第5章

无失真信源编码

主要内容

主要内容

5.4 最优编码

- 5.4.1 二元Huffman编码
- 5.4.2 多元Huffman编码
- 5.4.3 Huffman决策树
- 5.4.4 规范Huffman编码
- 5.4.5 马氏源的Huffman编码
- 5.4.6 香农码

5.5 几种实用的信源编码方法

- 5.5.1 算术编码
- 5.5.2 游程编码
- 5.5.3 LZ编码

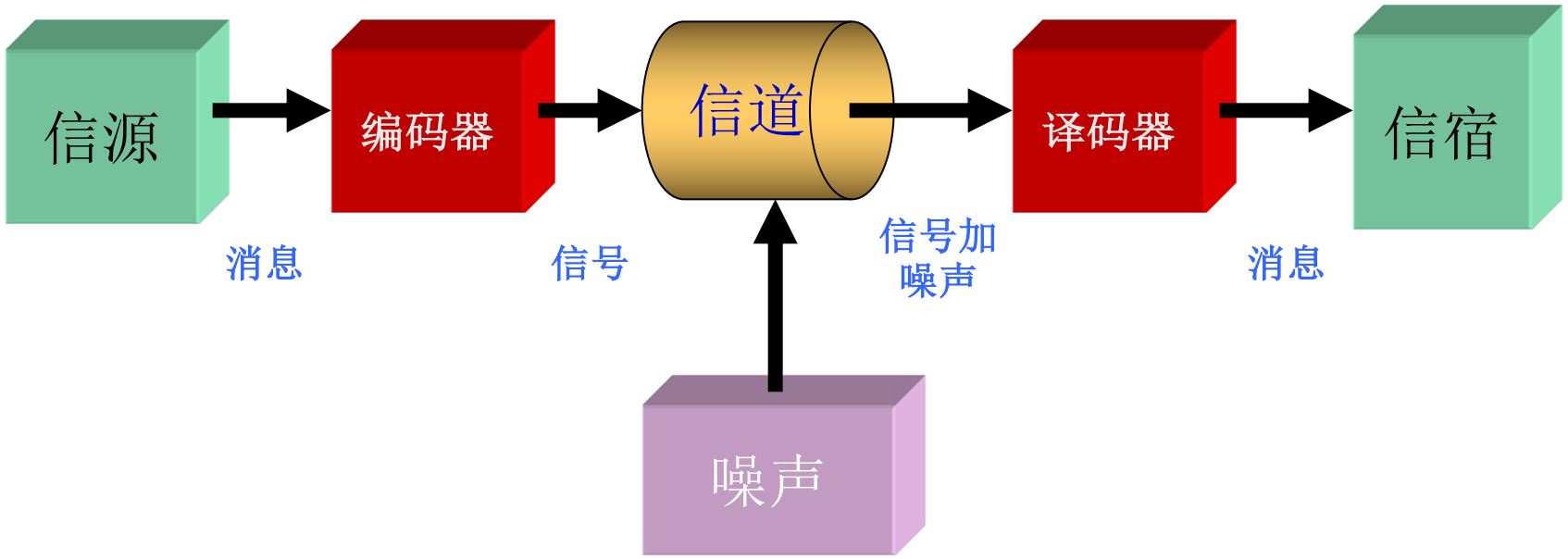
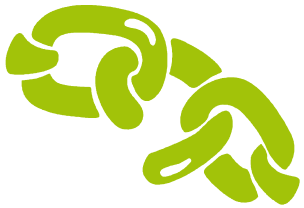
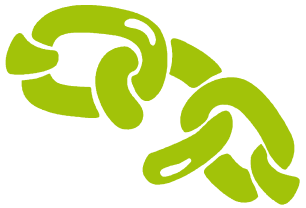


图1.2.1　通信系统模型

图1.2.2　编码器的组成



- 信源编码器的功能是将信源消息变成符号，目的是提高传输有效性，减少剩余度，也就是压缩每个信源符号传输所需代码（通常为二进制代码）的数目（对二进制代码称比特数）。
- 例如，一个信源含4个符号{a,b,c,d}，概率分别为1/2，1/4，1/8，1/8。如果不采用信源编码，每个信源符号至少需要用2个二进制代码传输。如果采用信源编码，分别将a,b,c,d编码成为：0，10，110，111，那么平均每信源符号只需1.75个二进制代码传输。
- 采用合适的信源编码确实能通过压缩码率提高传输有效性。所以，信源编码也称信源压缩编码。



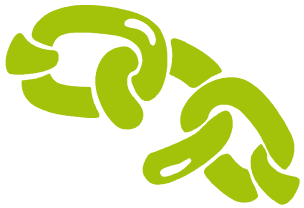
内容简介

- **信源编码**：将信源符号序列按一定的数学规律映射成由码符号组成的码序列的过程。
- **信源译码**：根据码序列恢复信源序列的过程。
- **无失真信源编码**：即信源符号可以通过编码序列无差错地恢复。（适用于离散信源的编码）
- **限失真信源编码**：信源符号不能通过编码序列无差错地恢复。（可以把差错限制在某一个限度内）

本章意义

- 信源编码的目的：提高传输有效性，即用尽可能短的码符号序列来代表信源符号。

- 无失真信源编码定理证明了：如果对信源序列进行编码，当序列长度足够长时 ，存在无失真编码使得传送每信源符号所需的比特数接近信源的熵。因此，采用有效的信源编码会使信息传输效率得到提高。



