

# 信息论与编码

Information Theory and Coding

西南交通大学

电子信息工程专业

2020

# 第4章 无失真信源编码

## 4-1 引言

R.W.Hamming

“The **coding theory** leads to **information theory** and the **information theory** provides the **bounds** on what can be done by suitable encoding of the information.”



# 第4章 无失真信源编码

## 4-1 引言

### International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	• — —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —		
L	• — • •		
M	— —		
N	— •		
O	— — —		
P	• — — •		
Q	— — • —		
R	• — •		
S	• • •		
T	—		

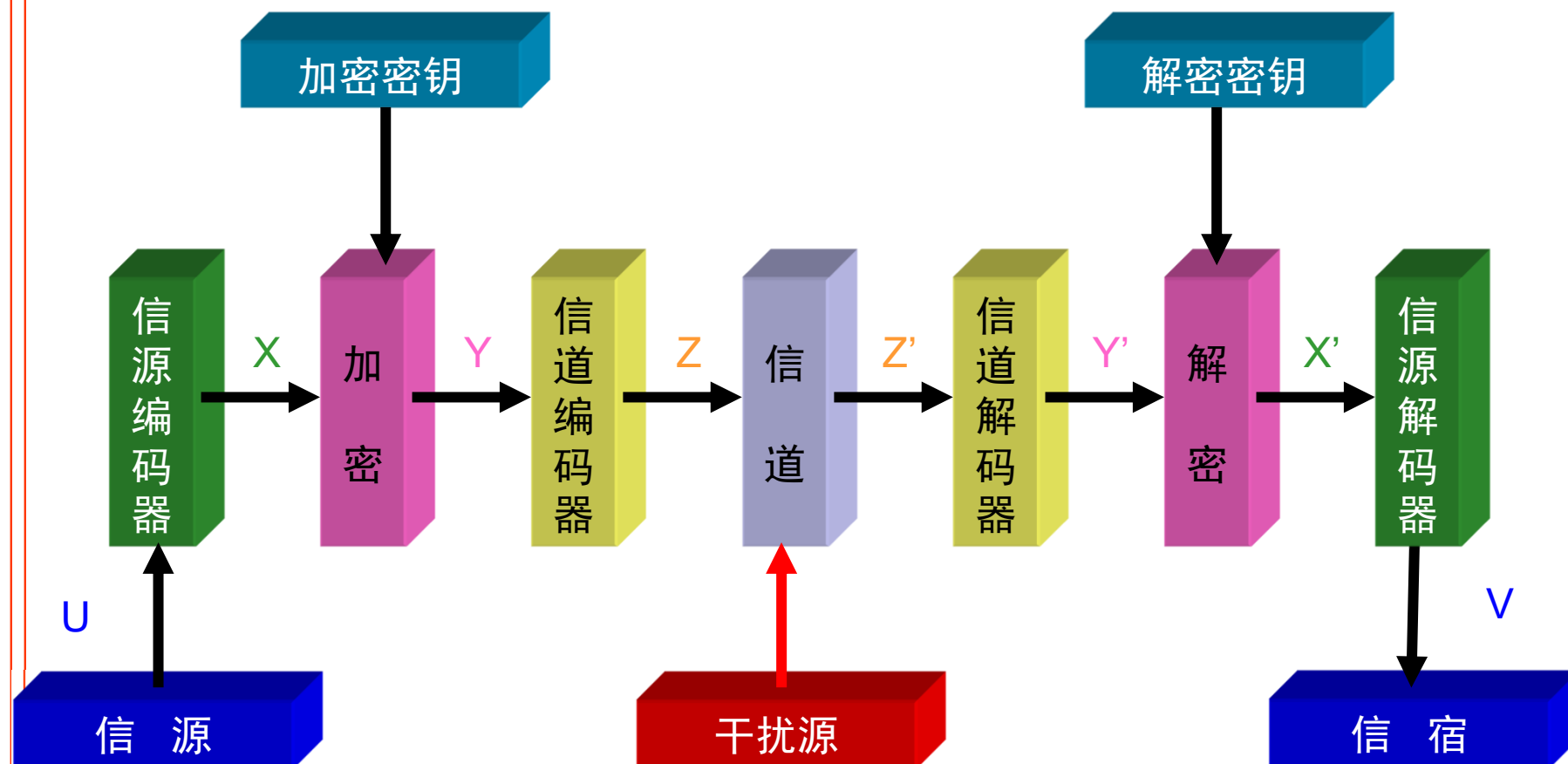
U	• • —
V	• • • —
W	• — —
X	— • • —
Y	— • — —
Z	— — • •

1	• — — — —
2	• • — — —
3	• • • — —
4	• • • • —
5	• • • • •
6	— • • • •
7	— • • • •
8	— • • • •
9	— — • • • •
0	— — — — •



# 第4章 无失真信源编码

## 4-1 引言



信息传输系统的模型

# 第4章 无失真信源编码

## 4-1 引言

信源编码 { 无失真信源编码 → 第一极限定理  
                  限失真信源编码 → 第三极限定理  
信道编码 — — — — — → 第二极限定理  
加密编码

# 第4章 无失真信源编码

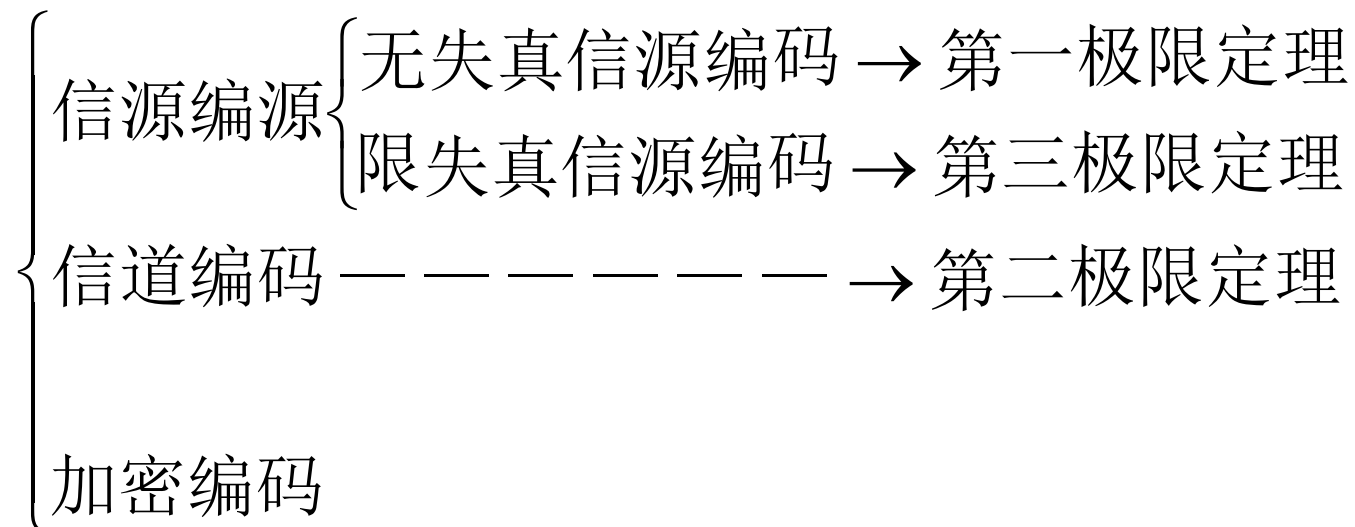
## 4-1 引言

$\left\{ \begin{array}{l} \text{信源编码} \left\{ \begin{array}{l} \text{无失真信源编码} \rightarrow \text{第一极限定理} \\ \text{限失真信源编码} \rightarrow \text{第三极限定理} \end{array} \right. \\ \text{信道编码} \text{ — — — — — } \rightarrow \text{第二极限定理} \\ \text{加密编码} \end{array} \right.$

信源编码的意义：

# 第4章 无失真信源编码

## 4-1 引言



信源编码的意义：

【1】适于信道的传输和信息的信息处理

【2】减少信源输出不必要的冗余度

$$\text{熵率: } \eta = \frac{H_{\infty}}{\log r} = \frac{H_{\infty}}{H_0}$$

$$\text{冗余度 } \gamma = 1 - \eta = \frac{H_0 - H_{\infty}}{H_0}$$

# 第4章 无失真信源编码

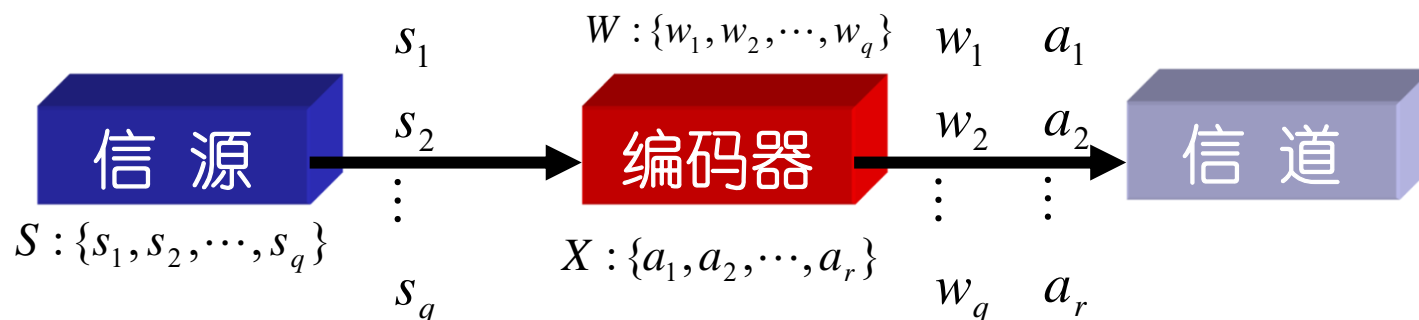
## 4-1 引言

信源编码的类型：

- 【1】 **无失真信源编码**：将信道所能传递的符号集作为码符号集，对信源所能发出的每一种符号进行一一对应的编码
  - 适用于离散信源的形式
  - 编码是可逆编码(即无失真的从代码恢复出信源符号)
  
- 【2】 **限失真信源编码**：对于连续信源，编成代码以后就无法无失真地恢复原来的连续值，因此只能采用限失真信源编码



## 4-2 编码的定义



**问题1:** 信源 $S$ 发出 $q$ 种不同的信源符号 $s_i$ 与信道所能传递的 $r$ 种码符号 $a_i$ 不一致

**问题2:** 信源 $S$ 发出 $q$ 种不同的信源符号 $s_i$ 以及由 $s_i$ 的序列代表的每一条消息实现无失真的传递

**信源  
编码**

用码符号集 $X : \{a_1, a_2, \dots, a_r\}$ 中的码符号 $a_i$  ( $i = 1, 2, \dots, r$ ) 对信源 $S$ 的每一种不同符号 $s_i$  ( $i = 1, 2, \dots, q$ ) 进行编码, 使信源 $S$ 适合信道的传输, 这种变换过程称为**信源编码**

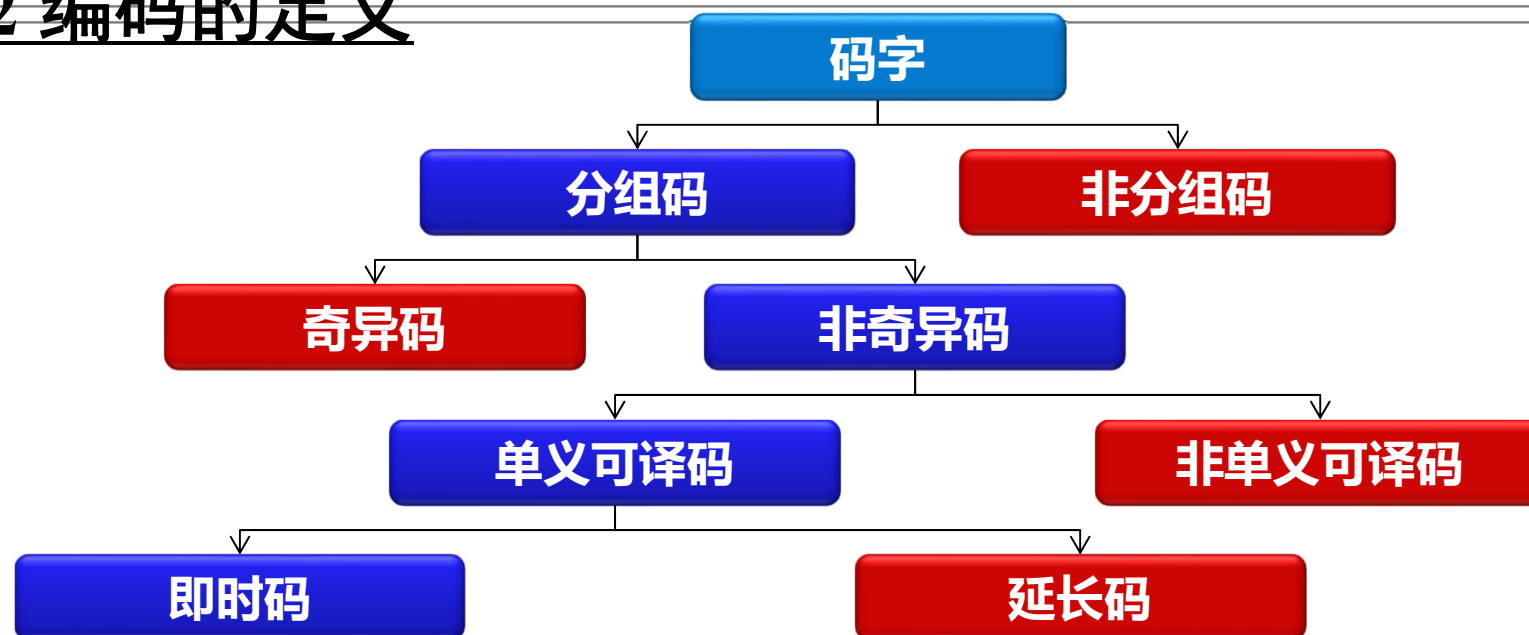
## 4-2 编码的定义

码字与信息率的关系：

- 有时消息太多，不可能或者没必要给每个消息都分配一个码字；
- 给多少消息分配码字可以做到几乎无失真译码？
- 传送码字需要一定的信息率，码字越多，所需的信息率越大。编多少码字的问题可以转化为对信息率大小的问题；
- 信息率越小越好，最小能小到多少才能做到无失真译码呢？

