrow$	
s_2s_2	0.01	$\uparrow$			

s_1s_1	0.81	$\longrightarrow$	0.81	$\longrightarrow$	0.81
s_1s_2	0.09	$\longrightarrow$	0.09	$\longrightarrow$	0.19
s_2s_1	0.09	$\longrightarrow$	0.1	$\uparrow$	
s_2s_2	0.01	$\uparrow$			

s_1s_1	0
s_1s_2	1
s_2s_1	
s_2s_2	

s_1s_1	0.81	$\longrightarrow$	0.81	$\longrightarrow$	0.81
s_1s_2	0.09	$\longrightarrow$	0.09	$\longrightarrow$	0.19
s_2s_1	0.09	$\longrightarrow$	0.1	$\uparrow$	
s_2s_2	0.01	$\uparrow$			

s_1s_1	0	$\longleftarrow$	0
s_1s_2	10	$\longleftarrow$	1
s_2s_1	11	$\downarrow$	
s_2s_2			

s_1s_1	0.81	$\longrightarrow$	0.81	$\longrightarrow$	0.81
s_1s_2	0.09	$\longrightarrow$	0.09	$\longrightarrow$	0.19
s_2s_1	0.09	$\longrightarrow$	0.1	$\uparrow$	
s_2s_2	0.01	$\uparrow$			

s_1s_1	0	$\longleftarrow$	0	$\longleftarrow$	0
s_1s_2	10	$\longleftarrow$	10	$\longleftarrow$	1
s_2s_1	110	$\longleftarrow$	11	$\downarrow$	
s_2s_2	111	$\downarrow$			

symbol	code	length l_i	probability p_i
s_1s_1	0	1	0.81
s_1s_2	10	2	0.09
s_2s_1	110	3	0.09
s_2s_2	1110	3	0.01

symbol	code	length l_i	probability p_i	$l_i p_i$
$s_1 s_1$	0	1	0.81	0.81
$s_1 s_2$	10	2	0.09	0.18
$s_2 s_1$	110	3	0.09	0.27
$s_2 s_2$	1110	3	0.01	0.03

symbol	code	length l_i	probability p_i	$l_i p_i$
$s_1 s_1$	0	1	0.81	0.81
$s_1 s_2$	10	2	0.09	0.18
$s_2 s_1$	110	3	0.09	0.27
$s_2 s_2$	1110	3	0.01	0.03

$$L = \sum l_i p_i = 1.29$$

$$L = \sum l_i p_i = 1.29$$

So average code length per symbol is now $\frac{1.29}{2} = 0.645$

$$L = \sum l_i p_i = 1.29$$

So average code length per symbol is now $\frac{1.29}{2} = 0.645$

... whereas previously it was 1 ...

$$L = \sum l_i p_i = 1.29$$

So average code length per symbol is now $\frac{1.29}{2} = 0.645$

... whereas previously it was 1 ...

... so the efficiency of the code **has** improved.

$$L = \sum l_i p_i = 1.29$$

So average code length per symbol is now $\frac{1.29}{2} = 0.645$

... whereas previously it was 1 ...

... so the efficiency of the code **has** improved.

We can extend this to triplets, quadruplets, etc.

There are some theorems that measure this process ...

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

Using this, and the fact that $\log_2(x) = \frac{\ln x}{\ln 2}$

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

Using this, and the fact that $\log_2(x) = \frac{\ln x}{\ln 2}$

... we can calculate the entropy of the source for both our examples so far ...

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

s_1 0.4

s_2 0.3

s_3 0.1

s_4 0.1

s_5 0.06

s_6 0.04

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

$$H(S) = - \sum_i \frac{p_i \ln(p_i)}{\ln 2}$$

s_1	0.4
s_2	0.3
s_3	0.1
s_4	0.1
s_5	0.06
s_6	0.04

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

s_1	0.4	$H(S) = - \sum_i \frac{p_i \ln(p_i)}{\ln 2}$ $= - \left(\begin{aligned} &\frac{0.4 \ln(0.4)}{\ln 2} + \frac{0.3 \ln(0.3)}{\ln 2} \\ &+ \frac{0.1 \ln(0.1)}{\ln 2} + \frac{0.1 \ln(0.1)}{\ln 2} \\ &+ \frac{0.06 \ln(0.06)}{\ln 2} + \frac{0.04 \ln(0.04)}{\ln 2} \end{aligned} \right)$
s_2	0.3	
s_3	0.1	
s_4	0.1	
s_5	0.06	
s_6	0.04	

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

s_1 0.4 s_2 0.3 s_3 0.1 s_4 0.1 s_5 0.06 s_6 0.04	$H(S) = - \sum_i \frac{p_i \ln(p_i)}{\ln 2}$ $= - \left(\begin{aligned} &\frac{0.4 \ln(0.4)}{\ln 2} + \frac{0.3 \ln(0.3)}{\ln 2} \\ &+ \frac{0.1 \ln(0.1)}{\ln 2} + \frac{0.1 \ln(0.1)}{\ln 2} \\ &+ \frac{0.06 \ln(0.06)}{\ln 2} + \frac{0.04 \ln(0.04)}{\ln 2} \end{aligned} \right)$ $= 2.14353$
--	--

The entropy of the source is defined as:

$$H(S) = - \sum_i p_i \log_2(p_i)$$

$$s_1 \quad 0.9$$

$$s_2 \quad 0.1$$

$$\begin{aligned} H(S) &= - \sum_i \frac{p_i \ln(p_i)}{\ln 2} \\ &= - \left(\frac{0.9 \ln(0.9)}{\ln 2} + \frac{0.1 \ln(0.1)}{\ln 2} \right) \\ &= 0.4689956 \end{aligned}$$

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

In our first example, $L = 2.2$

while $H(S) = 2.14353$

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

In our first example, $L = 2.2$

while $H(S) = 2.14353$

$$2.14353 \leq 2.2 \leq 3.14353$$

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

In our first example, $L = 2.2$

while $H(S) = 2.14353$

$$2.14353 \leq 2.2 \leq 3.14353$$

In our second example, $L = 1$

while $H(S) = 0.4689956$

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

In our first example, $L = 2.2$

while $H(S) = 2.14353$

$$2.14353 \leq 2.2 \leq 3.14353$$

In our second example, $L = 1$

while $H(S) = 0.4689956$

$$0.4689956 \leq 1 \leq 1.4689956$$

The important theorems are:

1.
$$H(S) \leq L \leq H(S) + 1$$

The corresponding theorem for the n -extensions is

2.
$$H(S) \leq \frac{L_n}{n} \leq H(S) + \frac{1}{n}$$

The important theorems are:

1.
$$H(S) \leq L \leq H(S) + 1$$

The corresponding theorem for the n -extensions is

2.
$$H(S) \leq \frac{L_n}{n} \leq H(S) + \frac{1}{n}$$

So $H(S)$ is the best average code length can be attained, and...

The important theorems are:

1. $H(S) \leq L \leq H(S) + 1$

The corresponding theorem for the n -extensions is

2. $H(S) \leq \frac{L_n}{n} \leq H(S) + \frac{1}{n}$

So $H(S)$ is the best average code length can be attained, and...

$$\lim_{n \rightarrow \infty} \frac{L_n}{n} = H(S)$$