§5.1 概述

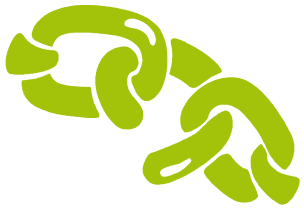
5.1 概述

5.1.1 信源编译码器模型

5.1.2 信源编码的分类

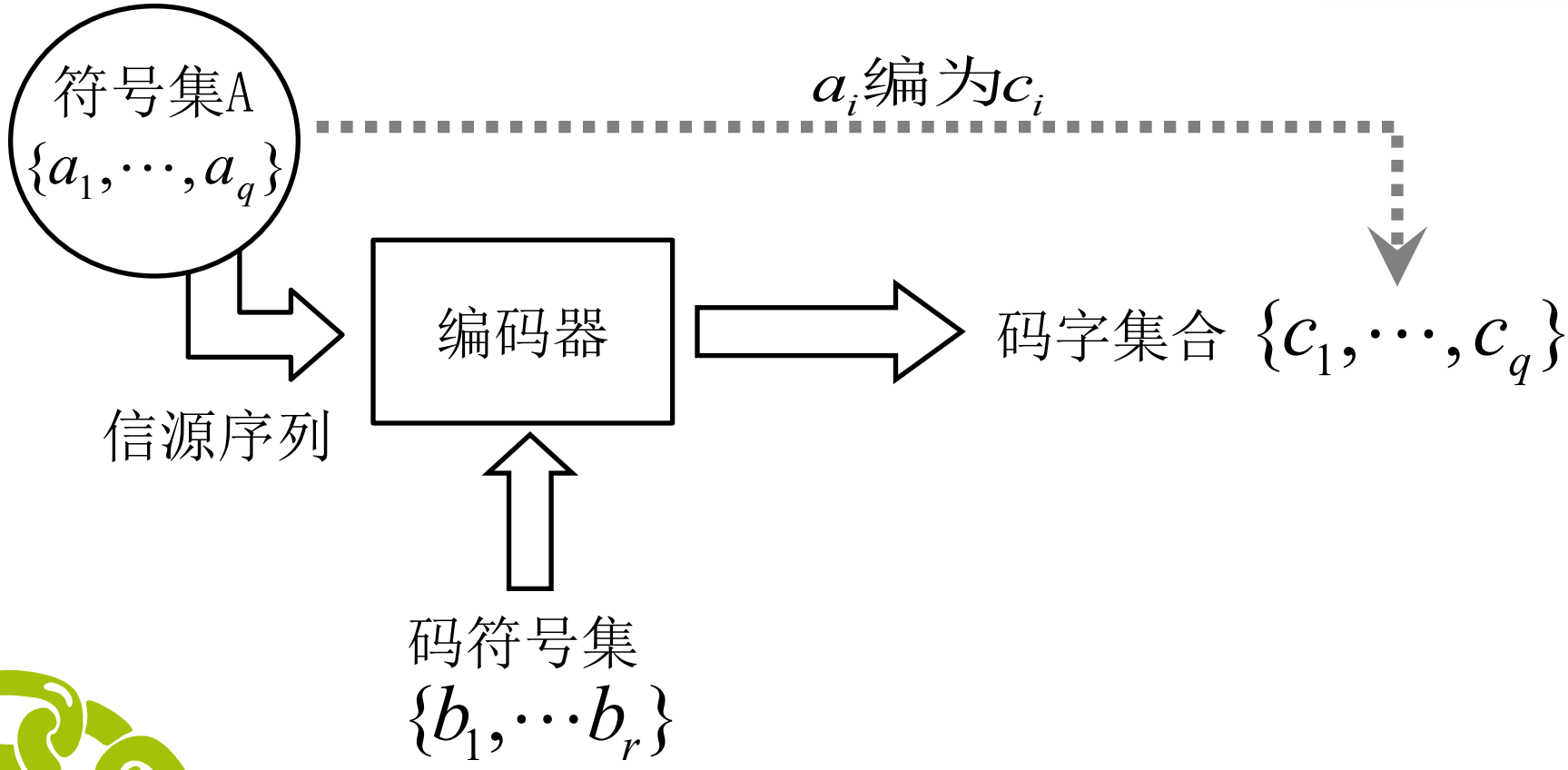
5.1.3 分组码

5.1.4 无损信源编码系统



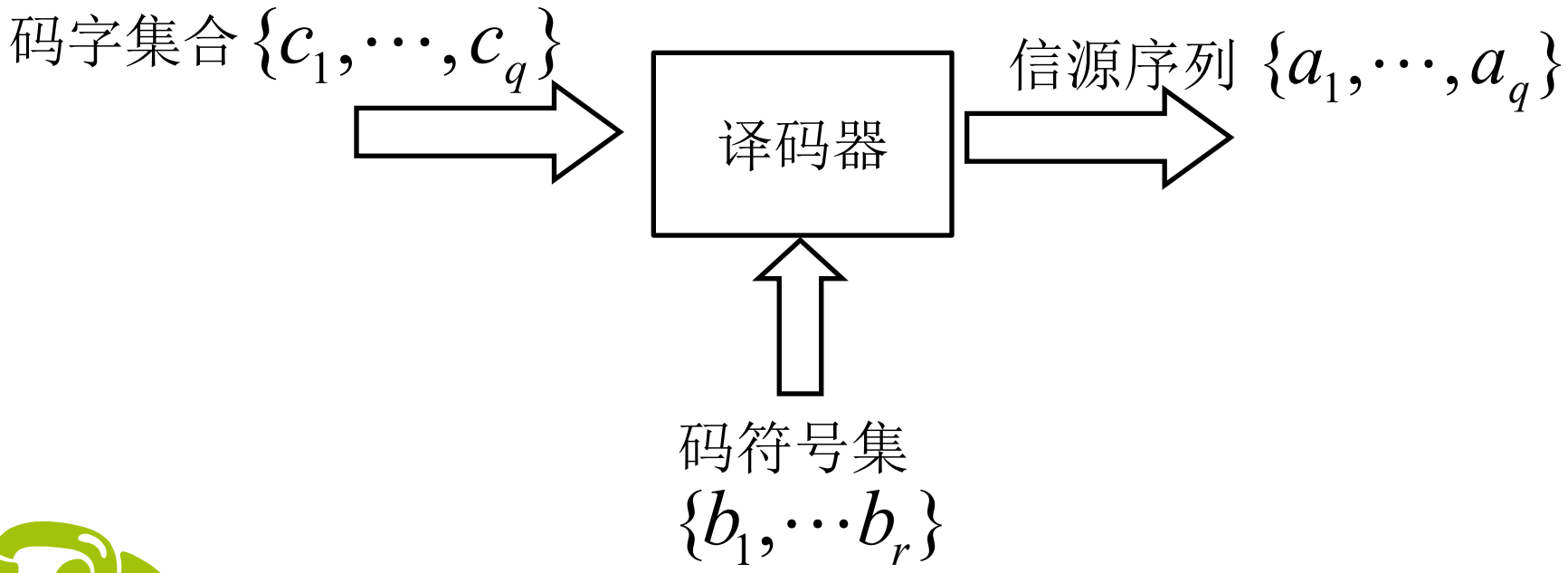
§5.1.1 信源编码器

分组码单符号信源编码器：



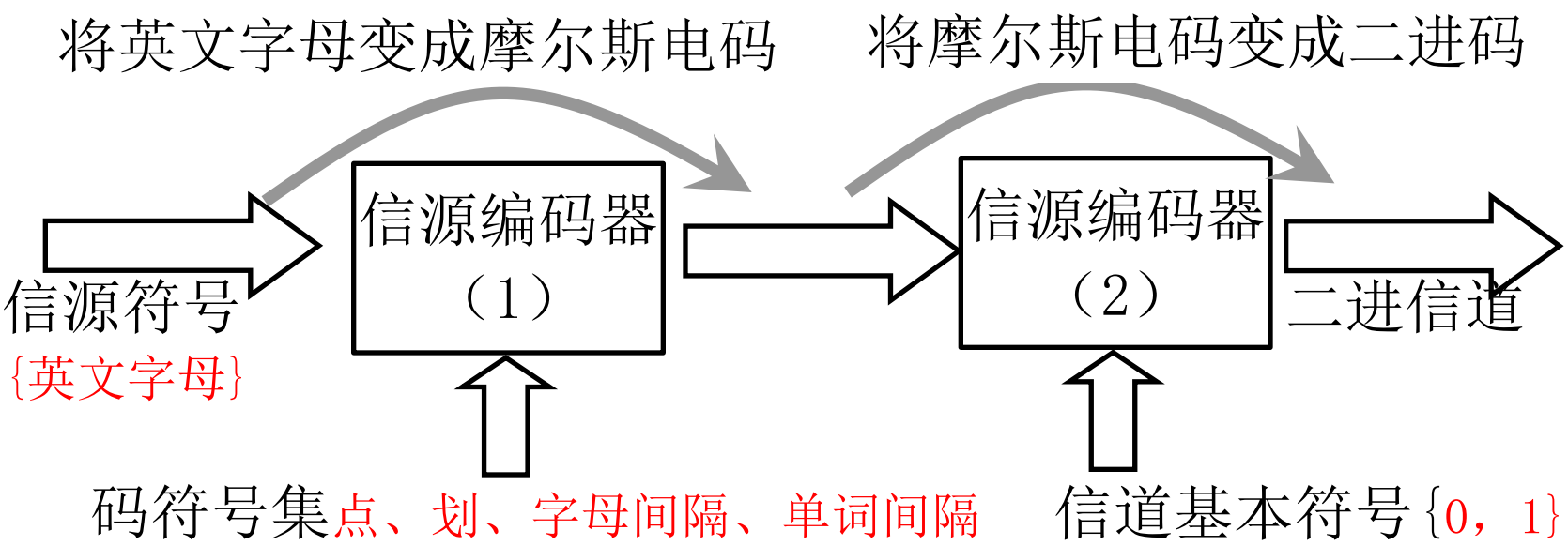
§5.1.1 信源译码器

分组码单符号信源译码器：



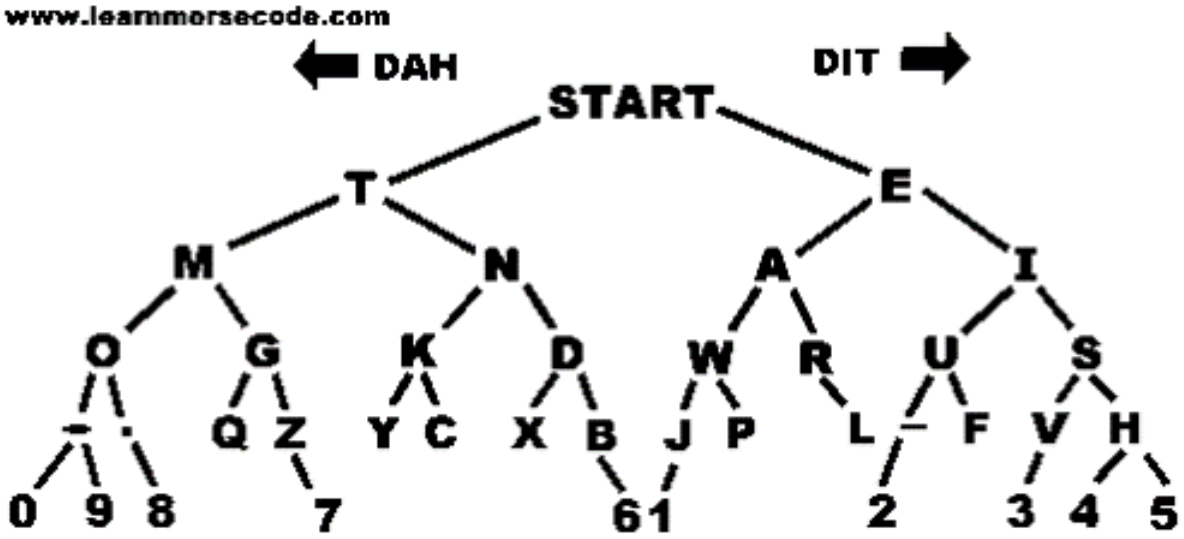
§5.1.1 简单信源编码器

摩尔斯信源编码器：



符号	点	划	字母间隔	单词间隔
电平	+ −	+++ −	− − −	− − − − − −
二进代码	1 0	1110	000	00000

§5.1.1 摩尔斯信源编码器



www.learnmorsecode.com

A ·−	I ··	Q −−·−	Y −·−−
B −···	J ·−−−	R ·−·	Z −−··
C −·−·	K −·−	S ···	Period ·−·−·−
D −··	L ·−··	T −	Comma −−··−−
E ·	M −−	U ··−	? ··−−··
F ··−·	N −·	V ···−	/ −··−·
G −−·	O −−−	W ·−−	@ ·−−·−·
H ····	P ·−−·	X −··−	