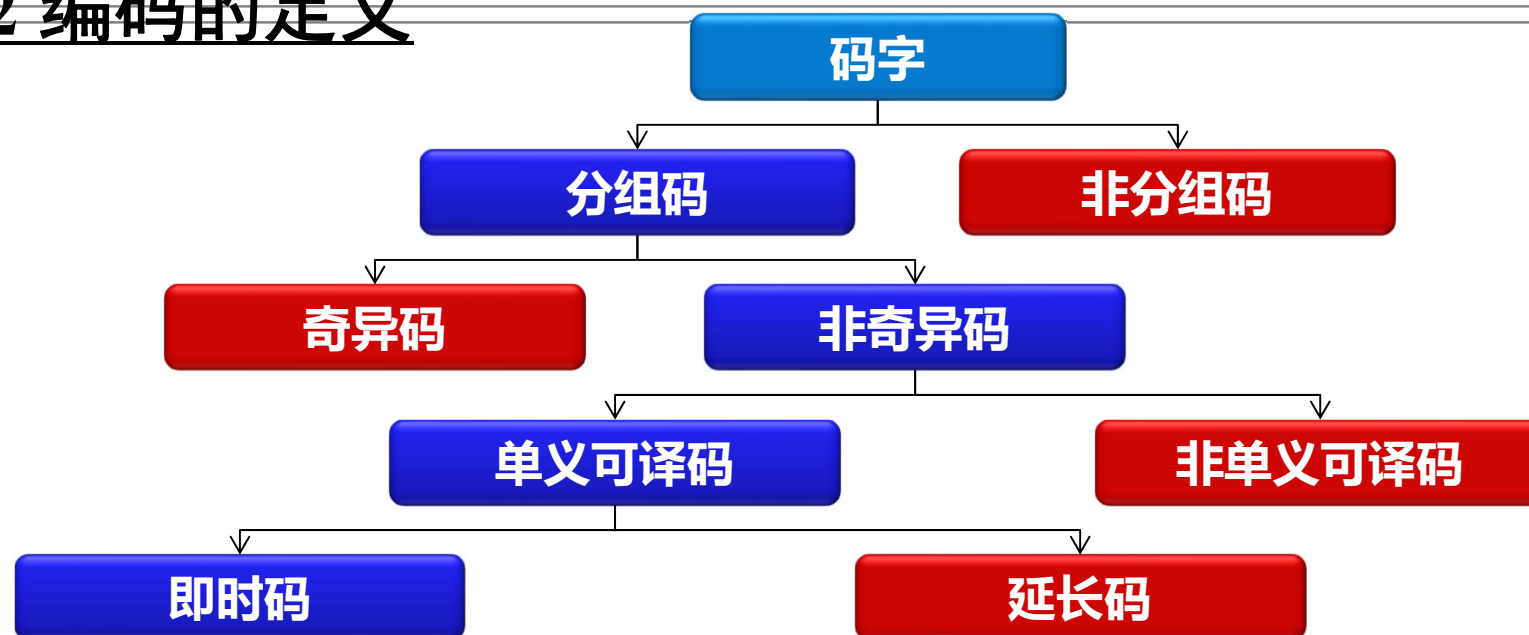
这些问题就是信源编码定理要研究的问题。

## 4-2 编码的定义



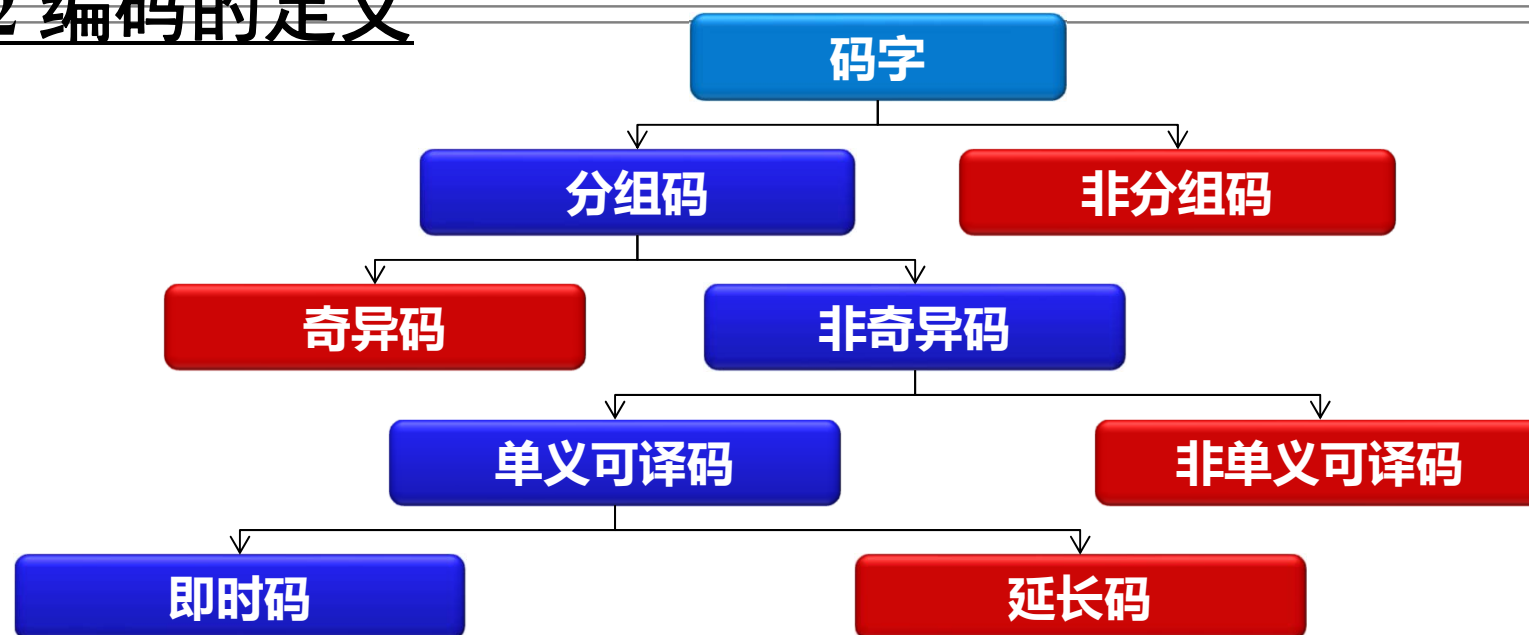
码字：用码符号集 $X : \{a_1, a_2, \dots, a_r\}$ 中的码符号 $a_i (i = 1, 2, \dots, r)$ 对信源 $S$ 的每一种不同符号 $s_i (i = 1, 2, \dots, q)$ 进行编码，构成由码符号 $a_i (i = 1, 2, \dots, r)$ 组成的序列，即码字 $w_i (i = 1, 2, \dots, q)$ 。

## 4-2 编码的定义



**定义：** 将信源消息分成若干组，得到符号序列  $S_i : s_{i1}s_{i2} \cdots s_{iN}$  序列中每个符号取自符号集  $S: \{s_1, s_2, \cdots, s_q\}$ ，而每个符号序列  $S_i$  依照固定的码表映射成一个码字  $w_i$ ，这样的码称为**分组码**

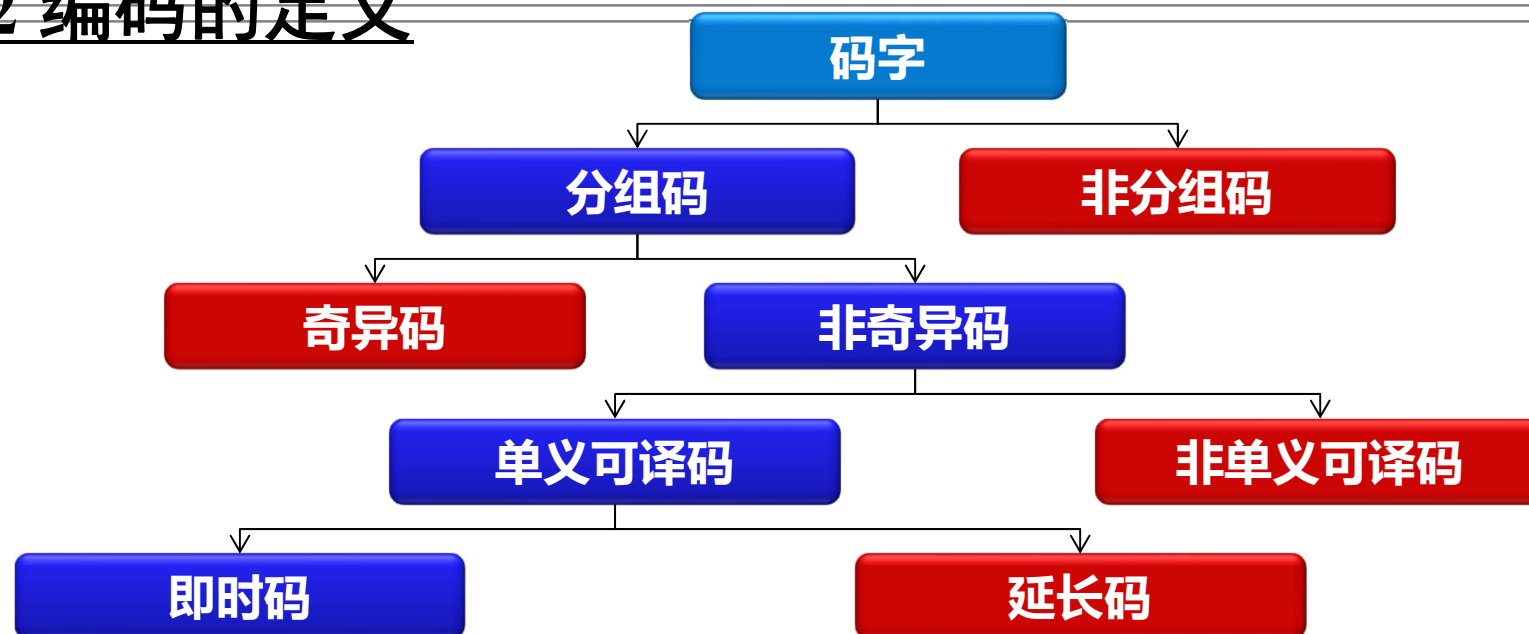
## 4-2 编码的定义



**定义：**若一组码中所有码字都不相同，即所有信源符号映射到不同的码符号序列，则称为**非奇异码**，否则称为**奇异码**

信源符号	码1	码2
S1	0	0
S2	11	10
S3	00	00
S4	11	01

## 4-2 编码的定义

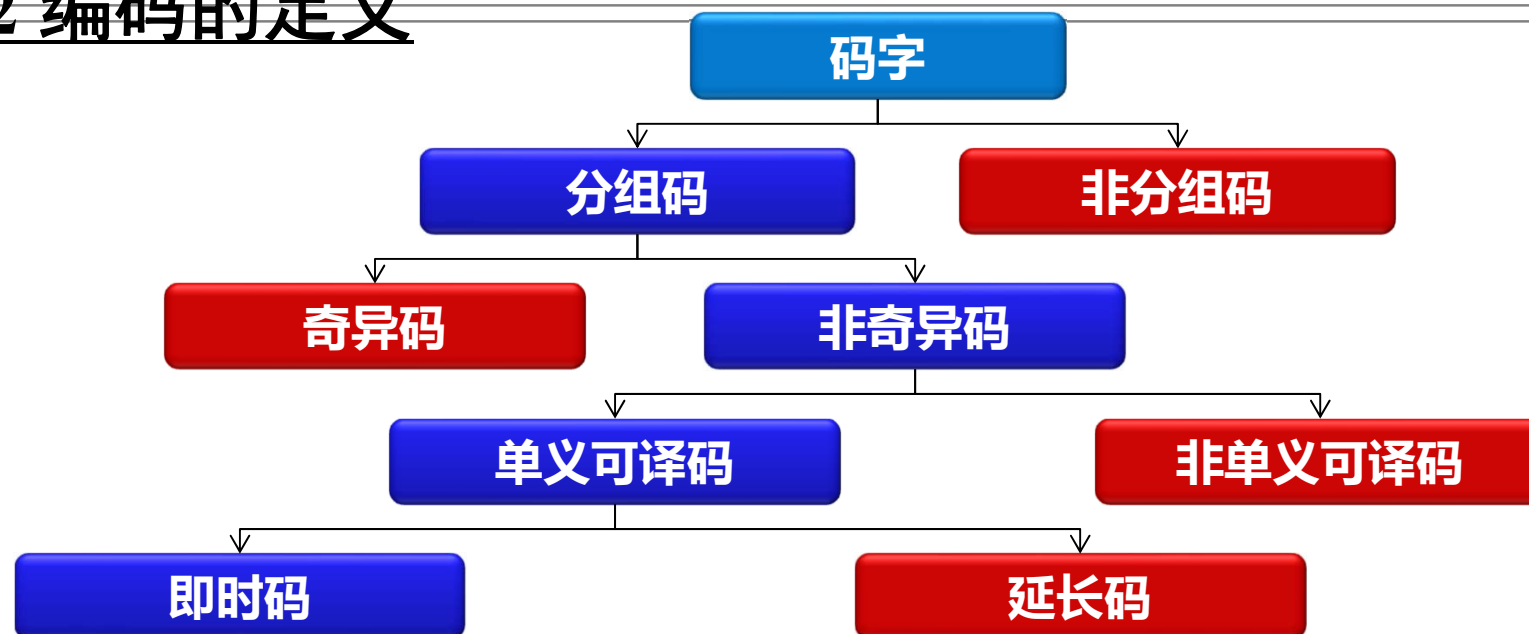


**定义：**若码的任意一串有限长的码符号序列只能被唯一地译成所对应的信源符号序列，称为**唯一可译码**或**单义可译码**

**定义：**若一组码字中，所有码字的码长都相同，称为**等长码**

信源符号	码2	码3
S1	0	00
S2	10	01
S3	00	10
S4	01	11

## 4-2 编码的定义



**定义：**无需参考后续码符号就可以即时作出译码判断的码，称为**即时码**

**定理：**一个唯一可译码成为即时码的充要条件是其中任何一个码字都不是其他码字的前缀。

信源符号	码4	码5
S1	1	1
S2	10	01
S3	100	001
S4	1000	0001

## 4-2 编码的定义

**例题：**如果信源 $S$ 由4种不同的符号 $S : \{s_1, s_2, s_3, s_4\}$ 组成，各自的先验概率为 $p(s_1), p(s_2), p(s_3), p(s_4)$ 。采用符号集为 $\{0,1\}$ 的二元信道。编码如下：

$s_i$	$p(s_i)$	码1: W(1)	码2: W(2)	码3: W(3)	码4: W(4)	码5: W(5)
$s_1$	$p(s_1)$	w1=0	w1=0	w1=00	w1=1	w1=1
$s_2$	$p(s_2)$	w2=11	w2=10	w2=01	w2=10	w2=01
$s_3$	$p(s_3)$	w3=00	w3=00	w3=10	w3=100	w3=001
$s_4$	$p(s_4)$	w4=11	w4=01	w4=11	w4=1000	w4=0001