1 ·−−−−	
2 ··−−−	
3 ···−−	
4 ····−	
5 ·····	
6 −····	
7 −−···	
8 −−−··	
9 −−−−·	
0 −−−−−	

§5.1.2 信源编码的分类

★**概率匹配编码**：信源符号的概率已知。

- **分组码**：先分组再编码。在分组码中，每一个码字仅与当前输入的信源符号组有关，与其他信源符号无关。

  包括：定长码、变长码（Huffman编码、费诺编码）

- **非分组码**：码序列中的符号与信源序列中的符号无确定的对应关系。例如算术编码。

★**通用编码**：信源符号的概率未知。

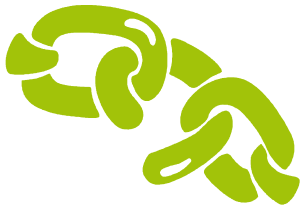
§5.1.2 信源编码的分类

★与非分组码的显著区别：分组码中包含码字

●如果一个码中各码字都不相同，则称为非奇异码；否则称为奇异码。

●如果任何有限长信源序列所对应的码序列都不与其他信源序列所对应的码序列重合，则称为唯一可译码。

要想实现无失真编码，必须要求分组码具有非奇异性和唯一可译性

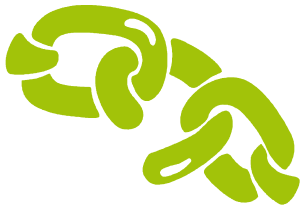


§5.1.2 信源编码的分类

★ 即时码与非即时码

如果在译码过程中只要接收到每个码字最后一个符号就可以立即将该码字译出，这种码称为即时码；否则称为非即时码。

即时码的优点是译码延迟小。



§5.1.3 分组码

★ 异前置码

❖设 $\vec{x}_k$ 为长度为 k 的码字，即 $\vec{x}_k = x_1, \cdots, x_k$，称 $x_1 x_2 \cdots x_j (1 \le j \le k)$ 为 $\vec{x}_k$ 的前置。

❖异前置码是唯一可译码

❖一个码中无任何码字是其他码字的前置

❖ 异前置码与即时码是等价的

★ 逗号码

❖ 用一个特定的码符号表示所有码字的结尾

❖ 逗号码是唯一可译码

§5.1.3 分组码

例5.1 设信源符号集为{a, b, c, d}，采用6种分组编码如下表，分析每一个码的唯一可译性

符号	码A	码B	码C	码D	码E	码F
a	0	0	00	0	1	0
b	0	1	01	10	01	01
c	1	10	10	110	001	011
d	10	11	11	111	0001	0111

即时码

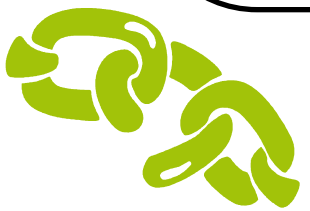
10 → c, ba

	等长	异前置码	逗号码	0表示开头
非奇异	✓	✓	✓	✓
唯一可译	✓	✓	✓	✓

§5.1.3 分组码

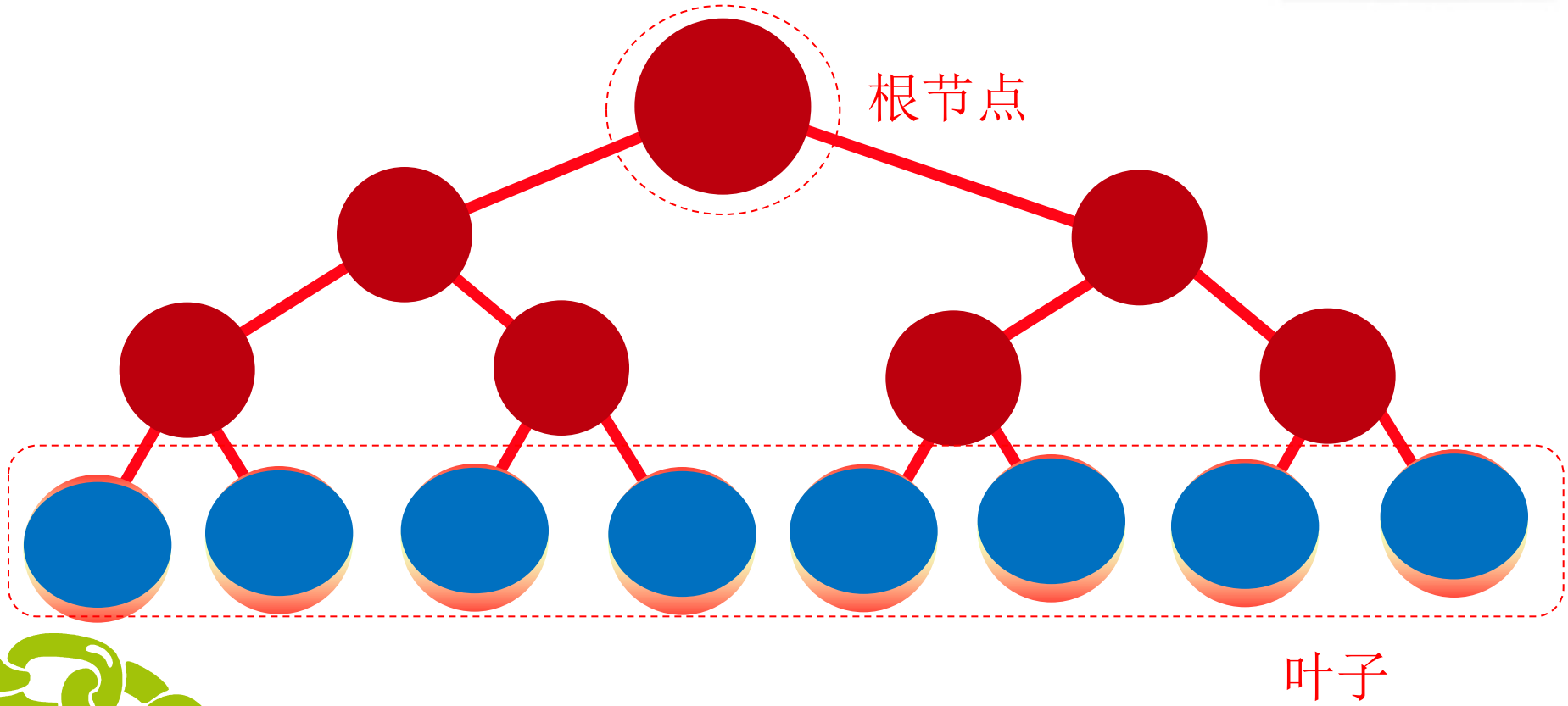
一些结论：

变长码		定长码
非奇异且异前置就唯一可译	→	只要非奇异，就唯一可译
速率变化→设置缓冲器	→	速率恒定→不需缓冲器
受误码影响大，逗号码除外	→	码长已知→容易同步
容易产生差错传播	→	无差错传播



§5.1.3 分组码

★ 码树是表示信源编码码字的重要工具之一

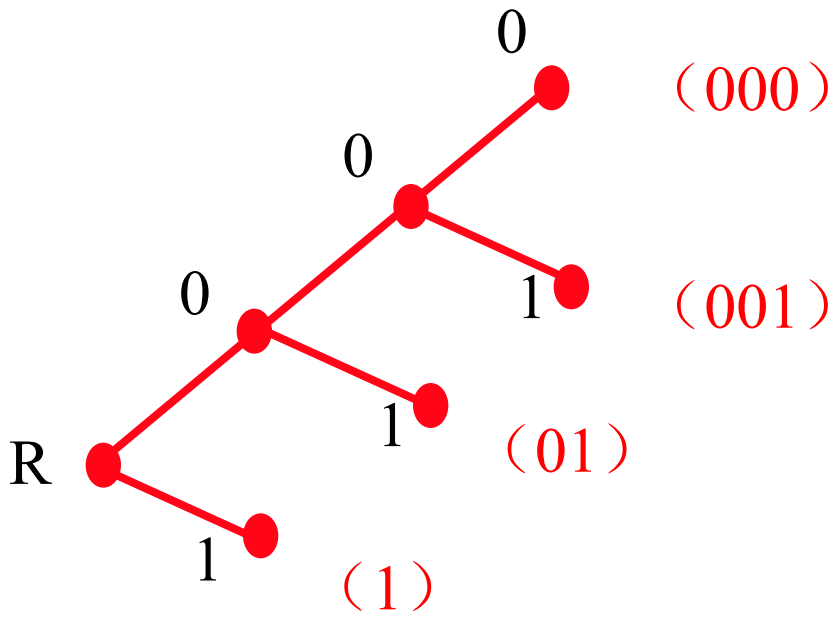


§5.1.3 分组码

例5.2

一个码C包含4个码字：{1, 01, 000, 001}， 试用码树来表示。

解：



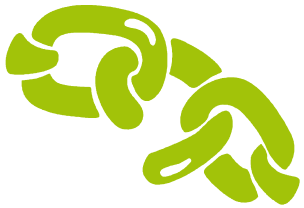
§5.1.3 分组码

一些结论：

★ 在码树中，n 阶节点的个数最多为 r^n

例：2进码树中，n 阶节点数目最多为 2^n

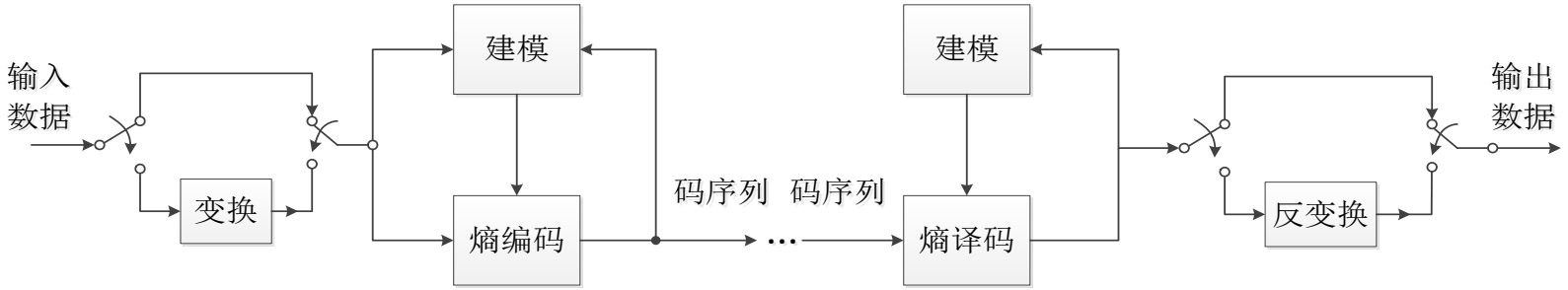
★ 非奇异码总能与码树建立一一对应的关系



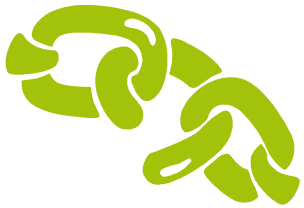
§5.1.4 无损信源编码

简介：

★ 在通用编码系统中，编译码器需要同步，即译码器要利用与编码器同样的信息建模。所以在编码器中要利用已经处理过的数据进行概率估计，这样才能使得译码器可以利用同样的信息建模



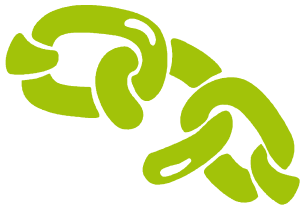
（a）编码器　　（b）译码器



§5.2 定长码

5.2 定长码

- 5.2.1 无失真信源编码条件
- 5.2.2 渐进均分特性
- 5.2.3 定长码信源编码定理



§5.2.1 无失真编码条件

★ 对于定长码，只要非奇异就唯一可译。这就要求码字的数目不少于被编码的信源序列的个数

★ 单信源符号编码：

❖设信源X包含n个符号，码符号集包含的符号数为r

$$n \le r^{l}$$

码长

式(5.1)

★ ❖ N长信源符号序列编码（N次扩展码）

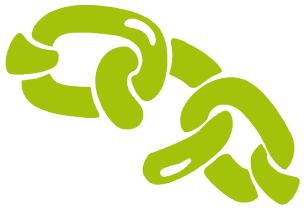
$$n^{N} \le r^{l} \text{ 或 } \frac{l}{N} \ge \frac{\log n}{\log r}$$

式(5.2)

平均每个信源符号所需码符号数

§5.2.1 无失真编码条件

例：英文字母26个加1个空格可看成共27个符号的信源。 如对单符号进行编码：



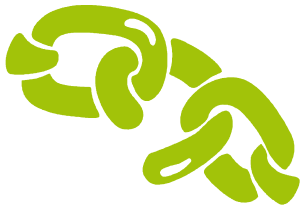
§5.2.1 无失真编码条件

解：

$$27 \le 2^l \Rightarrow l \ge \log 27 = 4.755, \quad 取 l \ge 5$$

注：

但是，如果采用适当的信源编码，理论上每信源符号所需二进码符号数可以远小于上面的值， 在理想情况下可以压缩到接近信源的熵1.4左右。本节就是从理论上证明这种压缩是可以实现的。

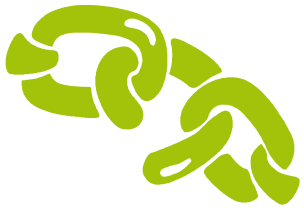


§5.2.2 渐近均分特性

定理5.1

设 X 为一离散无记忆信源，那么对任意给定 $\varepsilon>0,\delta>0,$ 总能找到一个正整数 N_0，使得任何长度为 $N\geq N_0$ 的信源序列都可以分成两组，且第一组中序列x出现的概率 $p(x)$ 满足：

$$\left|\frac{1}{N}\log p(\vec{x})+H(X)\right|<\delta \qquad (式5.3)$$



§5.2.2 渐进均分特性

证：

我们先证明（式5.3）。这里假设信源符号集为$A=\{a_1,a_2,\cdots a_q\}$，各符号出现的概率分别为p_i，$\vec{x}=x_1x_2\cdots x_N$为长度为N的序列，N_i为$\vec{x}$中符号a_i出现的次数。将信源序列按下列原则分成两：G_1、G_2，其中：

$$G_1 \quad \{\vec{x}: \left|\frac{N_i}{N}-p_i\right|<\varsigma, \quad i=1,\cdots,q\} \qquad (式5.4)$$

$$G_2 \quad \{\vec{x}: \quad 其它\}$$

根据大数定律，当序列足够长时，信源符号a_i出现的次数接近Np_i。因此，G_1中的序列的符号出现的次数符合大数定律，称典型序列。

§5.2.2 渐进均分特性

证：

从式（5.4）中可以看出，G_1随ς的不同而改变。

设$\vec{x} \in G_1$，则对于$\vec{x}$中的信源符号a_i，有

$$-\varsigma < \frac{N_i}{N} - p_i < \varsigma, \quad i = 1, \cdots q$$

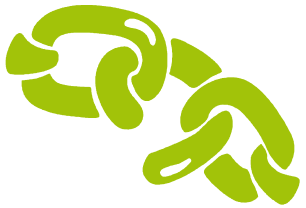
或 $$\frac{N_i}{N} = p_i + \theta_i \varsigma \quad \text{其中} \quad |\theta_i| < 1$$

由于信源是无记忆的，所以$\vec{x}$的概率为$p(\vec{x}) = p_1^{N_1} \cdots p_q^{N_q}$

$\vec{x}$的自信息负值为：

$$\log p(\vec{x}) = \sum_{i=1}^{q} N_i \log p_i = \sum_{i=1}^{q} N(p_i + \theta_i \varsigma) \log p_i$$

$$= -NH(X) + N\varsigma \sum_{i=1}^{q} \theta_i \log p_i$$



§5.2.2 渐进均分特性

证：

所以 $$\frac{\log p(\vec{x})}{N}+H(X)=\varsigma\sum_{i=1}^{q}\theta_i\log p_i$$

$$\left|\frac{\log p(\vec{x})}{N}+H(X)\right|=\varsigma\left|\sum_{i=1}^{q}\theta_i\log p_i\right|\prec\varsigma\sum_{i=1}^{q}\left|\log p_i\right|$$

选择 ς，使得

$$\varsigma=\frac{\delta}{\sum_{i=1}^{q}\left|\log p_i\right|}$$ （式5.5）

则（式5.3）成立。

§5.2.2 渐进均分特性

证：

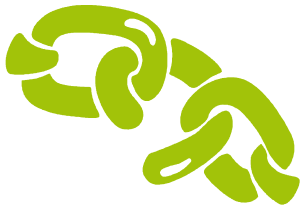
下面证明定理的后半部分。设 $\vec{x} \in G_2$，根据（式5.3），有

$$\left|\frac{\log p(\vec{x})}{N} + H(X)\right| \geq \delta \qquad \text{（式5.6）}$$