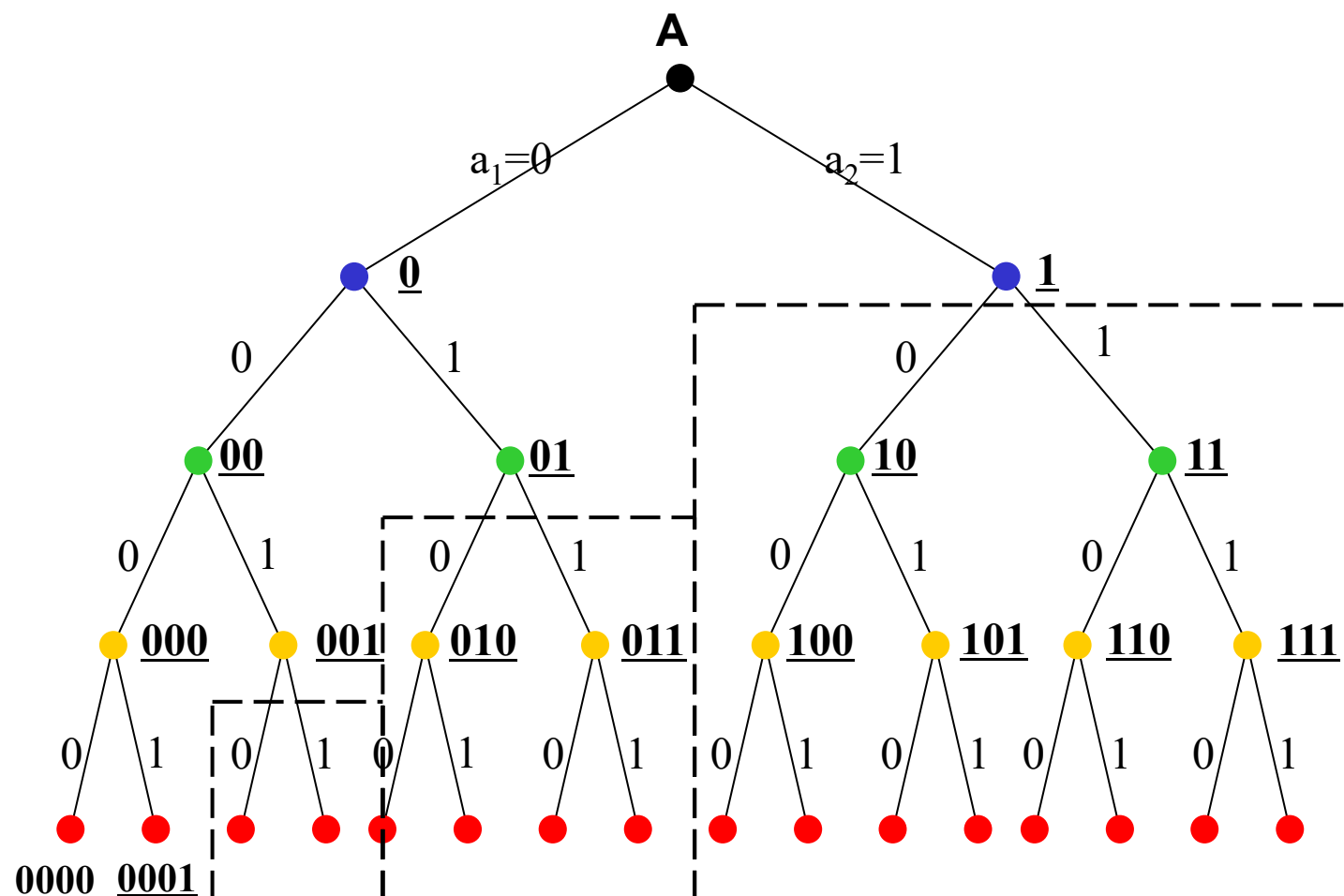
## 4-2 编码的定义

### 利用树图编制即时码：

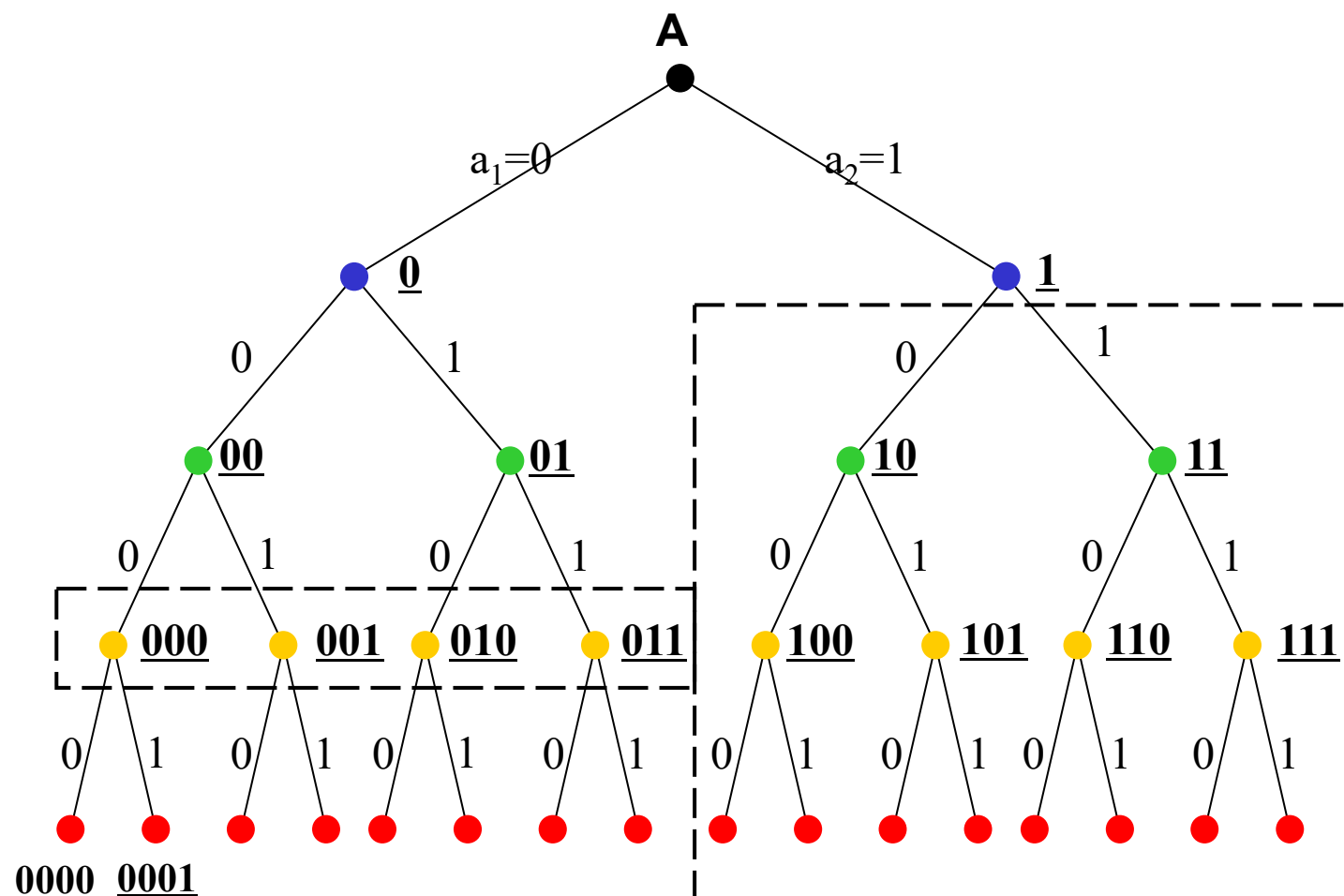
**例题：** 信源 $S$ 由4种不同的符号 $S : \{s_1, s_2, s_3, s_4\}$ 组成，采用码符号集为 $\{0,1\}$ 的二元信道，求与信源符号对应的码字 $w_i$ ，长度为 $n = 1, 2, 3, 4$

下面采用树图法编出单义可译码：

## 4-2 编码的定义



## 4-2 编码的定义



## 4-2 编码的定义

0	00	000	0000	
			0001	
		001	0010	
			0011	
	01	010	0100	
			0101	
		011	0110	
			0111	
1	10	100	1000	
			1001	
		101	1010	
			1011	
	11	110	1100	
			1101	
		111	1110	
			1111	

## 4-2 编码的定义

采用树图法构造单义可译码的步骤：

1. 从树根A出发，伸出  $r$  (码符号数)根树枝，在树枝旁依次标上码符号数 $a_1, a_2, \dots, a_r$ ，树枝的尽头为节点，从树根A出发，第一次延伸树枝所得节点为“一阶节点”，其数目等于 $r$ ；  
  
把“树根A”到一阶节点所有“树枝”上的码符号作为一阶节点各自的码符号序列。
2. 按步骤1方法得到 $r^2$ 个二阶节点及其对应的码符号序列；
3. 树枝延伸到 $n_{\max}$ (最大码长)阶节点终止；
4. 根据画好的码树及编码结构上的要求，挑选合适的“节点”，得到码字。

## 4-2 编码的定义

信源编码的方法：

**【1】定长编码：**码字长度固定，相应的编码定理称为定长信源编码定理，是寻求最小码长 $n$ 的编码方法。

**【2】变长编码：** $n$ 是变值，相应的编码定理称为变长编码定理。

(这里的 $n$ 最小意味着数学期望最小)

## 4-3 定长码及定长编码定理

### 定长编码:

- 1、对于有 $q$ 个信源符号的信源 $S$ 进行定长编码( $l$ 为码字长度),  
则信源 $S$ 存在唯一可译定长码的条件是  $q \leq r^l$
- 2、对于信源的 $N$ 次扩展信源进行定长编码, 则信源 $S$ 存在唯一可译定长码的条件是  $q^N \leq r^l$

其中 $r$ 是码符号集中码元数,  $l$ 是定长码码长。

$$N \log_2 q \leq l \log_2 r \Rightarrow \frac{l}{N} \geq \frac{\log_2 q}{\log_2 r} = \log_r q$$

## 4-3 定长码及定长编码定理

$$\frac{l}{N} \geq \log_r q$$

- 1、当 $r=2$ ,表示对于二元定长唯一可译码, 平均每个原始符号至少需要  $\log_2 q$  个二元符号表示;
- 2、当 $N=1$ , 则有  $l \geq \log_2 q$

**【问题】** 考虑英文电报32字母符号作为信源符号, 利用二元码进行定长编码: 则每个字母符号至少需要5位二元码才可唯一译码。

**【思考】** 考虑到第二章提到英文符号提供的极限熵近似接近 1.4bit/sym, **如何提高信息传递效率?**

**方法1:** \_\_\_\_\_ **方法2:** \_\_\_\_\_



## 4-3 定长码及定长编码定理

【例题】设有离散无记忆信源

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} a & b & c & d & e & f & g & h \\ 1/4 & 1/4 & 1/4 & 3/64 & 1/64 & 1/64 & 1/64 & 1/64 \end{bmatrix}$$

如果对信源符号采用定长二元编码,

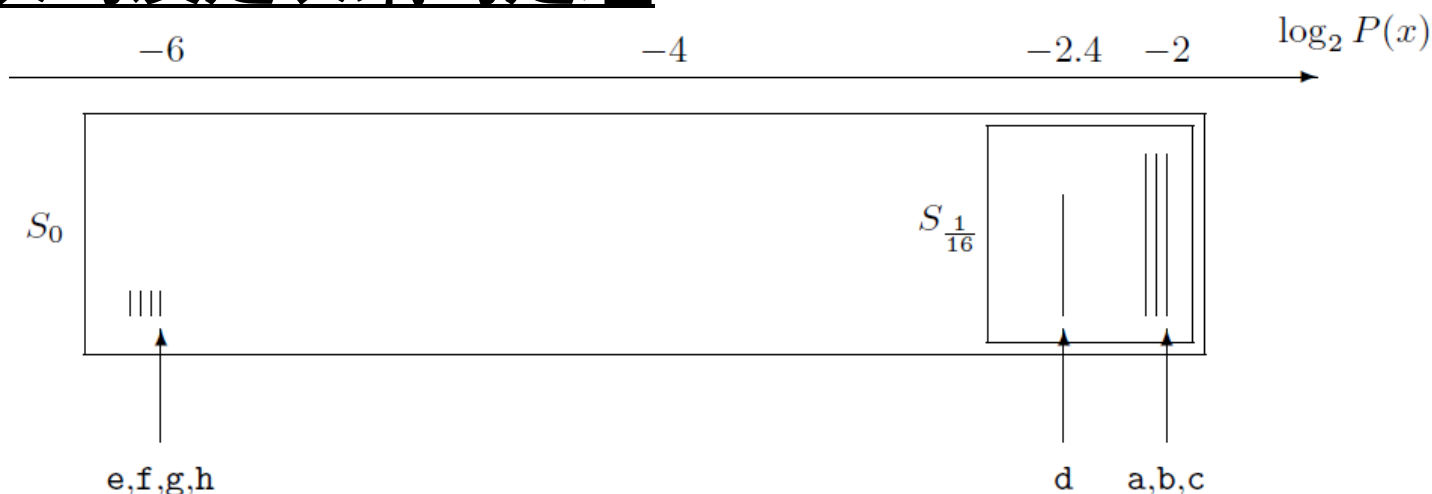
考虑编码错误概率  $P_E = \delta$  与编码形式。

(1) 考虑对每一个信源符号编等长码;

(2) 考虑接受错误概率  $1/16$ , 即  $\delta = 1/16$ ;

$\delta = 0$		$\delta = 1/16$	
$x$	$c(x)$	$x$	$c(x)$
a	000	a	00
b	001	b	01
c	010	c	10
d	011	d	11
e	100	e	—
f	101	f	—
g	110	g	—
h	111	h	—

## 4-3 定长码及定长编码定理

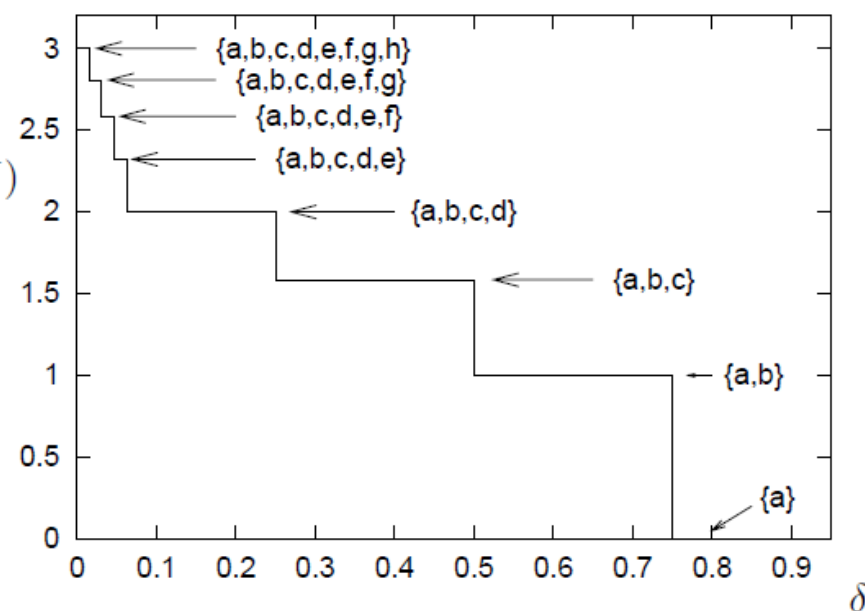


定义信源符号集合的一个最小子集  $S_\delta$ , 其中元素满足

$$P(x \in S_\delta) \geq 1 - \delta$$

每一个信源符号进行编码, 如果采用二进制编码, 则根据错误概率由**符号个数**反映的信息量为:

$$H_\delta(X) = \log_2 |S_\delta|$$



## 4-3 定长码及定长编码定理

【例题】考虑离散信源的  $\begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.9 & 0.1 \end{bmatrix}$  N次扩展信源

$X = x_1 x_2 x_3, \dots, x_N$ , 定义  $r(x)$  为某序列中1的个数, 可知该序列的出现概率为  $P(x) = p_0^{N-r(x)} p_1^{r(x)}$

类似可以定义最小集合  $S_\delta$  并计算  $H_\delta(X^N)$  当  $N=4, 10$  试确定其最小集合并绘制  $H_\delta(X^N)$  曲线。

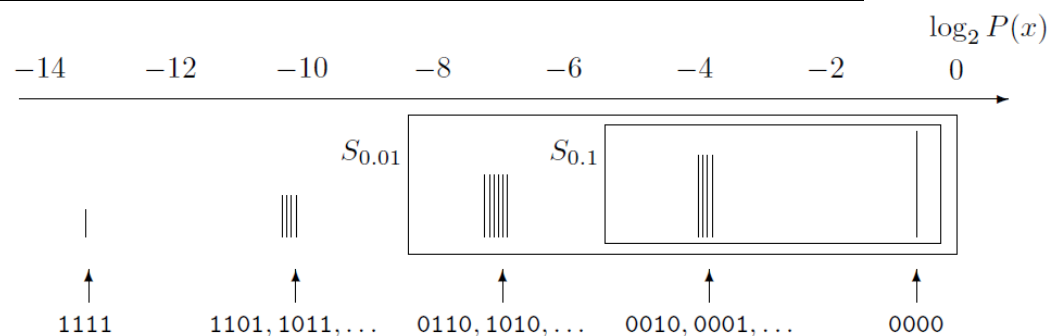
解: 当  $N=4$  时, 根据  $r(x) = 0, 1, 2, 3, \dots, r_{\max}(\delta)$ , 确定  $S_\delta$  最小集合

如  $r(x)=0$ ,  $S_\delta = \{0000\}$  对应错误概率  $\delta=0.3439$ ,  $\log(\delta) = -1.54$

$r(x)=1$ ,  $S_\delta = \{0001\}$  或  $\{0010\}$  或  $\{0100\}$  或  $\{1000\}$  对应错误概率