因为信源是无记忆，所以，$p(\vec{x}) = p(x_1)\cdots p(x_N)$

得到

$$\log p(\vec{x}) = \sum_{i=1}^{N} \log p(x_i) \qquad \text{（式5.7）}$$



§5.2.2 渐进均分特性

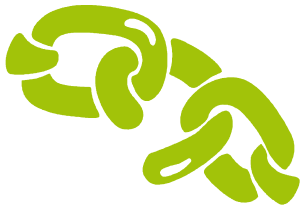
证：

将（式5.5 ）代入（式5.6），得

$$\left|\frac{1}{N}\sum_{i=1}^{N}\log p(x_i)+H(X)\right|\geq\delta$$

令 $z_i=\log p(x_i)$ 可得 $E(z_i)=-H(X)$，所以

$$E\left\{\frac{1}{N}\sum_{i=1}^{N}\log p(x_i)\right\}=\frac{1}{N}\sum_{i=1}^{N}E(z_i)=-H(X)\qquad\text{（式5.8）}$$



§5.2.2 渐进均分特性

证：

根据Chebyshev不等式：$p\left\{\left|\xi-\bar{\xi}\right|>\delta\right\}\leq\frac{Var(\xi)}{\delta^2}$，其中$\xi$为随机变量；这样就得到：

其中 $p_r\left\{\vec{z}:\left|\frac{1}{N}\sum_{i=1}^{N}z_i-\bar{z}\right|\geq\delta\right\}\leq\frac{\sigma^2}{N\delta^2}$

$$\vec{z}=(z_1,z_2,\cdots z_N),\ \bar{z}=E(\frac{1}{N}\sum_{i=1}^{N}z_i),\qquad \sigma^2=Var(z_i)$$

所以 $p_r\left\{\vec{x}:\left|\frac{\log p(\vec{x})}{N}+H(X)\right|\geq\delta\right\}\leq\frac{\sigma^2}{N\delta^2}$ （式5.11）

§5.2.2 渐进均分特性

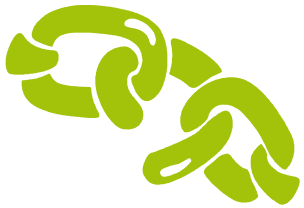
证：

其中，自信息的方差

$$\sigma^2 = Var[\log p(x_i)]$$

$$= E\left[\log^2 p(x_i)\right] - H^2(X) = \sum_{i=1}^{q} p_i \log^2 p_i - H^2(X)$$（式5.10）

取 $\frac{\sigma^2}{N_0\delta^2} = \varepsilon$，则当 $N > N_0$时，有 $\frac{\sigma^2}{N\delta^2} < \frac{\sigma^2}{N_0\delta^2} = \varepsilon$



§5.2.2 渐进均分特性

★ 总结

★对离散无记忆信源，给定 $\varepsilon,\delta>0$，令 $N_0=\frac{\sigma^2}{\varepsilon\delta^2}$（式5.12）取$N\geq N_0$；那么对长度为N的信源序列，满足下式的为典型序列，否则为非典型序列。

$$\{\vec{x}:\left|\frac{N_i}{N}-p_i\right|<\varsigma,\ i=1,\cdots,q\}$$

★ 定理说明，当N足够大时，典型序列 $\vec{x}$的 $\frac{-\log p(\vec{x})}{N}$ 的值接近信源的熵。

★对于有记忆的马氏源，定理5.1也成立

§5.2.2 渐进均分特性

★ 典型序列的概率估计 $\left|\frac{1}{N}\log p(\vec{x})+H(X)\right|<\delta$

$$设\vec{x}\in G_1 \Rightarrow -\delta<\frac{\log p(\vec{x})}{N}+H(X)<\delta$$

$$\Rightarrow -N[H(X)+\delta]<\log p(\vec{x})<-N[H(X)-\delta]$$

设取2为底 $\Rightarrow 2^{-N[H(X)+\delta]}<p(\vec{x})<2^{-N[H(X)-\delta]}$

简记为: $$p(\vec{x})=2^{-N[H(X)\pm\delta]}$$ （式5.13）

❖当 δ 足够小时，每个典型序列的概率 $p(\vec{x})$ 接近 $2^{-NH(X)}$，其偏差不大于 $2^{N\delta}$;

❖此时序列的长度需要很大

§5.2.2 渐进均分特性

★ 典型序列的个数估计

设N_G为 G_1 中序列的个数

先估计上界：

利用概率估计的下界 $\Rightarrow N_G \cdot 2^{-N[H(X)+\delta]} < N_G \cdot \min\limits_{\vec{x}} p(\vec{x}) \le 1$

$$N_G < 2^{N(H(X)+\delta)} \qquad \text{(式5.14)}$$

再估计下界：

利用概率估计的上界 $\Rightarrow 1-\varepsilon \le N_G \cdot \max\limits_{\vec{x}} p(\vec{x}) < N_G \cdot 2^{-N[H(X)-\delta]}$

$$N_G > (1-\varepsilon)2^{N[H(X)-\delta]} \qquad \text{(式5.15)}$$

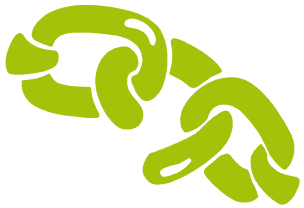
$$(1-\varepsilon)2^{N[H(X)-\delta]} < N_G < 2^{N[H(X)+\delta]} \qquad \text{(式5.16)}$$

§5.2.2 渐进均分特性

定理5.2 平稳无记忆信源AEP

★ 设离散无记忆信源X，当长度N足够大时：

- 典型序列接近等概率 $2^{-NH(X)}$，数目近似于 $2^{NH(x)}$
- 非典型序列出现的概率接近为零
- $-\frac{1}{N}\log p(\vec{x}) \to H(X)$ （以概率收敛） （式5.17a）



§5.2.2 渐进均分特性

定理5.3 平稳遍历信源AEP

★ 设离散遍历信源X，当长度N足够大时：

- 典型序列接近等概率$2^{-NH_\infty(x)}$，数目近似于 $2^{N_\infty H(x)}$
- 非典型序列出现的概率接近为零
- $-\frac{1}{N}\log p(\vec{x}) \to H_\infty(X)$ （以概率收敛） （式5.17b）

定理5.2是定理5.3的一种特例：
无记忆信源熵率就等于单符号信源熵。

§5.2.2 渐进均分特性

总结

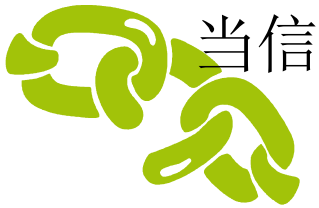
★设信源序列数为 n^N，编码序列数为 r^l：

如果每个信源序列都至少要有一个码字，即需要 $r^l \geq n^N$。

★随着信源序列长度的增加，基本上是典型序列出现，这样我们仅考虑对典型序列的编码：

所以实际需要 $r^l \geq 2^{NH_\infty(X)}$ 个码字。

当信源的熵小于 $\log_2 n$ 时，就会使得码字的长度 l 减小。



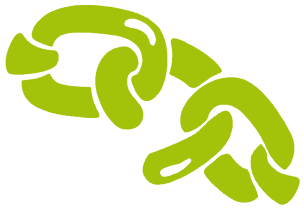
§5.2.3 定长码信源编码定理

定理5.4

离散无记忆信源的熵为H(X)，码符号集的符号数为r将长度为N的信源序列编成长度 l的码序列。只要满足：

$$\frac{l}{N}\log r \geq H(X)+\delta \qquad \text{(式5.18)}$$

则当N足够大时，译码差错可以任意小（$<\varepsilon$）；若上述不等式不满足，肯定会出现译码差错。



§5.2.3 定长码信源编码定理

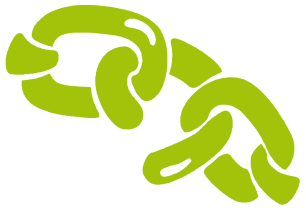
证明思路

★ 正定理

典型序列的个数小于 $2^{N[H(X)+\delta]}$

$$\frac{l}{N}\log r \geq H(X)+\delta \Rightarrow r^{l} \geq 2^{N[H(X)+\delta]}$$

★ 在编码时，可以使所有典型序列都有对应的码字，而最坏的情况是所有的非典型序列无对应的码字。



§5.2.3 定长码信源编码定理

证明思路

★ 逆定理　　若不满足上式 ⇒ $\frac{l}{N}\log r < H(X)$

$$I(X^N;Y^l) = H(X^N) - H(X^N/Y^l) \le H(Y^l) \le lH(Y) = l\log r$$

$$H(X^N) = NH(X)$$

$$H(X^N/Y^l) \ge NH(X) - l\log r > 0$$

★ 已知编码序列条件下信源序列的不确定性，如果无差错译码，该不确定性为零。

§5.2.3 定长码信源编码定理

相关定义

★ 定长码编码速率(简称码率)

$$R=\frac{l\log r}{N}\quad(\text{比特}/\text{信源符号})\qquad(式5.19)$$

- 它表示编码后，一个信源符号平均所携带的最大信息量，也可以理解为传送一个信源符号平均所需的比特数。
- 压缩码率实际就是减小编码速率。

§5.2.3 定长码信源编码定理

相关定义

★ 编码效率

$$\eta = \frac{H(X)}{R} = \frac{NH(X)}{l\log r} \qquad \text{(式5.20)}$$

NH(X)表示N长信源序列的所包含的信息量

llogr表示码序列所能携带的最大信息量。

当N足够大时，η 可以接近1

由渐近均分特性，当 R减小时，η 增加。

压缩码率和提高编码效率是同样的含义。

§5.2.3 定长码信源编码定理

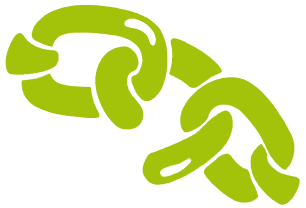
相关定义

★ 信息传输速率：每个传输符号所含信息量

$$R_c = \frac{NH(X)}{l} \quad \text{(比特/码符号)} \qquad \text{(式5.21)}$$

$$\eta = \frac{R_c}{\log r} \qquad \text{(式5.22)}$$

❖ 对于二进编码，编码效率与信息传输速率数值相同。



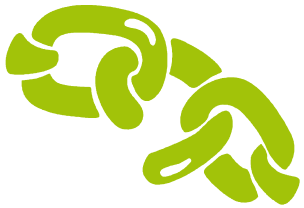
§5.2.3 定长码信源编码定理

结论

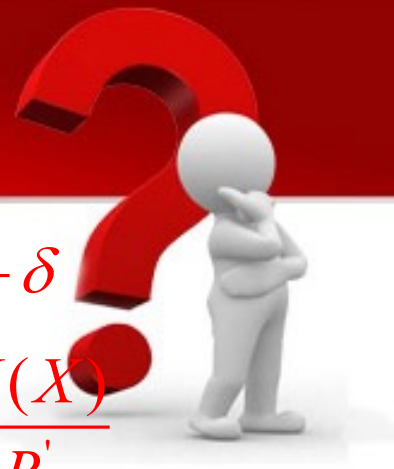
★ 无失真信源编码的另一种表述

如果编码速率 $R > H(X)$，则存在无失真编码。

反之，肯定有失真。



§5.2.3 定长码信源编码定理



结论

$$\frac{l}{N}\log r \geq H(X)+\delta$$

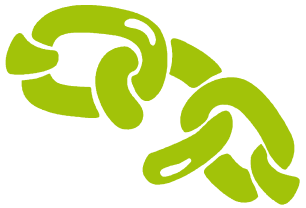
$$R' = \frac{l\log r}{N}, \eta = \frac{H(X)}{R'}$$

★ 信源序列长度与编码效率关系

$$\eta = \frac{H(X)}{H(X)+\delta} \Rightarrow \eta\,\delta + \eta\,H(X) = H(X) \qquad \Rightarrow \delta = \frac{1-\eta}{\eta}H(X)$$

$$N \geq \frac{\sigma^2}{\varepsilon\delta^2} = \left(\frac{\eta}{1-\eta}\right)^2 \cdot \frac{\sigma^2}{H^2(X)\varepsilon} \qquad \text{(式5.23)}$$

信源给定后，若要求编码效率越高，N 越大，
要求译码差错越低，N值也越大。



§5.2.3 定长码信源编码定理

例5.3　一离散无记忆信源的模型如下，要求用二元定长码编码，已知$\eta = 0.96$，$\varepsilon \le 10^{-5}$ 试估计信源序列的最小长度N。

$$\begin{bmatrix} S \\ P(s) \end{bmatrix} = \begin{bmatrix} s_1 & s_2 \\ 3/4 & 1/4 \end{bmatrix}$$

§5.2.3 定长码信源编码定理

解：

信源的熵

$$H(S)=-\frac{3}{4}\log\frac{3}{4}-\frac{1}{4}\log\frac{1}{4}=0.811$$

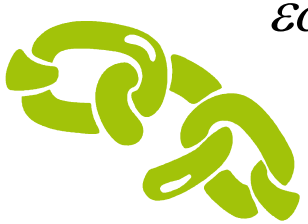
自信息方差

$$\sigma^2=\frac{3}{4}\log^2\frac{3}{4}+\frac{1}{4}\log^2\frac{1}{4}-0.811^2=0.4715$$

$$N\geq\frac{\sigma^2}{\varepsilon\delta^2}=\left(\frac{\eta}{1-\eta}\right)^2\cdot\frac{\sigma^2}{H^2(X)\varepsilon}$$

$$\Rightarrow N\geq\frac{0.96^2\times0.4715}{(1-0.96)^2\times0.811^2\times10^{-5}}=4.13\times10^7$$

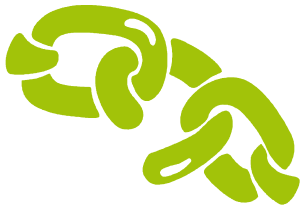
不现实：编码时延大，信源要求长



§5.2.3 定长码信源编码序列

结论

要达到一定误码要求，信源序列长度需很长，所以编码器难于实现。

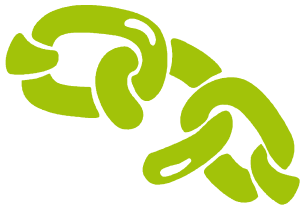


§5.3 变长码

5.3 变长码

5.3.1 异前置码的性质

5.3.2 变长码信源编码定理



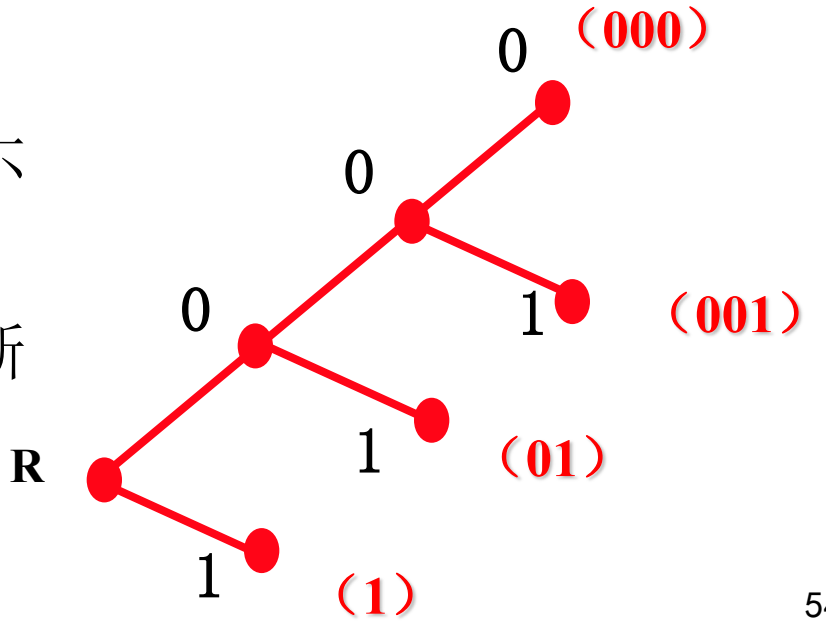
§5.3.1 异前置码性质

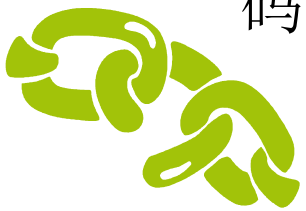
★变长码可用非全码树来描述。下图就是一个异前置码的码树。

全码树图中每个叶子节点都在最底层，从左至右排列

★只有端点（树叶）对应码字。

- ❖ 对应码字的端点与根之间不能有其它的节点作为码字
- ❖ 端点不能向上延伸再构成新码字



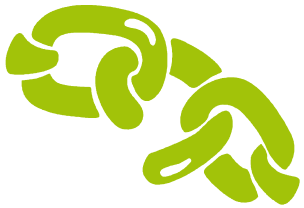


§5.3.1 异前置码性质

定理5.5 （Kraft定理）

★ 若信源符号数为n，码符号数为r，对信源符号进行编码，相应码长度为 $l_1, l_2, \cdots, l_n$，则异前置码存在的充要条件是：

$$\sum_{i=1}^{n} r^{-l_i} \le 1 \quad \text{(Kraft不等式)} \qquad \text{（式5.24）}$$



§5.3.1 异前置码性质

证

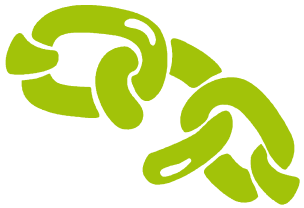
★充分性： 做一个l_M阶全树，树叶总数 r^{l_M}

取l_1阶的任一节点作为第一个码字，去掉的树叶

$$r^{l_M-l_1} = r^{l_M} / r^{l_1}$$

$$r^{l_M-l_1} + r^{l_M-l_2} + \ldots + r^{l_M-l_n}$$

$$= r^{l_M}(r^{-l_1} + \ldots + r^{-l_n}) = r^{l_M}\sum_{i=1}^{n} r^{-l_i} \le r^{l_M}$$