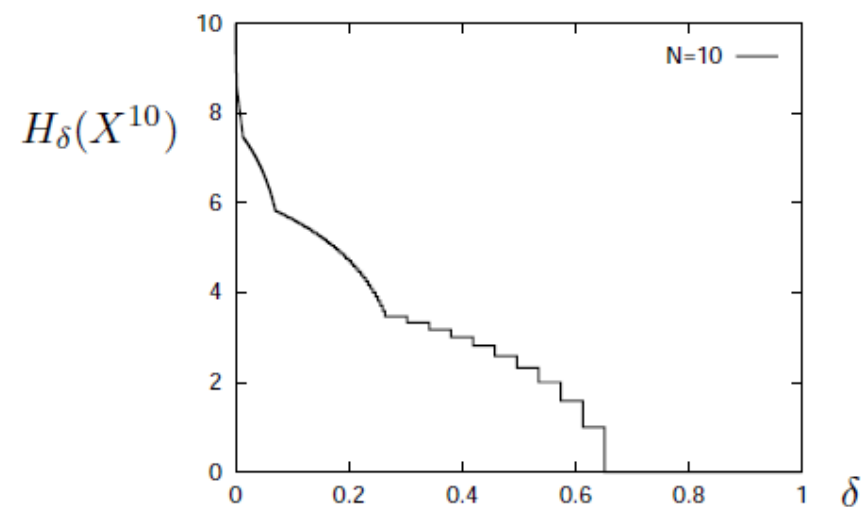
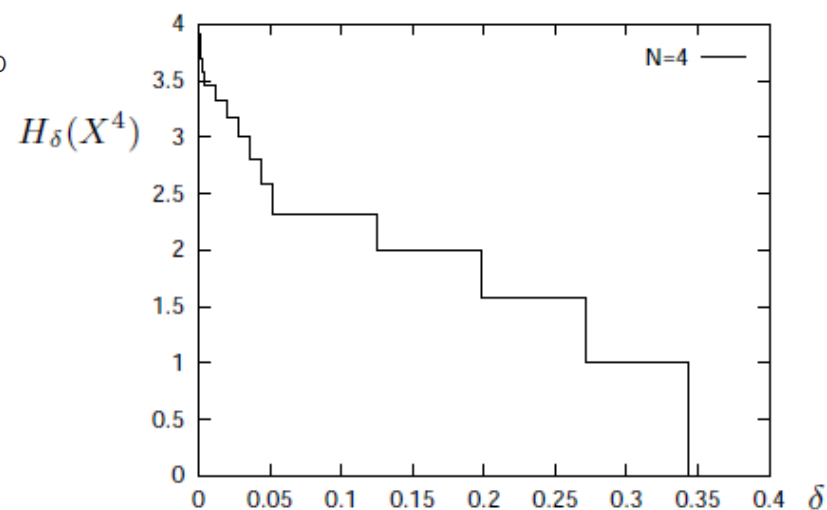
$\delta=0.0729$ ,  $\log(\delta) = -3.78$

## 4-3 定长码及定长编码定理

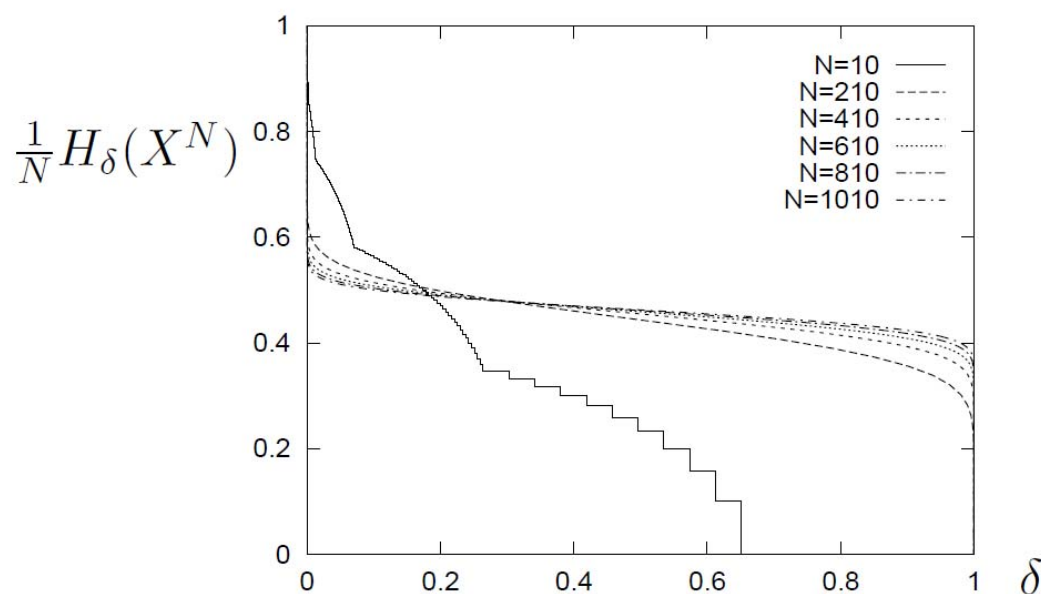


类似，当 $N=4$ 时，对所有符号的序列按照其概率大小排序。

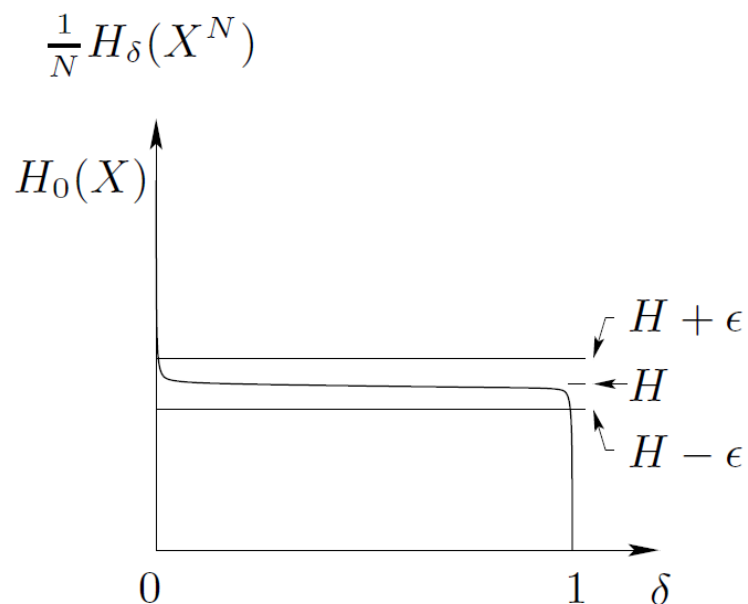
根据概率及序列含有1的个数，确定错误概率 $\delta$ ，并绘制



## 4-3 定长码及定长编码定理



类似，可以确定当  $N=10$  到 1010 时绘制  $\frac{1}{N}H_\delta(X^N)$



而当  $N$  趋于无穷大时，可得到左图（即是对香农信源编码理论的解释）

$$\left| \frac{1}{N}H_\delta(X^N) - H \right| < \epsilon$$

## 4-3 定长码及定长编码定理

**【问题】** 考虑英文电报32字母符号作为信源符号，利用二元码进行定长编码：则每个字母符号至少需要5位二元码才可唯一译码。

**【思考】** 考虑到第二章提到英文符号提供的极限熵近似接近1.4bit/sym, **如何提高信息传递效率？**

**方法1：** 考虑符号间的依赖关系，对信源的扩展信源进行编码，考虑符号间的依赖关系，有些信源符号序列不再出现，从而使得可能出现的符号序列个数小于 $q^N$ 。

**方法2：** 对于概率等于0或者非常小的序列不予编码，虽然会造成误差，但是当N足够长时这种误差概率可以任意小，既可以作到无失真的编码。

## 4-3 定长码及定长编码定理

### 定长信源编码定理：

离散无记忆信源的熵为 $H(s)$ ,如果对信源长度为 $N$ 的序列进行**定长编码**, 码符号集 $X$ 中有 $r$ 个码符号, 码长为 $l$ ,对于任意 $\varepsilon > 0$ , 只要满足  $\frac{l}{N} \geq \frac{H(S) + \varepsilon}{\log_2 r}$  则当 $N$ 足够大时, 可实现几乎无失真编码, 即译码错误概率任意小; 反之, 如果  $\frac{l}{N} < \frac{H(S) - 2\varepsilon}{\log_2 r}$  则不可能实现几乎无失真编码, 即当 $N$ 足够大时, 译码错误概率为1。

**定长信源编码定理给出了定长编码时每个信源符号最少所需的码符号的理论极限, 该值由 $H(s)$ 决定。**

## 4-3 定长码及定长编码定理

### 定长信源编码定理：

**[1]** 将定理中的公式改写成  $l \log_2 r > NH(s)$

左边表示长度为  $l$  的码符号序列能载荷的最大信息量；

右边表示长度为  $N$  的信源序列平均携带的信息量。

**定长编码定理**的意义：只要码字包含的信息量大于信源携带的信息量，总可实现几乎无失真编码。

**[2]** 定理的一般性证明是通过计算信源符号自信息的均值与方差，把信源消息分成两个互补集合，一个有编码，一个无编码，再利用契比雪夫不等式，求出有编码集合中码字个数的上下限，分别用上限证明正定理部分，用下限证明逆定理部分。



## 4-3 定长码及定长编码定理

**定义：** 设熵为 $H(S)$ 的离散无记忆信源，若对信源长为 $N$ 的符号序列进行定长编码，码符号集中码符号个数为 $r$ ，设码字长度为 $l$ ，定义  $R' = \frac{l}{N} \log_2 r$  比特/信源符号为**编码速率**，它表示平均每个信源符号编码后能够载荷的最大信息量。

可见编码速率大于信源的熵，才能实现几乎无失真编码，为衡量编码效果，引入

**定义：** 定义编码效率为  $\eta = \frac{H(S)}{R'} = \frac{H(S)}{\frac{l}{N} \log_2 r}$

因此，最佳等长编码的效率为  $\eta = \frac{H(S)}{H(S) + \varepsilon}$

可知  $\varepsilon = \frac{1-\eta}{\eta} H(S)$

## 4-3 定长码及定长编码定理

根据定长编码定理：可以证明  $N \geq \frac{D[I(s_i)]}{\varepsilon^2 \delta}$

【1】信源熵 $H(S)$ 是一个界限/临界值。当编码器输出的信息率超过这个临界值时，就能无失真译码，否则就不行。

【2】信源编码定理从理论上说明了编码速率接近于1，即

$$\eta = \frac{H(S)}{\frac{l}{N} \log_2 r} \rightarrow 1$$

理想编码器的存在性，代价是在实际编码时取无限长的信源符号( $N \rightarrow \infty$ )进行统一编码。

【3】可得在已知方差和信源熵的条件下，信源序列长度 $N$ 与编码效率和允许错误概率的关系：
$$N \geq \frac{D[I(s_i)]}{H^2(S)} \frac{\eta^2}{(1-\eta)^2 \delta}$$

## 4-3 定长码及定长编码定理

【例题】设有离散无记忆信源

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ 0.4 & 0.18 & 0.1 & 0.1 & 0.07 & 0.06 & 0.05 & 0.04 \end{bmatrix}$$

如果对信源符号采用定长二元编码，要求编码效率为90%，  
允许错误概率小于 $10^{-6}$ ，计算所需信源序列长度 $N$

解：可知信源熵 $H(S)=2.55\text{bit/sym}$

$$\text{自信息方差 } D[I(s_i)] = \sum_{i=1}^8 p(s_i)[- \log p(s_i)]^2 - H^2(S) = 7.82$$

$$\varepsilon = \frac{1-\eta}{\eta} H(S) = 0.28$$

$$N \geq \frac{D[I(s_i)]}{\varepsilon^2 \delta} = \frac{7.82}{0.28^2 \times 10^{-6}} \approx 9.8 \times 10^7 \approx 10^8$$

## 4-3 定长码及定长编码定理

### ■ N次扩展信源的数学模型

由于单符号离散信源的数学模型为：

$$\begin{bmatrix} X \\ P(X) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_r \\ p_1 & p_2 & \cdots & p_r \end{bmatrix} \text{ 其中 } \sum_{i=1}^r p_i = 1, H(X) = -\sum_{i=1}^r p_i \log p_i$$

那么信源的N次扩展信源用 $X^N$ 表示，它是具有 $r^N$ 个符号的新的离散信源，它的数学模型为：

$$\begin{bmatrix} X^N \\ P(\mathbf{X}) \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{r^N} \\ p(\alpha_1) & p(\alpha_2) & \cdots & p(\alpha_{r^N}) \end{bmatrix} \text{ 其中 } \sum_{i=1}^{r^N} p(\alpha_i) = 1$$

$$\text{且 } \alpha_i = a_{i1}a_{i2}a_{i3} \cdots a_{iN}, p(\alpha_i) = p(a_{i1}a_{i2}a_{i3} \cdots a_{iN})$$

## 4-3 定长码及定长编码定理

■当 $N$ 的值很大时，根据大数定律，这个序列会以很高的概率出现以下情况：符号 $x_1$ 约重复出现 $NP_1$ 次，符号 $x_2$ 约重复出现 $NP_2$ 次，...，符号 $x_N$ 约重复出现 $NP_N$ 次。

■这意味着当 $N$ 足够大时，将会以趋向于1的概率出现以下情况：扩展后的很多序列都具有相同的组成，因此也具有相同的概率。也就是说，当 $N$ 足够大时，扩展后有相当多的符号序列是等概的。

■可以将具有上述结构的序列称为典型序列，而将其余的序列称为非典型序列，如果典型序列的概率之和很大而非典型序列的概率之和很小，则仅用典型序列来代表扩展信源在很多场合是可行的。

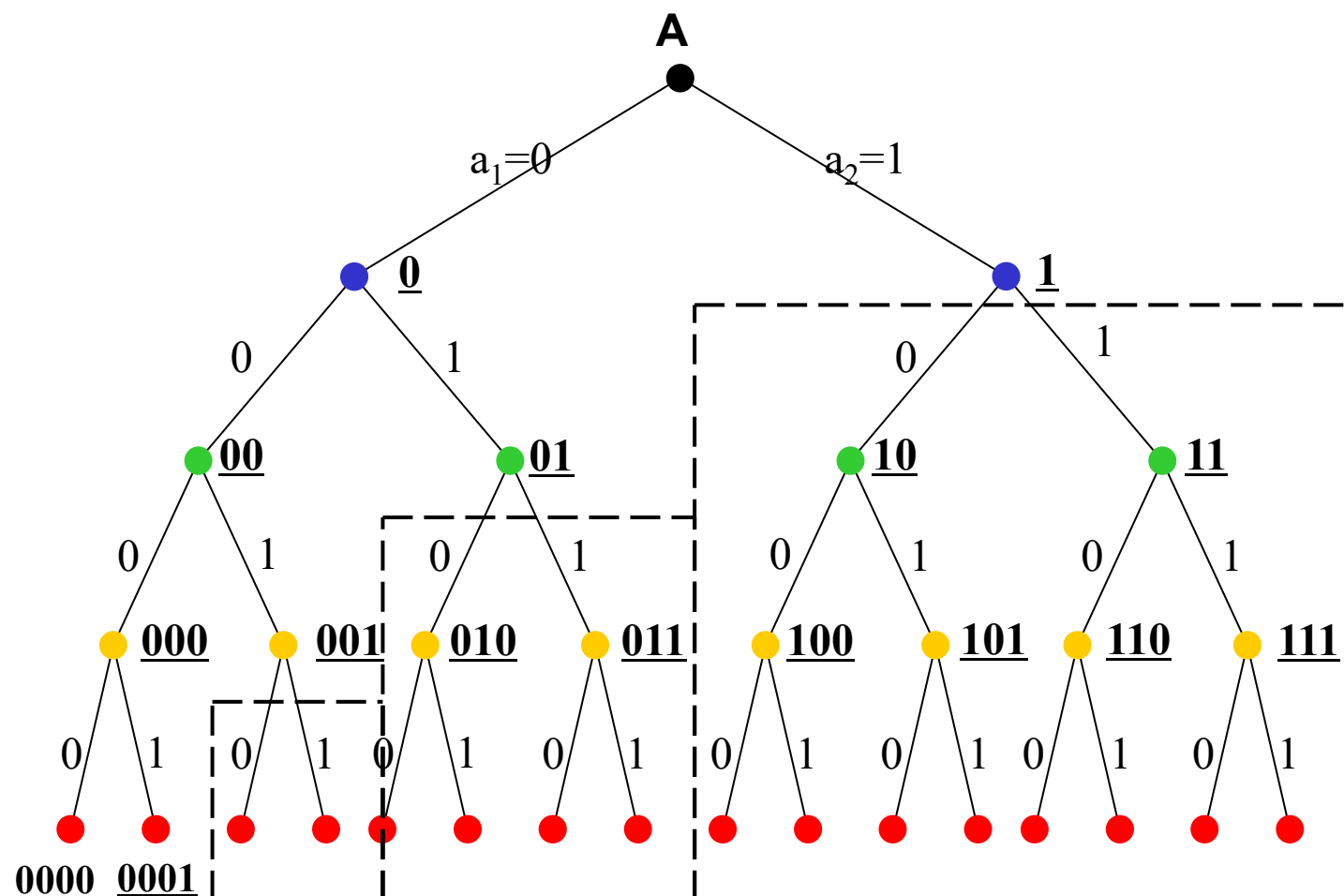
■既然所有的非典型序列的概率之和很小，对信源输出来说，忽略它们而引入的误差可以小于任何给定的值。