§5.3.1 异前置码性质

证

★ 必要性：

构造一个码全树，最高阶为码字最大长度 l_M
对于阶为 l_k 的节点，占用的树叶数为 $r^{l_M - l_K}$

$$\sum_{K=1}^{n} r^{l_M - l_K} = r^{l_M} \sum_{K=1}^{n} r^{-l_K} \le r^{l_M}$$

❖当码满足 Kraft 不等式时，未必就是异前置码

❖异前置码并不唯一，例如 0, 1 交换。

§5.3.1 异前置码性质

例5.4

下表列出了3种变长码的编码，并给出了对应每个码的所有的码长 和具有同一码长的码字的个数，其中码符号集为{0,1,2,3}。试问对每个码是否存在相应的异前置码？

码字 码 个数 码长	码1	码2	码3
1	3	2	1
2	3	7	7
3	3	3	3
4	3	3	7
5	4	5	4

§5.3.1 异前置码性质

解：

利用 Kraft 不等式来验证。

对于码1：$3\times4^{-1}+3\times4^{-2}+3\times4^{-3}+3\times4^{-4}+4\times4^{-5}$

$$=3\left[\frac{1}{4}+(\frac{1}{4})^2+(\frac{1}{4})^3+(\frac{1}{4})^4+(\frac{1}{4})^5\right]+(\frac{1}{4})^5$$

$$=1$$

实际上，可以用码树来验证，方法更简单。

$\Rightarrow$ 存在相应的异前置码 注意：只是存在异前置码！

同理： 码2不存在相应的异前置码；码3存在相应的异前置码。

§5.3.1 异前置码性质

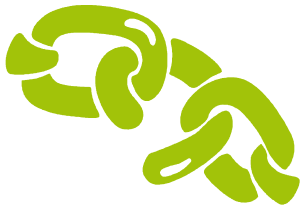
定理5.6

★ 若一个码是唯一可译码且码字长为 $l_1, l_2, \cdots, l_n$ 则必满足Kraft不等式，即：

$$\sum_{i=1}^{n} r^{-l_i} \leq 1$$

n：信源符号数

r：码符号数



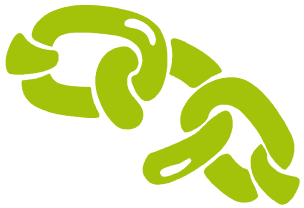
§5.3.1 异前置码性质

结论

★ 满足kraft不等式并不一定唯一可译，因为异码可能满足kraft不等式。

推论5.1

★ 任意唯一可译码都可用异前置码代替，而不改变码字的任一长度。



§5.3.2 变长码信源编码定理

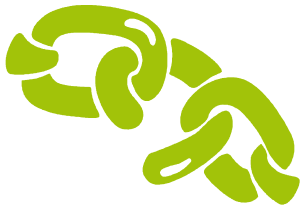
★ 单信源符号编码的平均码长：

$$\bar{l}=\sum_{k=1}^{n}p_k l_k$$

表示平均每个信源符号所需码符号的个数

★ 对于N次扩展源编码，原信源符号平均码长为

$$\bar{l}=\frac{1}{N}\sum_{k=1}^{n^N}p_k l_k$$



§5.3.2 变长码信源编码定理

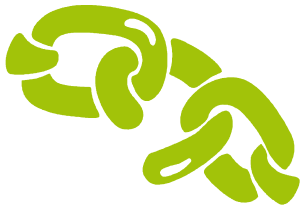
★ 单信源符号编码的平均码长：

$$\bar{l}=\sum_{k=1}^{n}p_k l_k \qquad (式5.25)$$

表示平均每个信源符号所需码符号的个数

★ 对于N次扩展源编码，原信源符号平均码长为

$$\bar{l}=\frac{1}{N}\sum_{k=1}^{n^N}p_k l_k \qquad (式5.26)$$

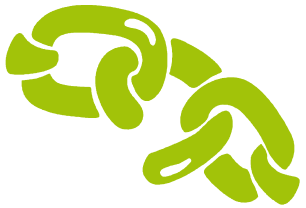


§5.3.2 变长码信源编码定理

定理5.7 单符号信源变长码编码定理

★ 给定熵为H(X)的离散无记忆信源X，用r元码符号集对单信源符号进行编码， 则存在唯一可译码，其平均码长满足：

$$\frac{H(X)}{\log r} \le \bar{l} < \frac{H(X)}{\log r} + 1 \qquad \text{(式5.27)}$$



§5.3.2 变长码信源编码定理

证：

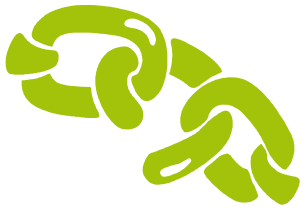
(1) 证明不等式前半部

$$H(X)-\bar{l}\log r=-\sum_i p_i\log p_i-\sum_i p_i l_i\log r$$

$$=\sum_i p_i\log\frac{1}{p_i r^{l_i}}\le\sum_i p_i\left(\frac{1}{p_i r^{l_i}}-1\right)\log e$$

$$=(\log e)\left(\sum_{i=1}^{q} r^{-l_i}-\sum_i p_i\right)\le 0$$

等式成立条件 $\dfrac{1}{p_i r^{l_i}}-1=0$ 即 $p_i=r^{-l_i}$ (式5.28)



§5.3.2 变长码信源编码定理

证：

(2) 证明不等式后半部

$$\frac{1}{r^{l_i}} \le p_i < \frac{1}{r^{l_i-1}} \qquad l_i = \lceil \log_r(1/p_i) \rceil \qquad \text{（式5.29）}$$

$$\Rightarrow \begin{cases} l_i \ge \log_r(1/p_i) \Rightarrow p_i \ge \dfrac{1}{r^{l_i}} \\ l_i - 1 < \log_r(1/p_i) \Rightarrow p_i < \dfrac{1}{r^{l_i-1}} \end{cases} \qquad \text{（式5.30）}$$

可以验证码长满足Kraft不等式，存在唯一可译码

§5.3.2 变长码信源编码定理

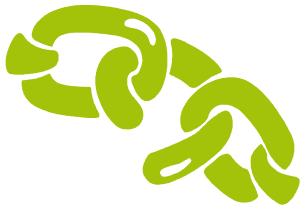
证：

(2) 证明不等式后半部

$$\sum_{i=1}^{q} p_i \log p_i < \sum_{i=1}^{q} p_i \log \frac{1}{r^{l_i-1}}$$

$$= (\log r)(\sum_{i=1}^{q} p_i - \sum_{i=1}^{q} p_i l_i) = (1-\bar{l})\log r$$

$$-H(X) < (1-\bar{l})\log r \Rightarrow \bar{l} < \frac{H(X)}{\log r} + 1$$



§5.3.2 变长码信源编码定理

定理5.8 有限序列信源变长码编码定理

★ 若对长度为N的离散无记忆信源序列进行编码，则存在唯一可译码，且使每信源符号平均码长满足：

$$\frac{H(X)}{\log r} \le \bar{l} < \frac{H(X)}{\log r} + \frac{1}{N} \qquad \text{(式5.31)}$$

且对任何唯一可译码左边不等式都要满足。

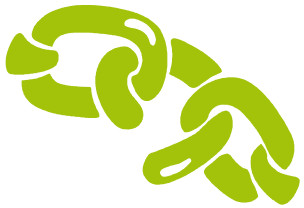
证明：长度为N的扩展源，熵和平均码长均为扩展前的N倍

§5.3.2 变长码信源编码定理

定理5.9

★ 对于离散平稳遍历马氏源，有：

$$\frac{H_\infty(X)}{\log r} \le \bar{l} < \frac{H_\infty(X)}{\log r} + \frac{1}{N} \qquad \text{(式5.32)}$$



§5.3.2 变长码信源编码定理

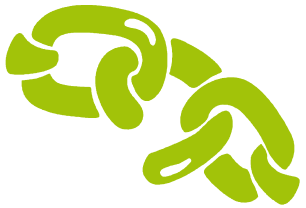
定理5.10

★ 若对任意信源X的N次扩展源 X^N 进行编码，当N足够大时，总能找到唯一可译的r进编码，使得X的平均码长任意接近信源的熵 $H_r(X)$

$$\bar{l} \to H_r(X) = \frac{H_\infty(X)}{\log r} \qquad \text{(式5.33)}$$

证明思路：

利用定理(5.9)的不等式，就可得到定理的结果



§5.3.2 变长码信源编码定理

相关定义

★ 编码速率：

$$R = \bar{l} \log r \qquad \text{(式5.34)}$$

★ 编码效率：

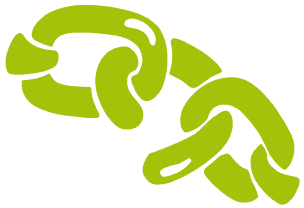
$$\eta = \frac{H}{R} = \frac{H}{\bar{l} \log r} \qquad \text{(式5.35)}$$