## 4-4 变长编码定理

### 单义可译定理：

设信源 $S$ 的符号集 $S:\{s_1, s_2, \dots, s_q\}$ ；码符号集 $X:\{a_1, a_2, \dots, a_r\}$ ； $q$ 个码字长度分别为 $n_1, n_2, \dots, n_q$ ，则存在唯一可译码的充分必要条件是 $q, r, n_i (i = 1, 2, \dots, q)$ 满足克拉夫特不等式，
$$\sum_{i=1}^q r^{-n_i} \leq 1$$

## 4-2 编码的定义



## 4-4 变长编码定理

**定理：** 设信源 $S$ 的符号集 $S:\{s_1, s_2, \dots, s_q\}$ ；码符号集 $X:\{a_1, a_2, \dots, a_r\}$ ； $q$ 个码字长度分别为 $n_1, n_2, \dots, n_q$ ，则存在唯一可译码的充分必要条件是 $q, r$ ， $n_i (i = 1, 2, \dots, q)$ 满足克拉夫特不等式，
$$\sum_{i=1}^q r^{-n_i} \leq 1$$

**例题：** 设信源符号集 $S:\{s_1, s_2, s_3, s_4\}$ ，采用二进制信道，码字长度分别为 $n_1=1, n_2=2, n_3=2, n_4=3$ ，是否存在这样的唯一可译码？如果码字长度为 $n_1=1, n_2=2, n_3=3, n_4=3$ 时，又是否存在这样的唯一可译码？



## 4-4 变长编码定理

解：根据克拉夫特（Kraft）不等式

$$\sum_{i=1}^q r^{-n_i} \leq 1$$

(1)、若  $n_1=1, n_2=2, n_3=2, n_4=3$  , 则  $\sum_{i=1}^4 2^{-n_i} = 9/8 > 1$  ,  
不满足Kraft不等式, 故不存在  $n_1=1, n_2=2, n_3=2, n_4=3$   
的唯一可译码。

(2)、若  $n_1=1, n_2=2, n_3=3, n_4=3$  , 则  $\sum_{i=1}^4 2^{-n_i} = 1$  ,  
满足Kraft不等式, 故存在  $n_1=1, n_2=2, n_3=3, n_4=3$   
的唯一可译码。

## 4-4 变长编码定理

**定理：**若存在一个码长为 $n_i (i = 1, 2, \dots, q)$ 的单义可译码，则一定存在一个具有相同码长的即时码

**结论：**

- (1) 克拉夫特不等式只能用于说明唯一可译码是否存在，并不能作为判别依据
- (2) 任何一个结构为 $q, r, n_i (i = 1, 2, \dots, q)$ 的唯一可译码，必定满足克拉夫特不等式；满足克拉夫特不等式的具有结构为 $q, r, n_i (i = 1, 2, \dots, q)$ 的编码，至少可以构成一个结构为 $q, r, n_i (i = 1, 2, \dots, q)$ 的即时码

## 4-4 变长编码定理

### 4.4.1 平均码长和有效性

**定义：**一个信源符号所需要的平均码符号数，就应该等于 $q$ 个码字长度 $n_i$ 在信源 $S$ 的概率空间 $P: \{p(s_1), p(s_2), \dots, p(s_q)\}$ 的统计平均值，即

$$\bar{n} = \sum_{i=1}^q p(s_i) n_i \quad \text{--单位：码符号 / 信源符号}$$

于是将 $\bar{n}$ 称为即时码的**平均码长**

## 4-4 变长编码定理

**定义：**即时码  $W : \{w_1, w_2, \dots, w_q\}$  每一个码符号所携带的平均信息量，即信息传输率（又称为码率）为：

$$R = \frac{H(S)}{\bar{n}} = \frac{\text{比特/信源符号}}{\text{码符号/信源符号}} = \frac{\text{比特}}{\text{码符号}}$$

若离散无噪声信道每传递一个码符号需用  $t$  秒钟时间，则离散无噪声信道在每秒钟时间内能传输的平均信息量，即信道的平均信息传输速率为：

$$R_t = \frac{R}{t} = \frac{H(S)}{\bar{n}t} = \frac{\text{比特/码符号}}{\text{秒/码符号}} = \frac{\text{比特}}{\text{秒}}$$

## 4-4 变长编码定理

**问题：**即时码 $W$ 的平均码长 $\bar{n}$ 是衡量其有效性高低的标准，因此要提高无失真信源编码的有效性，就要设法降低 $\bar{n}$

**解决方法：**

信源符号与码字之间的搭配问题

概率大的信源符号赋予码字长度小的码字；

概率小的信源符号赋予码字长度大的码字；

## 4-4 变长编码定理



问题:

平均码长能否无限制的小呢?  
有没有限度呢?

答案:

不是, 有限度

平均码长的界限定理

## 4-4 变长编码定理

### 4.4.2 平均码长的界限定理(单符号离散信源)

**定理：** 若离散无记忆信源  $S : \{s_1, s_2, \dots, s_q\}$  的信息熵为  $H(S)$ ，采用码符号集  $X : \{a_1, a_2, \dots, a_r\}$  进行变长编码，码字长度  $n_1, n_2, \dots, n_q$ ，且  $q, r, n_i$  满足 *Kraft* 不等式，一定存在一种无失真信源编码方法，其码字平均长度  $\bar{n}$  满足下列不等式：

$$\frac{H(S)}{\log r} \leq \bar{n} < \frac{H(S)}{\log r} + 1$$

## 4-4 变长编码定理

### 4.4.2 平均码长的界限定理

**定义：** 当平均码长 $\bar{n}$ 达到最小值时，相应的这种无失真信源编码称为最佳码

**结论：** 采用正确的编码原则，合理搭配码字长度 $n_i$ 与信源符号的概率分布 $p(s_i)$ ，可以使平均码长 $\bar{n}$ 减小，但是它的下降是有限度的，其下界是 $H(S)/\log r$ ；同时，平均码长在 $\bar{n}$ 不超过其上界值 $H(S)/\log r + 1$ 的情况下就可以构成即时码



## 4-4 变长编码定理

### 4.4.2 平均码长的界限定理

$$\begin{aligned}
 \frac{H(S)}{\log r} &= \frac{-\sum_{i=1}^q p(s_i) \log p(s_i)}{\log r} \\
 &= \frac{-\sum_{i=1}^q p(s_i) \log_r p(s_i)}{\log_r 2} \bigg/ \frac{\log_r r}{\log_r 2} \\
 &= -\sum_{i=1}^q p(s_i) \log_r p(s_i) \\
 &= H_r(S) \text{ --- } \frac{r\text{进制信息单位}}{\text{信源符号}}
 \end{aligned}$$

## 4-4 变长编码定理

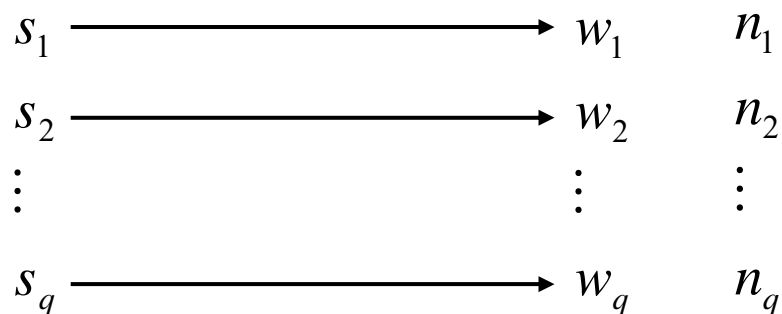
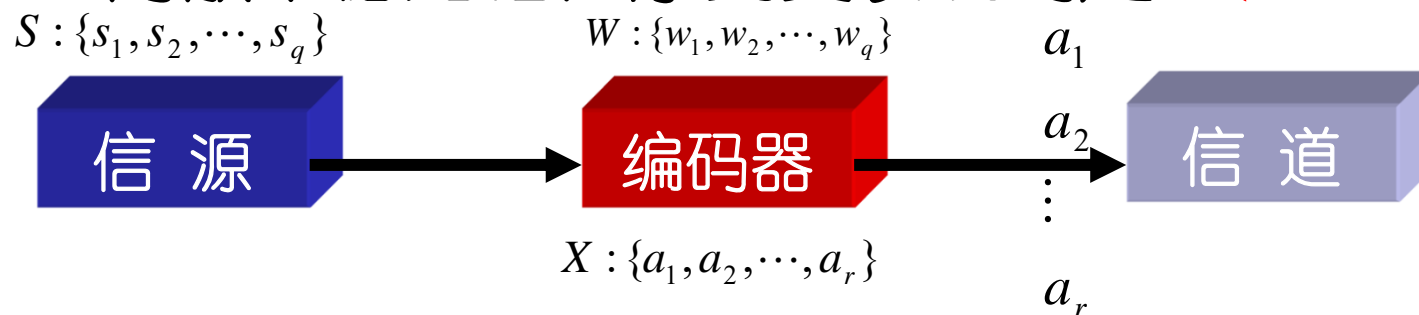
### 4.4.2 平均码长的界限定理

码字平均长度 $\bar{n}$ 满足下列不等式：

$$H_r(S) \leq \bar{n} < H_r(S) + 1$$

## 4-4 变长编码定理

### 4.4.3 离散平稳无记忆序列变长编码定理(香农第一定理)

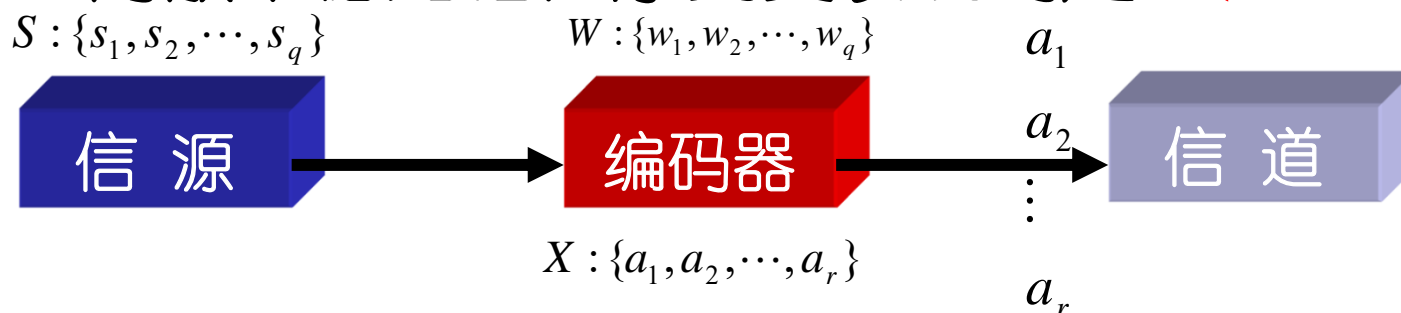


信源空间:

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & \cdots & s_q \\ p(s_1) & p(s_2) & \cdots & p(s_q) \end{bmatrix}$$

## 4-4 变长编码定理

### 4.4.3 离散平稳无记忆序列变长编码定理(香农第一定理)



$$\begin{array}{ccccccc}
 s_{11}s_{12} \cdots s_{1N} & \longrightarrow & \alpha_1 & \longrightarrow & w_1 & n_{N1} \\
 s_{21}s_{22} \cdots s_{2N} & \longrightarrow & \alpha_2 & \longrightarrow & w_2 & n_{N2} \\
 \vdots & & \vdots & & \vdots & \vdots \\
 s_{q^N1}s_{q^N2} \cdots s_{q^NN} & \longrightarrow & \alpha_{q^N} & \longrightarrow & w_{q^N} & n_{Nq^N}
 \end{array}$$

信源 $S$ 的 $N$ 次扩展信源 $\mathbf{S} = S^N = (S_1 S_2 \cdots S_N)$ 的信源空间

$$\begin{bmatrix} \mathbf{S} \\ P \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{q^N} \\ p(\alpha_1) & p(\alpha_2) & \cdots & p(\alpha_{q^N}) \end{bmatrix} \quad \text{其中 } \alpha_i = (s_{i1}s_{i2} \cdots s_{iN})$$

$$s_{i1}, s_{i2}, \dots, s_{iN} \in \{s_1, s_2, \dots, s_q\}, i1, i2, \dots, iN = 1, 2, \dots, q$$

$$i = 1, 2, \dots, q^N$$

## 4-4 变长编码定理

### 4.4.3 离散平稳无记忆序列变长编码定理(香农第一定理)

**定理：** 若离散无记忆信源 $S : \{s_1, s_2, \dots, s_q\}$ 的 $N$ 维扩展信源 $S^N$ 的平均符号熵为 $H_N(S) = H(S)$ ，必然存在一种无失真信源编码方法，使得平均码长 $\bar{n}$ 满足不等式

$$\frac{H(S)}{\log r} \leq \bar{n} < \frac{H(S)}{\log r} + \frac{1}{N}$$

将信源 $S$ 的 $N$ 次扩展信源 $S^N$ 的消息作为编码对象，使非延长码的码字与消息一一对应，则当信源扩展次数 $N$ 足够大时，信源 $S$ 的每一个信源符号 $s_i$ 所需要的平均码符号数，即平均码长可以无限接近于下界 $H(S)/\log r$ ，接近的程度随 $N$ 增加而增加

## 4-4 变长编码定理

平均码长： $\bar{n}$  --- 码符号/信源符号

信源熵值： $H(S)$  ---  $bit$  / 信源符号

无噪信道信息传输率： $R = H(S) / \bar{n}$  ---  $bit$  / 码符号

由于  $\frac{H(S)}{\log r} \leq \bar{n} \Rightarrow R \leq \log r = C$  ---  $bit$  / 码符号

**可变长无失真信源编码定理(香农第一定理)**，还可以称为**无噪声信道编码定理**，若信道的信息传输率 $R$ 不大于信道容量 $C$ ，总能对信源的输出进行适当的编码，使得在无噪声信道上能无失真的以最大信息传输率 $C$ 传输信息；但是要使信道的信息传输率 $R$ 大于信道容量 $C$ 而无差错地传输信息则是不可能的

## 4-4 变长编码定理

### 4.4.4 编码效率和编码的剩余度

定义：假设对信源进行无失真信源编码所得到的平均码长为 $\bar{n}$ ，定义 $\eta = \frac{H(S)}{\bar{n} \log r}$ 为编码效率

定义：编码的剩余度定义为 $\gamma = 1 - \eta = 1 - \frac{H(S)}{\bar{n} \log r}$   
用来衡量各种编码与最佳编码的差距

## 4-4 变长编码定理

**例题：**对于信源空间为  $\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ 3/4 & 1/4 \end{bmatrix}$ ，的离散

无记忆信源，采用二元 码符号（0,1）构造  
非延长码，在不同情况 下分析其编码效率

解：(1)用码符号对信源符号进行编码时，有

信源符号	码符号	码字
$s_1$	0	$W_2$
$s_2$	1	$W_1$

平均码长：  $\bar{n}_1 = 1$  (码符号/信源符号)

编码效率为：  $\eta_1 = \frac{H(S)}{\bar{n}_1 \log r} = 0.8113$



## 4-4 变长编码定理

(2) 用码符号对二次扩展信源进行编码时，有

消息 $\alpha_i$	$p(\alpha_i)$	码字
$a_1 a_1$	9/16	0
$a_1 a_2$	3/16	10
$a_2 a_1$	3/16	110
$a_2 a_2$	1/16	111

平均码长： $\bar{n}_2 = 27/32$  (码符号/信源符号)

编码效率为： $\eta_2 = \frac{H(S)}{\bar{n}_2 \log r} = 0.9615$

(3) 采用同样的方法，可计算

三次扩展信源的编码效率  $\eta_3 = 0.985$

四次扩展信源的编码效率  $\eta_4 = 0.991$

## 4-5 变长编码方法

### 4.5.1 香农编码方法

香农编码方法步骤：

(1)将信源消息符号按照其出现的概率大小依次排列

$$p(s_1) \geq p(s_2) \geq \cdots \geq p(s_q)$$

(2)确定满足以下不等式的整数码长度 $n_i$

$$-\log p(s_i) \leq n_i \leq -\log p(s_i) + 1$$

(3)为了编成唯一可译码，计算第 $i$ 个消息的累加概率

$$P_i = \sum_{k=1}^{i-1} p(s_k)$$

(4)将累加概率 $P_i$ 变成二进制数

(5)取 $P_i$ 二进制数的小数后 $n_i$ 位即为该消息符号的二进制码字

## 4-5 变长编码方法

**例：** 设信源共有7个符号消息，其概率和累加概率如表下表  
 如示，试用Shannon编码方法进行编码。

符号	符号概率	累加概率	$-\log p(a_i)$	码长	码字
$a_1$	0.20	0	2.34	3	000
$a_2$	0.19	0.2	2.41	3	001
$a_3$	0.18	0.39	2.48	3	011
$a_4$	0.17	0.57	2.56	3	100
$a_5$	0.15	0.74	2.74	3	101
$a_6$	0.10	0.89	3.34	4	1110
$a_7$	0.01	0.99	6.66	7	1111110

平均码长：

$$\bar{n} = \sum_{i=1}^q p(s_i) n_i = 3.14 \text{ (码符号 / 信源符号)}$$

编码效率：

$$\eta = \frac{H(S)}{\bar{n} \log r} = \frac{2.61}{3.14} = 0.8312$$

**Shannon编码方法冗余度大，实用性不强，但它是依据  
 编码定理而来的，具有重要的理论意义**

## 4-5 变长编码方法

### 4.5.2 费诺编码方法 (Fano)

费诺编码方法属于概率匹配编码，它不是变长编码方法方法，费诺编码方法步骤：

(1) 将信源消息符号按照其出现的概率大小依次排列

$$p(s_1) \geq p(s_2) \geq \dots \geq p(s_q);$$

(2) 将依次排列的信源符号按照概率值分为两大组，使两个组的概率之和接近相同，对各个组分别赋予一个二进制码元0,1；

(3) 将每一大组中的信源符号再分成两组，使划分后两个组的概率和接近相等，对各个组分别赋予一个二进制码元0,1；

(4) 如此重复，直到每个分组只剩下一个信源符号为止；

(5) 信源符号所对应的码字就是费诺码；

## 4-5 变长编码方法

信源符号	$p(s_i)$	第一次分组		第二次分组		第三次分组		第四次分组		码字	码长
$s_1$	0.2	0.57	0	0.2	0					00	2
$s_2$	0.19			0.37	1	0.19	0			010	3
$s_3$	0.18					0.18	1			011	3
$s_4$	0.17	0.43	1	0.17	0					10	2
$s_5$	0.15			0.26	1	0.15	0			110	3
$s_6$	0.1					0.11	1	0.1	0	1110	4
$s_7$	0.01							0.01	1	1111	4

平均码长:

$$\bar{n} = \sum_{i=1}^q p(s_i) n_i = 2.74 \text{ (码符号 / 信源符号)}$$

编码效率:

$$\eta = \frac{H(S)}{\bar{n} \log r} = 0.9526$$

**Fano 编码方法比Shannon 编码方法的平均码长小，编码效率高**

## 4-5 变长编码方法

### 4.5.3 霍夫曼编码方法 (Huffman)

Huffman提出一种无失真信源编码方法 (1952年)

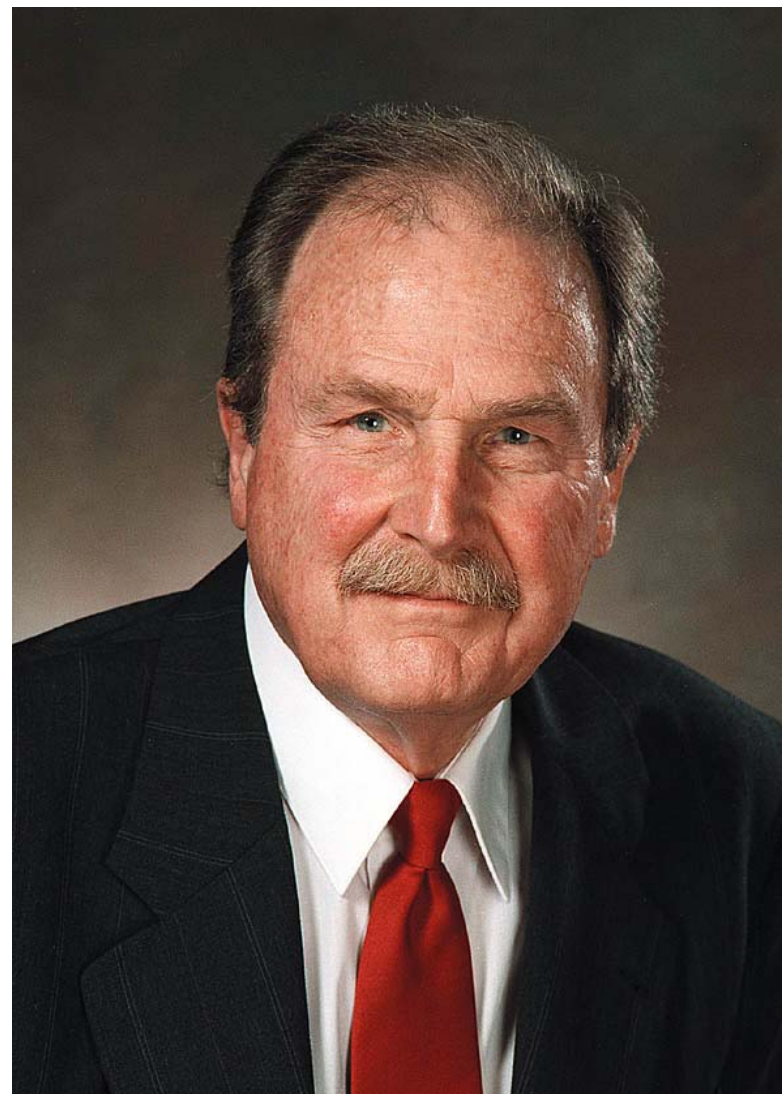
它针对给定信源的信源空间和规定的码符号集，合理利用信源的统计特性，构造的唯一可译的非延长码，具有尽可能小的平均码长，使无失真信源编码的有效性达到了可能范围中的最好值

## 4-5 变长编码方法

### 4.5.3 霍夫曼编码方法 (Huffman)

**D. A. Huffman** 在 1952 年的论文“最小冗余度代码的构造方法 ( **A Method for the Construction of Minimum Redundancy Codes** ) ”中提出的。

1952 年时，**Huffman** 还是麻省理工学院的一名学生，他为了向老师**Robert Fano**教授证明自己可以不参加某门功课的期末考试，才设计了这个看似简单，但却影响深远的编码方法。



## 4-5 变长编码方法

### 4.5.3 霍夫曼编码方法 (Huffman)

若以 $X : \{a_1, a_2, \dots, a_r\}$ 为码符号集，用霍夫曼编码方法，

对信源空间为 $\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & \dots & s_q \\ p(s_1) & p(s_2) & \dots & p(s_q) \end{bmatrix}$ 的离散无记

忆信源 $S$ ，进行无失真信源编码

其步骤如下：



## 4-5 变长编码方法

二元Huffman编码步骤：

- (1) 将 $q$ 个信源消息符号按照其出现的概率大小依次排列 $p(s_1) \geq p(s_2) \geq \cdots \geq p(s_q)$ ;
- (2) 取其中两个概率最小的信源符号分别配以0,1两个码元。将这两个概率最小的信源符号合并成一个符号（虚拟符号），得到只包含 $(q-1)$ 个符号的新信源，称为信源 $S$ 的第一次缩减信源 $S_1$ ;

## 4-5 变长编码方法

- (3) 将缩减信源 $S_1$ 的符号依然按照概率大小以递减的次序排列，再将其中两个最小的概率对应的信源符号分别配以0,1两个码元，再合并成一个虚拟信源符号，从而得到 $q-2$ 个符号的缩减信源 $S_2$ ；
- (4) 如此重复，直到只剩下两个信源符号为止，分别配以0,1两个码元；
- (5) 从最后一级缩减信源开始向前返回，就得到各个信源符号 $s_i$ 所对应的码符号序列也就是码字 $w_i$ ；

## 4-5 变长编码方法

**例题：**用二元信道以 $X:\{0,1\}$ 为码符号集，用Huffman编码方法对信源空间为 $[S,P]$ 的离散无记忆信源 $S$ 进行编码

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \\ 0.2 & 0.19 & 0.18 & 0.17 & 0.15 & 0.1 & 0.1 \end{bmatrix}$$

## 4-5 变长编码方法

**例题：**用二元信道以 $X:\{0,1\}$ 为码符号集，用Huffman编码方法对信源空间为 $[S,P]$ 的离散无记忆信源 $S$ 进行编码

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \\ 0.2 & 0.19 & 0.18 & 0.17 & 0.15 & 0.1 & 0.1 \end{bmatrix}$$

[illegible]

## 4-5变长编码方法

平均码长：

$$\bar{n} = \sum_{i=1}^7 p(s_i) n_i$$

= 2.72 码符号 / 信源符号

信息传输率：

$$R = \frac{H(s)}{\bar{n}} = \frac{2.61}{2.72}$$

= 0.9596 bit / 码符号

信源符号	码字	码长
$s_1$	10	2
$s_2$	11	2
$s_3$	000	3
$s_4$	001	3
$s_5$	010	3
$s_6$	0110	4
$s_7$	0111	4

**定义：**对于某一信源和码符号集，若有一个唯一可译码，其平均码长 $\bar{n}$ 小于所有其它唯一可译码的平均码长 $\bar{n}$ ，则称该码为最佳码

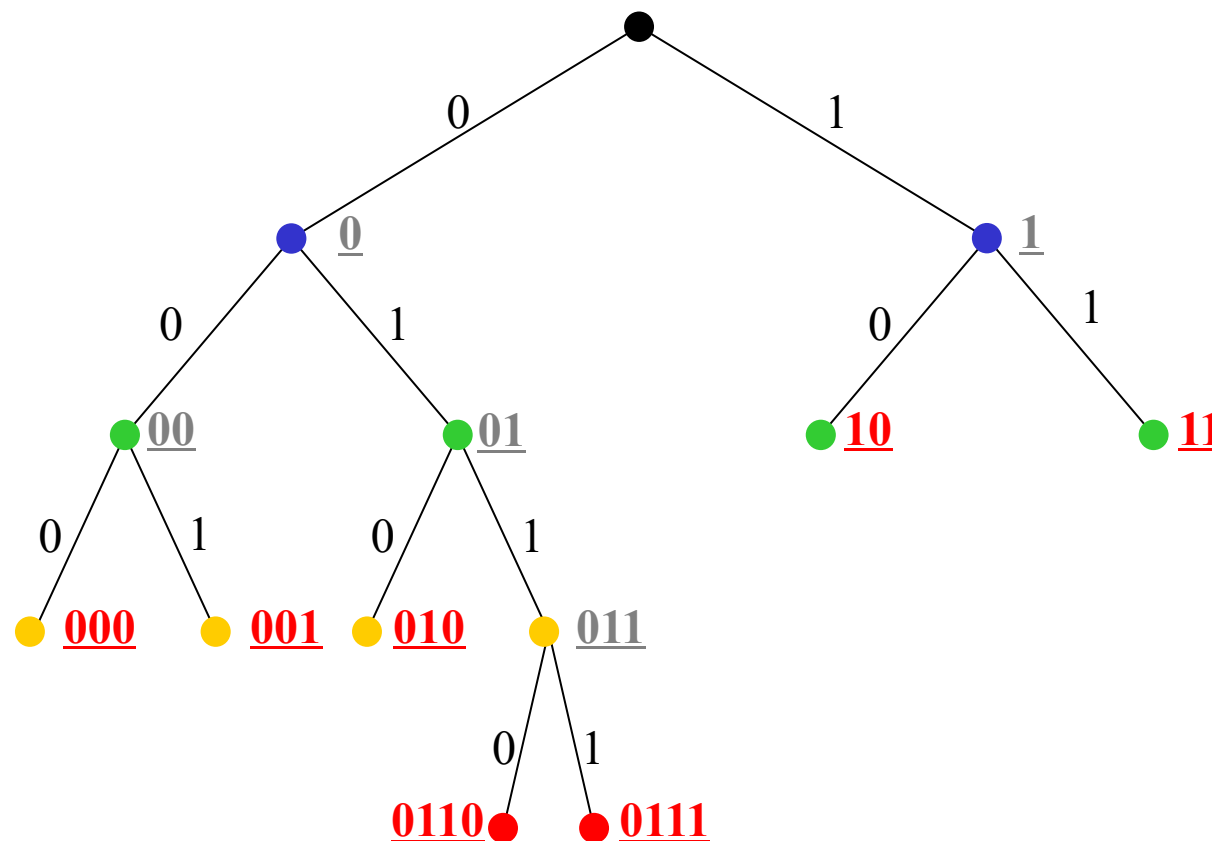
## 4-5 变长编码方法

### 问题1: Huffman编码是否是唯一的?

- (1)在各次信源缩减过程中，赋予信源概率最小的最后两个符号的码元0, 1是任意安排的，所以可以得到不同的Huffman编码，但是不会影响码字长度
- (2)对信源进行缩减的过程中，两个概率最小的符号合并以后的概率与其它的信源符号的概率相同时，缩减所得到的虚拟符号的排列位置可以有多种选择，也会造成结构相同，有效性相同，但是形式不同的Huffman编码

## 4-5 变长编码方法

**问题2:** Huffman编码得到的是否是非延长码?



## 4-5 变长编码方法

**例题：**采用二元信道，以 $X:\{0,1\}$ 为码符号集，用 Huffman 编码方法对信源空间为 $[S,P]$

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \end{bmatrix}$$

的离散无记忆信源 $S$ 进行无失真信源编码



# 4-5 变长编码方法

信源符号 $S_i$	概率 $p(s_i)$	第1次 缩减	缩减 信源 $S_1$	第2次 缩减	缩减 信源 $S_2$	第3次 缩减	缩减 信源 $S_3$	第4次 缩减	码字	码长	平均码长
$s_1$	0.4	→	0.4	→	0.4	↘	0.6	0 1.0	1	1	2.2 码符号/ 信源符号
$s_2$	0.2	→	0.2	↘	0.4	0 1	0.4	1	01	2	
$s_3$	0.2	→	0.2	0 1	0.2				000	3	
$s_4$	0.1	0 1	0.2						0010	4	
$s_5$	0.1								0011	4	
$s_1$	0.4	→	0.4	↘	0.4	↘	0.6	0 1.0	00	2	2.2 码符号/ 信源符号
$s_2$	0.2	↘	0.2	↘	0.4	0 1	0.4	1	10	2	
$s_3$	0.2	↘	0.2	0 1	0.2				11	2	
$s_4$	0.1	0 1	0.2						010	3	
$s_5$	0.1								011	3	

## 4-5 变长编码方法

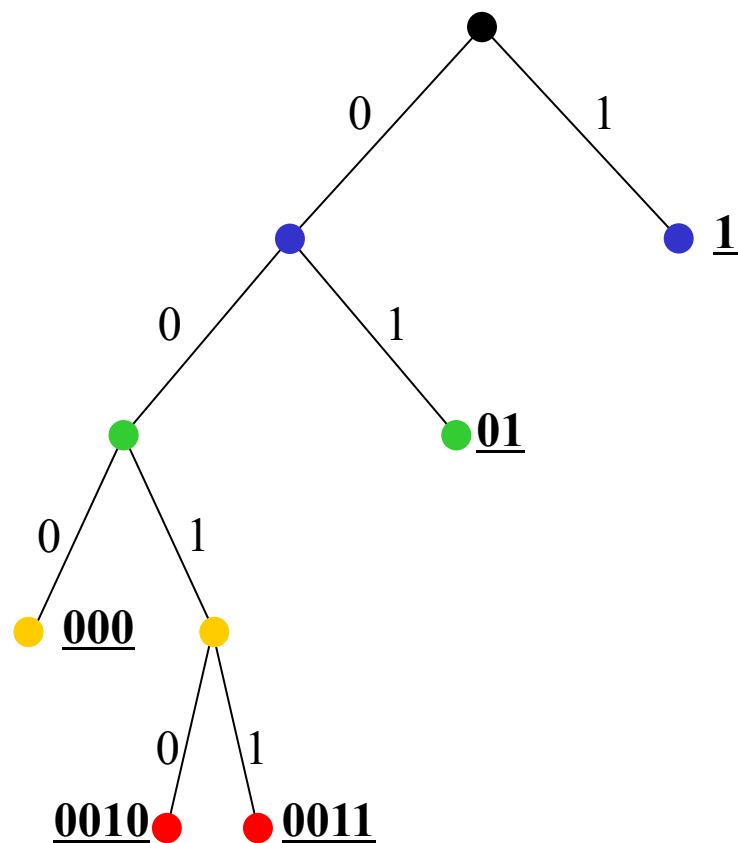
**定义：** 设码字  $W : \{w_1, w_2, \dots, w_q\}$  中，各个码字的长度为  $n_i$ ，平均码长为  $\bar{n}$ ，则码字  $W : \{w_1, w_2, \dots, w_q\}$  中各个码字的长度  $n_i$  与其平均码长  $\bar{n}$  的差的平均值  $\sigma_w^2 = E\{(n_i - \bar{n})^2\}$  称为码字  $W$  的码长方差

Huffman编码方法的特点：

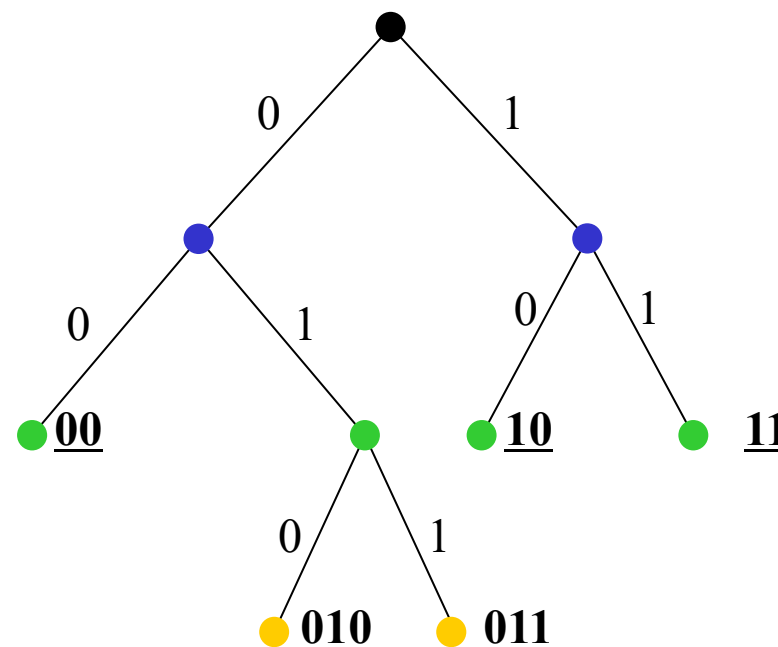
- (1) 保证概率大的信源符号对应短码，概率小的信源符号对应长码，并且所有短码得到了充分利用
- (2) 每次缩减信源中概率最小的  $r$  个符号对应的码符号序列中总是最后一个码符号不同，前面各位码符号均相同，保证了Huffman编码是即时码

## 4-5 变长编码方法

相应码树及其码方差如下：



$$\begin{aligned}\sigma_w^2 &= E\{(n_i - \bar{n})^2\} \\ &= \sum_{i=1}^5 p(s_i)(\bar{n} - n_i)^2 = 1.36\end{aligned}$$



$$\begin{aligned}\sigma_w^2 &= E\{(n_i - \bar{n})^2\} \\ &= \sum_{i=1}^5 p(s_i)(\bar{n} - n_i)^2 = 0.16\end{aligned}$$

## 4-5 变长编码方法

### Maltab Huffman 编码实现:

```
[dict, avglen] = huffmandict(symbols, prob)
```

symbols: 信源符号

prob: 信源符号概率

dict: Huffman编码输出码字

avglen: 编码输出的平均码字长度

### 举例:

```
symbols = [1:5] % Alphabet vector
```

```
prob = [.3 .3 .2 .1 .1] % Symbol probability vector
```

```
[dict, avglen] = huffmandict(symbols, prob)
```

### Maltab Huffman 编码实现:

```
Sig_encoded = huffmanenco (sig, dict)
```

```
Deco = huffmandeco (Sig_encoded, dict)
```

## 4-5 变长编码方法

```
>> symbols = [1:5]           % Alphabet vector
prob = [.3 .3 .2 .1 .1]      % Symbol probability vector
[dict, avglen] = huffmandict(symbols, prob)
```

symbols =

1 2 3 4 5

prob =

0.3000 0.3000 0.2000 0.1000 0.1000

dict =

5×2 [cell](#) 数组

```
{[1]} {1×2 double}
{[2]} {1×2 double}
{[3]} {1×2 double}
{[4]} {1×3 double}
{[5]} {1×3 double}
```

	1	2	3	4
1	1	[0,1]		
2	2	[0,0]		
3	3	[1,0]		
4	4	[1,1,1]		
5	5	[1,1,0]		

```
>> dict{5,:}
```

ans =

5

ans =

1 1 0

## 4-5 变长编码方法

### r元Huffman编码步骤:

- (1)将 $q$ 个信源符号 $s_1, s_2, \dots, s_q$ 按照其概率分布 $p(s_1), p(s_2), \dots, p(s_q)$ 的大小,以递减的次序,从上到下排成一行;
- (2)对处于最下面的概率最小的 $r$ 个信源符号,一一对应的赋予码符号 $a_1, a_2, \dots, a_r$ ,将这 $r$ 个概率最小的符号相应的概率相加,所得到的和用一个虚拟符号表示,与余下的 $(q-r)$ 个信源符号组成含有 $[(q-r)+1]$ 个符号的第一次缩减信源 $S_1$ ;

## 4-5 变长编码方法

(3) 缩减信源 $S_1$ 中的符号，仍按照其概率大小，从上到下排列，对处于最下面的概率最小的 $r$ 个符号，按照步骤2中的同样顺序，一一对应的赋予码符号 $a_1, a_2, \dots, a_r$ ，将这 $r$ 个概率最小的符号相应的概率相加，所得到的和用一个虚拟符号表示，与余下的 $\{[(q-r)+1]-r\}$ 个符号组成含有 $\{[(q-r)+1]-r+1\}$ 个符号的第二次缩减信源 $S_2$ ；

(4) 按照以上方法依次继续下去，每次缩减所减少的符号数是 $(r-1)$ ，缩减到第 $l$ 次时，总共减少的符号数是 $[(r-1)l]$ ，第 $l$ 次缩减信源所含有的符号数是 $[q-(r-1)l]$ ，当缩减信源 $S_l$ 含有符号数 $[q-(r-1)l]$ 大于码符号数 $r$ 时，缩减过程继续下去；

## 4-5 变长编码方法

(5)当第 $\alpha$ 次缩减信源 $S_\alpha$ 中所含有的符号数 $[q - (r - 1)\alpha]$ 正好等于码符号集符号数 $r$ 时,表明缩减过程到了最后一次,对于这余下的 $r$ 个符号,按照以前的顺序依次赋予 $a_1, a_2, \dots, a_r$ , 且这 $r$ 个概率之和一定等于1;

(6)从最后赋予的码符号开始,沿着每一信源符号在各次缩减过程中得到的码符号的行进路线向前返回,到达每一信源符号,按照先后次序,把返回路途中所遇到的码符



## 4-5 变长编码方法

号排成码符号序列,这个序列就是返回路线  
终点信源符号相应的码字,完成编码过程;  
(7)当第 $\alpha$ 次缩减信源 $S_\alpha$ 中所含有的符号数  
 $[q - (r - 1)\alpha]$ 小于码符号集符号数 $r$ 时, 停  
止缩减过程, 在原来概率大小排列的下方,  
增加 $m$ 个概率为0的虚假信源符号  
 $s_1', s_2', \dots, s_m'$ , 且 $m$ 等于码符号集符号数 $r$   
与第 $\alpha$ 次缩减信源 $S_\alpha$ 中含有的符号数之差,  
此时组成新的信源 $S'$ ,符号数为 $Q = q + m$ .  
然后按照步骤1-5对新信源进行缩减;

## 4-5 变长编码方法

**例题：**采用三元信道以 $X:\{0,1,2\}$ 为码符号集，用 Huffman 编码方法对信源空间为 $[S,P]$ 的离散无记忆信源 $S$ 进行无失真信源编码

$$\begin{bmatrix} S \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ 0.24 & 0.2 & 0.18 & 0.16 & 0.14 & 0.08 \end{bmatrix}$$

解：首先验证信源符号数 $q$ ，码符号数 $r$ 的编码系统是否满足 $q-(r-1)a=r$

该题中： $6-(3-1)*2=2<3$

其次，设置： $m=r-[q-(r-1)a]$ 个虚假信源符号，其概率为0

该题中： $m=r-[q-(r-1)a]=3-[6-(3-1)*2]=1$  个虚假信源符号  $s_1'$  其概率为 $p(s_1')=0$

## 4-5 变长编码方法

得到新的信源空间如下

$$\begin{bmatrix} S' \\ P \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s'_1 \\ 0.24 & 0.2 & 0.18 & 0.16 & 0.14 & 0.08 & 0 \end{bmatrix}$$
[illegible]

## 4-6 实用的无失真信源编码方法

游程编码主要用于黑白二值文件、传真的数据压缩。由于传真文件中连“0”和连“1”较多。**这些连“0”或连“1”的字符串称为游程(Run-Length)**。对游程长度进行霍夫曼编码或其他编码处理就可以达到压缩数据的目的。

下图是一幅 $10 \times 50$ 黑白二值图像。

其中有55个自游程以及54个黑游程

## 4-6.1 游程编码方法

游程编码(Run-Length Coding)又称“**运行长度编码**”或“**行程编码**”，是一种统计编码，该编码属于无损压缩编码。

**基本原理是：**用一个符号值或串长代替具有相同值的连续符号（连续符号构成了一段连续的“行程”。行程编码因此而得名），使符号长度少于原始数据的长度。

例如：111112222233334444455555555

游程编码为：(1, 6) (2, 5) (3, 4) (4, 5) (5, 5)。可见，游程编码的位数远远少于原始字符串的位数。

## 4-6.1 游程编码方法

根据游程编码的定义和性质可知，游程编码是一种二值图像的很有效表示方法。

在对图像数据进行编码时，沿一定方向排列的具有相同灰度值的像素可看成是连续符号，用字串代替这些连续符号，可大幅度减少数据量。

在游程编码中经常运用两种方法：

一种是使用1的起始位置和1 的游程长度；

另一种是仅仅使用游程长度，如果第一个编码值为0，则表示游程长度编码是从0像素的长度开始。

### 4-6.1 游程编码方法

如果一个127像素白游程进行编码。仅需1个byte表示，为11111111，第一位为颜色，后7位表示长度。

因此压缩率为  $127/8=15.9$

但是上图每一个游程均未超过127，如果用前述压缩方法：

$$8 \cdot (54 + 55) = 872 \text{ bit} > 500 \text{ bit}$$

## 没有达到压缩效果，所需bit位反而增加，怎么办？

## 4-6.1 游程编码方法

### 改进办法:

- 1) 考虑到颜色交替出现，只用给第一个游程颜色进行编码。特殊情况是如果游程超过127时采用一个全1的码字表示后续码字不变色。此时需要 $1+7(54+55)=764\text{bit} < 872\text{bit}$
- 2) 考虑到整体编码游程长度特点。可以进一步减少表示游程长度的比特数。如采用3bit表示游程长度（0-7），上例仅需 $459\text{bit} < 500\text{bit} < 764\text{bit} < 872\text{bit}$
- 3) 考虑到不同颜色游程分布情况，可以对黑白游程选用不同长度游程进行编码。



## 4-6.1 游程编码方法

### MH (Modified Huffman) 编码

**文件传真：**是指一般文件、图纸、手写稿、表格、报纸等文件的传真，这类图像的像素只有黑、白两个灰度等级，因此文件传真编码属于二值图像的压缩编码。

**直接编码：**将文件图纸在空间上离散化，如把一页文件分成 $n \times m$ 个像素，由于文件传真是二值电平的，则每个像素可用一位二进制码（0或1）代表，这种方式称为直接编码。

国际规定：一张A4幅面文章（210mm×297mm）有1188或2376条扫描线，按每条扫描线有1728个像素的扫描分辨率（相当于垂直4或8线/mm，水平8点/mm）计算，一张A4文件约有2.053M像素/公文纸或4.106M像素/A4公文纸

## 4-6.1 游程编码方法

### MH (Modified Huffman) 编码

如果对A4文件扫描后用Huffman编码方法，Huffman编码是根据信源的统计特性分配码长与码型的，这就要求确切掌握文件信源的所有可能的样本概率。

那么需要对以1728的行扫描分辨率的结果，即 $2^{1728}$ 种可能的黑白游程进行编码和译码（增加实现的复杂度）。

要统计上述各种情况及其困难，更无法用Huffman编码进行编码压缩。就算可以建立编码字典也会因为字典太大（对照表和转换编码）无法实现。

## 4-6.1 游程编码方法

### MH (Modified Huffman) 编码

#### MH编码特点:

- 1) 在码表的制定上，不是根据实际待传送文件的游程分布，而是以CCITT推荐的八种文件样张或我国原邮电部推荐的七种典型样张所测定的游程概率分布为依据来制定的。
- 2) 为了进一步减小码表数，采用截断Huffman编码方法。根据对传真文件的统计结果可知，黑、白游程长度在0 ~ 63的情况居多，不需对全部的 $2^{1728}$ 种黑、白游程长度进行编码，而是将码字分为**结尾码**和**构造码**（或称形成码）两种。**结尾码R**，是针对游程长度为0 ~ 63的情况，直接按游程统计特性制定对应的Huffman码表；而**构造码K**是对长度为64的倍数的游程长度进行编码。

## MH码表——结尾码R

游程长度	白游程码字	黑游程码字	游程长度	白游程码字	黑游程码字
0	00110101	0000110111	32	00011011	000001101010
1	000111	010	33	00010010	000001101011
2	0111	11	34	00010011	000011010010
3	1000	10	35	00010100	000011010011
4	1011	011	36	00010101	000011010100
5	1100	0011	37	00010101	000011010101
6	1110	0010	38	00010111	000011010110
7	1111	00011	39	00101000	000011010111
8	10011	000101	40	00101001	000001101100
9	10100	000100	41	00101010	000001101101
10	00111	0000100	42	00101011	000011011010
11	01000	0000101	43	00101100	000011011011
12	001000	0000111	44	00101101	000001010100
13	000011	00000100	45	00000100	000001010101
14	110100	00000111	46	00000101	000001010110
15	110101	000011000	47	00001010	000001010111
16	101010	0000010111	48	00001011	000001100100
17	101011	0000011000	49	01010010	000001100101
18	0100111	0000001000	50	01010011	000001010010
19	0001100	00001100111	51	01010100	000001010011
20	0001000	00001101000	52	01010101	000000100100
21	0010111	00001101100	53	00100100	000000110111
22	0000011	00000110111	54	00100101	000000111000
23	0000100	00000101000	55	01011000	000000100111
24	0101000	00000010111	56	01011001	000000101000
25	0101011	00000011000	57	01011010	000001011000
26	0010011	000011001010	58	01011011	000001011001
27	0100100	000011001011	59	01001010	000000101011
28	0011000	000011001100	60	01001011	000000101100
29	00000010	000011001101	61	00110010	000001011010
30	00000011	000001101000	62	00110011	000001100110
31	00011010	000001101001	63	00110100	000001100111

## MH码表——构造码K

游程长度	白游程码字	黑游程码字	游程长度	白游程码字	黑游程码字
64	11011	0000001111	960	011010100	0000001110011
128	10010	000011001000	1024	011010101	0000001110100
192	010111	000011001001	1088	011010110	0000001110101
256	0110111	000001011011	1152	011010111	0000001110110
320	00110110	000000110011	1216	011011000	0000001110111
384	00110111	000000110100	1280	011011001	0000001010010
448	01100100	000000110101	1344	011011010	0000001010011
512	1100101	0000001101100	1408	011011011	0000001010100
576	01101000	0000001101101	1472	010011000	0000001010101
640	01100111	0000001001010	1536	010011001	0000001011010
704	011001100	0000001001011	1600	010011010	0000001011011
768	011001101	0000001001100	1664	011000	0000001100100
832	011010010	0000001001101	1728	010011011	0000001100101
896	011010011	00000011100110	EOL	000000000001	000000000001

- 当游程长度 $L < 64$ 时，可直接引用结尾码表示；
- 当游程长度 $L$ 在 $64 \sim 1728$ 之间，用一个构造码加上相应的结尾码即成为相应的码字。

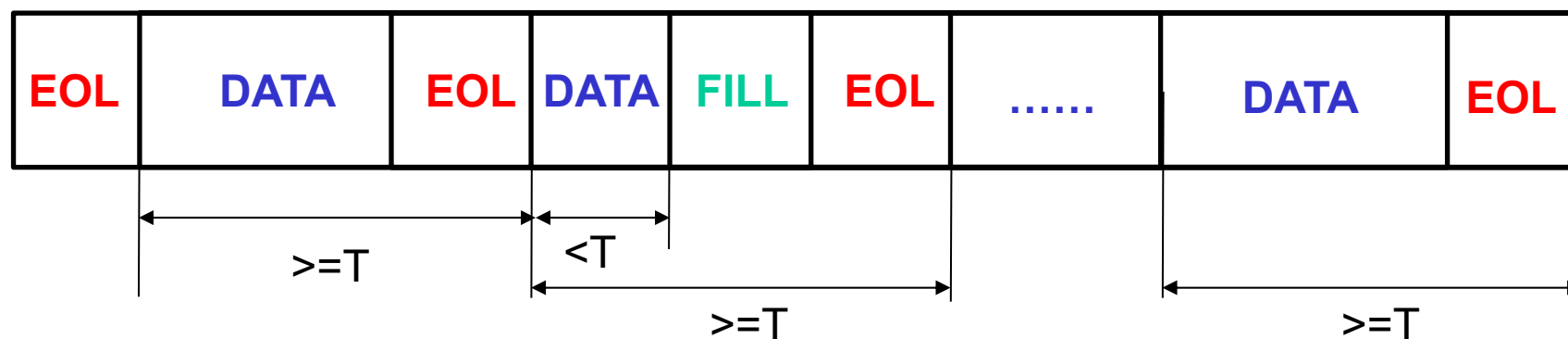
如  $128 = 128 + 0$

$129 = 128 + 1$

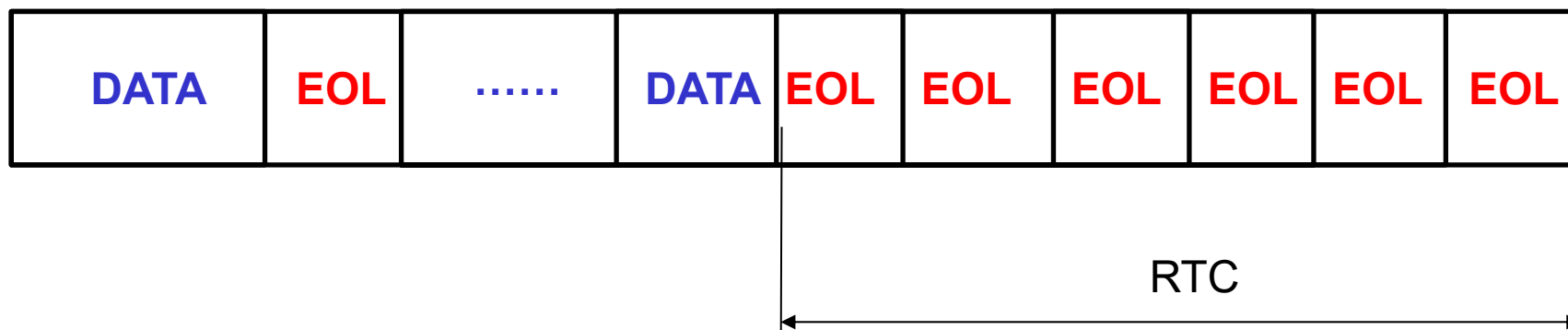
$1650 = 1600 + 50$

- 第一个游程规定为白游程，每行用一个EOL结束码终止。
  - 为了保证收发双方保持同步，数据行起点都用白游程长度开始。
  - 但是如果实际编码是黑游程开始，那么也要先放置一个长度为0 的白游程序字，以此达到同步。
- 每页数据之前加入一个结束码（End of Line, EOL），每页数据尾部连续使用6个结束码。
  - EOL 是一个11个“0”后跟一个“1”即00000 00000 0 1
  - 在每一行编码扫描线结束加入一个EOL；
  - 在每一页第一条扫描线数据之前加入一个EOL；
  - 转回控制规程码RTC：在每一页最后一个扫描线结束加入6个EOL。
- 填充码：为了保证每行数据传输时间大于等于协议约定的最小传输时间，在扫描线数据和EOL间加入可变长度的“0”串。

## ■ 一页报文的开头



## ■ 一页报文的结尾



## 4-6.1 游程编码方法

**【例题】** 假设某一个传真数据中某一行像素点分布如下：

|-73白-|-7黑-|-11白-|-18黑-|-1619白-|

求：1)MH 编码， 2) 编码后总比特数， 3) 压缩率

解：查表知：

73白=64白+9白

1619白=1600白+19白

行结束符 EOL

64白	9白	7黑	11白	18黑	1600白	19白	EOL
11011	10100	00011	01000	0000001000	010011010	0001100	000000000001

编码后码字长度58位，

该行的压缩率为： $1728/58=29.8$



- 假设下面三行扫描线分别是一页传真文件的第一行、第二行及最后一行，请分别进行编码。



686黑 455白 355黑 155白 13黑 5白 14黑 45白




85白 720黑 108白 3黑 64白 6黑 30白 712黑



832白 728黑 68白 64黑 36白



640黑+46黑  
  
 0000001001010  
 +000001010110



64白+ 0白  
  
 11011  
 +00110101



## 4-6.2 LZW编码方法

LZW编码方法 (Lempel-Ziv-Welch)

最早于1977年公布。1985年获专利。



Abraham Lempel



Jacob Ziv

### 核心思想:

编码：用短的编码代替字符串，它不对输入的字符串做任何分析，只是将收到的每一个新字符串添加到一张字符串表中，当已经出现的字符串再次出现时，即采用一个短的编码代替，从而实现了压缩。

译码：即根据算法输出流重建压缩前的输入。LZW解压缩时不需传递字符串表。将输入流作为数据，可以重建编码过程中一样的压缩表。

## 4-6.2 LZW编码方法

LZW算法输出的编码可以是任意长度的，但是必须大于一个字符的编码长度，开始256个编码（假设每一个字符长8bit）默认给标准字符，余下的编码在处理过程中会被分配给其他字符串。

LZW码也称基于字典的编码方法，它是定长码

### (1) 基于字典编码的基本原理

计算机文件是以字节为单位组成的。LZW码是一种自适应编码，它的字典是直接由被压缩文件在编码过程中生成的。

字典基础压缩算法的简单范例：

以标记（**token**）来取代词组（**phrase**）。如果标记的位元数量是少于词组所需的位元数目，那么压缩就如此产生。

## 4-6.2 LZW编码方法

未压缩的文本为：

"**I am dumb** and because **I am dumb**, I can't even tell you that **I am dumb**. "

压缩过的文本：

"**\$1** and because **\$1**, I can't even tell you that **\$1**.  
**\$1**=[**I am dumb**]"

### (2) 字典的构成

字典的容量为4096(0 ~ 4095),序号用12bit表示. 最后一个单词(第4095个单词)为空。

## 4-6.2 LZW编码方法

### (3) 算法

#### 【编码步骤】

①字典初始化

②动态数据初始化：初始化新单词存放位置指针P，将它指向字典的第一个空位置。

③如果文件再没有字符了，输出当前单词W的序号。编码结束。如果文件中还有字符，把当前单词W作为前缀，再从被压缩文件中读入一个字符CH，把CH作为尾字符，得到一个单词 $W_1$ 。

④如果字典中已有 $W_1$ ，则将 $W_1$ 看作当前单词W，返回③。如果字典中没有 $W_1$ （发现一个新单词），先将原单词W的序号输出，再加新单词 $W_1$ ，增加到字典中，然后把刚刚读入的字符CH作为当前单词W，返回③。

## 4-6.2 LZW编码方法

### 编码伪代码示例

STRING = 取得输入字符

**WHILE** 仍然有输入 **DO**

    CHARACTER = 取得输入字符

**IF** STRING+CHARACTER 在字符串表中 **THEN**

        STRING = STRING + CHARACTER

**ELSE**

        输出 STRING 的编码

        将 STRING+CHARACTER 添加到字符串表中

        STRING = CHARACTER

**END of IF**

**END of WHILE**

Output the code for STRING

## 4-6.2 LZW编码方法

### 【译码步骤】

①字典初始化

②动态数据初始化

③如果压缩中已经没有码字，解码结束。否则继续读入一个码字。

④如果读入的码字是无效码字FFF，则解码结束，否则下一步。

⑤如果在字典中已经有该码字对应的单词，则采用递归算法，输出该单词的内容。并将单词的第一个有效字符作为尾字符，将已经记忆的前一个码字作为前缀，组成一个新单词，写入字典中，然后将当前码字记忆下来，返回③；否则，首先在字典中生成新的单词，然后再输出这个单词，将新单词的码字记忆下来，返回③。这时的新单词一定是首尾相同的单词。



## 4-6.2 LZW编码方法

### 译码伪代码示例

读取OLD\_CODE

输出OLD\_CODE译码

**WHILE** 仍然有输入 **DO**

    读取 NEW\_CODE

    STRING = 根据字符串表转换 NEW\_CODE

    输出 STRING

    CHARACTER = STRING 中第一个字符

    将 OLD\_CODE + CHARACTER 添加到字符串表中

    OLD\_CODE = NEW\_CODE

**END of WHILE**

## 4-6.2 LZW编码方法

### 实例：

编码原文：TOBEORNOTTOBEORTOBEORNOT#

#代表信息结束符号，编码为0

A~Z共计26个英文字符进行编码1~26

Symbol	Binary	Decimal	Symbol	Binary	Decimal
#	00000	0	N	01110	14
A	00001	1	O	01111	15
B	00010	2	P	10000	16
C	00011	3	Q	10001	17
D	00100	4	R	10010	18
E	00101	5	S	10011	19
F	00110	6	T	10100	20
G	00111	7	U	10101	21
H	01000	8	V	10110	22
I	01001	9	W	10111	23
J	01010	10	X	11000	24
K	01011	11	Y	11001	25
L	01100	12	Z	11010	26
M	01101	13			

## 4-6.2 LZW编码方法

### 实例：

编码原文：TOBEORNOTTOBEORTOBEORNOTTO#

Current Sequence	Next Char	Output		Extended Dictionary	Comments
		Code	Bits		
NULL	T				
T	O	20	10100	27: TO	27 = first available code after 0 through 26
O	B	15	01111	28: OB	
B	E	2	00010	29: BE	
E	O	5	00101	30: EO	
O	R	15	01111	31: OR	
R	N	18	10010	32: RN	32 requires 6 bits, so for next output use 6 bits
N	O	14	001110	33: NO	
O	T	15	001111	34: OT	
T	T	20	010100	35: TT	
TO	B	27	011011	36: TOB	
BE	O	29	011101	37: BEO	
OR	T	31	011111	38: ORT	
TOB	E	36	100100	39: TOBE	
EO	R	30	011110	40: EOR	
RN	O	32	100000	41: RNO	
OT	#	34	100010		# stops the algorithm; send the cur seq
		0	000000		and the stop code

## 4-6.2 LZW编码方法

### 实例：

编码原文： TOBEORNOTTOBEORTOBEORNOTTO #

编码效果：

源文件长度  $25\text{codes} \times 5\text{bits/code} = 125\text{bits}$

编码后长度  $(6\text{ codes} \times 5\text{ bits/code}) + (11\text{ codes} \times 6\text{ bits/code})$   
 $= 96\text{ bits}$

节约  $125-96=29\text{bits}$

reducing the message by almost 22%

## 4-6.2 LZW编码方法

**实例：**

编码原文： TOBEORNOTTOBEORTOBEORNOT #

Input		Output Sequence	New Dictionary Entry				Comments
Bits	Code		Full		Conjecture		
10100	20	T			27:	T?	
01111	15	O	27:	TO	28:	O?	
00010	2	B	28:	OB	29:	B?	
00101	5	E	29:	BE	30:	E?	
01111	15	O	30:	EO	31:	O?	
10010	18	R	31:	OR	32:	R?	created code 31 (last to fit in 5 bits)
001110	14	N	32:	RN	33:	N?	so start reading input at 6 bits
001111	15	O	33:	NO	34:	O?	
010100	20	T	34:	OT	35:	T?	
011011	27	TO	35:	TT	36:	TO?	
011101	29	BE	36:	TOB	37:	BE?	36 = TO + 1st symbol (B) of
011111	31	OR	37:	BEO	38:	OR?	next coded sequence received (BE)
100100	36	TOB	38:	ORT	39:	TOB?	
011110	30	EO	39:	TOBE	40:	EO?	
100000	32	RN	40:	EOR	41:	RN?	
100010	34	OT	41:	RNO	42:	OT?	
000000	0	#					

## 4-6.2 LZW编码方法

### **(4) 适用文件类型**

不适合小文件的压缩(因为压缩编码初期, 由于字典中的单词很少, 字典对压缩效果的贡献也很少, 主要是进行字典的扩充), 也不适合太大的文件(因字典容量有限, 文件太大时字典满了, 效率将受到制约). 适合内容有明显单词结构的文件(如文本文件、程序文件)。