★ 信息传输速率：

$$R_c = \frac{H}{\bar{l}} \qquad \text{(式5.36)}$$

★ 编码剩余度：

$$\gamma = 1 - \eta \qquad \text{(式5.37)}$$



§5.3.2 变长码信源编码定理

结论

★ 平均码长的上、下界

- ❖ $\bar{l} \geq \dfrac{H(X)}{\log r}$ ⇨ 对所有唯一可译码都要满足
- ❖ $\bar{l} < \dfrac{H(X)}{\log r} + 1$ ⇨ 无需一定满足，但存在这种关系，通常希望越小越好

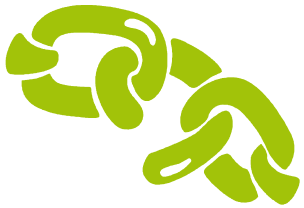
★ $\bar{l} = \dfrac{H(X)}{\log r}$ 时， $\eta = 1$，此时：

- ❖ 各信源符号出现概率为 $p_i = (1/r)^{l_i}$， l_i为整数
- ❖ 每码元平均所带信息量为 $\dfrac{H(X)}{\bar{l}} = \log r$, 码元符号独立且等概

§5.3.2 变长码信源编码定理

例5.6

设二元独立序列，$x_1=1011, x_2=1111$，其中0符号概率 $p_0=1/4$，求两个序列香农码的码长。

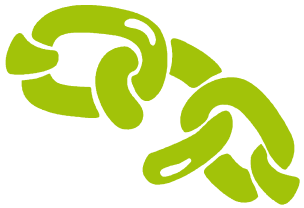


§5.3.2 变长码信源编码定理

解：

1) $l_s(x_1) = -\log_2 p(1011) = -\log_2 (1/4)\times(3/4)^3 = 3.24$比特

2) $l_s(x_2) = -\log_2 p(1111) = -\log_2 (3/4)^4 = 1.66$比特



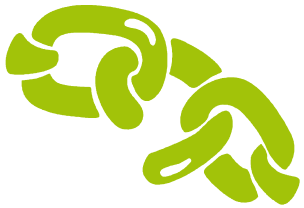
§5.3.2 变长码信源编码定理

例5.6

用例5.3的信源模型，i）对单信源符号进行二元编码，即 $s_1 \to 0$，$s_2 \to 1$，求平均码长和编码效率；ii）编成2次扩展码，信源序列与码序列的映射关系为：

$$s_1s_1 \to 0, s_1s_2 \to 10, s_2s_1 \to 110, s_2s_2 \to 111$$

求平均码长和编码效率。



§5.3.2 变长码信源编码定理

解：

1) $\bar{l}=1,\quad \eta=\dfrac{H(X)}{\bar{l}}=0.811$

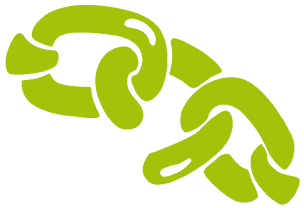
2) 信源序列的概率：

$$p(s_1s_1)=(3/4)\times(3/4)=9/16 \quad p(s_1s_2)=(3/4)\times(1/4)=3/16$$

$$p(s_2s_1)=(1/4)\times(3/4)=3/16 \quad p(s_2s_2)=(1/4)\times(1/4)=1/16$$

且：$\bar{l}=(1\times9/16+2\times3/16+3\times3/16+3\times1/16)/2$

$=27/32$



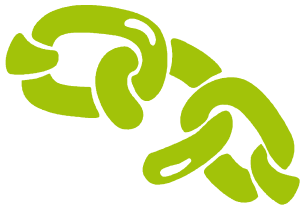
§5.3.2 变长码信源编码定理

解：

$$\eta = \frac{H(X)}{\overline{l}} = 0.811/(27/32) = 0.961$$

结论：

与例5.3相比，可以看出，为得到同样编码效率所用的编码符号数比定长码小得多。 因此容易达到高的编码效率，是变长码的显著优点。

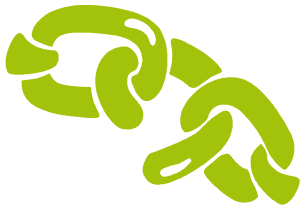


§5.3.2 变长码信源编码定理

定理5.11 香农第一定理

★ 如果信源编码码率（即编码后传送一个信源符号平均需要比特数）不小于信源的熵率，就存在无失真信源编码，反之就不存在无失真信源编码。

$$R \geq H \Leftrightarrow \text{存在无失真信源编码}$$



§5.3.2 变长码信源编码定理

一些结论

香农第一定理

①存在 $R \geq H$ 的无失真信源编码

（例：可以构造码率为R的香农码）

②H是无损压缩码率的下界，只要信源序列足够长，这个下界可达。

（例：可以构造N次扩展源的香农码，令N趋于无穷）

③不存在 $R < H$ 的无失真信源编码

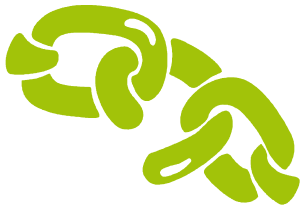
④上面①、②构成正定理的内容，而③构成逆定理的内容

⑤该定理适用于所有类型信源（包括无记忆和有记忆信源）和所有无损信源编码方式（分组码、非分组码及其他类型的编码）

§5.4 最优编码

5.4 最优编码

- 5.4.1 二元Huffman编码
- 5.4.2 多元Huffman编码
- 5.4.3 Huffman决策树
- 5.4.4 规范Huffman编码
- 5.4.5 马氏源的Huffman编码
- 5.4.6 香农码



§5.4.1 二元哈夫曼编码

★ 若一个唯一可译码的平均码长小于所有其它唯一可译码，则称该码为最优码(或紧致码)。

应注意：最优是唯一可译码之间的比较，因此它的平均码长未必达到编码定理的下界。

定理5.12

任意对于一个含n个符号的信源，存在最优的二进制码，其中有两个最长的码字有相同的长度且仅最后一个码位有别，即其中一个的最末尾是0，而另一个的最末尾是1（或者相反）

§5.4.1 二元哈夫曼编码

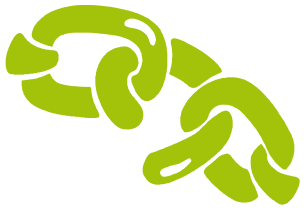
证：

★ 首先证明对于最优码，概率小的符号对应长度长的码字。

★ 证明最长的码字有两个长度相同，且只有最后一位不同。

一个最优码唯一可译 $\Rightarrow$ 满足Kraft不等式

$\Rightarrow$ 存在与其同样码长的异前置码



§5.4.1 二元哈夫曼编码

二元最优异前置码的构造方法

★ 设信源S为 $p(a_1)\geq\ldots\geq p(a_n)$ ，对应的码字为 $\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_n$

★ 将概率最小的两个码符号 a_{n-1},a_n 合并，从而产生新的信源S‘ $\{a'_1,\ldots,a'_n\}$

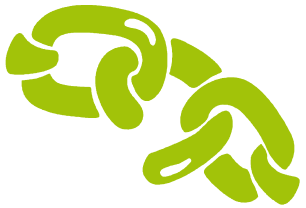
★ 设 $\{a'_1,\ldots,a'_n\}$，对应的码字为 $\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_{n\text{-}2}$ 对新信源编码后，按下面的关系就可恢复原来信源的码字：

$$\vec{x}_i=\vec{x}'_i,\qquad i=1,\ldots,n-2$$

$$\vec{x}_{n-1}=\vec{x}'_{n-1}+"0"$$

$$\vec{x}_n=\vec{x}'_n+"1"$$

（式5.40）



§5.4.1 二元哈夫曼编码

★若 $\vec{x}_i'$ 对信源 S' 是最优的异前置码，则 $\vec{x}_i$ 对信源 S 也是最优的异前置码

证：

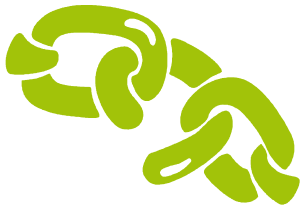
设 $S' \to l_1', \cdots, l_{n-1}'$ $S \to l_1, \cdots, l_n$ $\Rightarrow l_i = \begin{cases} l_i', & 1 \le i \le n-2 \\ l_{n-1}' + 1, & i = n-1, n \end{cases}$

对S，有

$$\bar{l} = \sum_{i=1}^{n} p_i l_i = \sum_{i=1}^{n-2} p_i' l_i' + p_{n-1} l_{n-1} + p_n l_n$$

$$= \sum_{i=1}^{n-2} p_i' l_i' + (p_{n-1} l_{n-1}) l_{n-1}' + p_{n-1} + p_n$$

$$= \bar{l}' + p_{n-1} + p_n$$



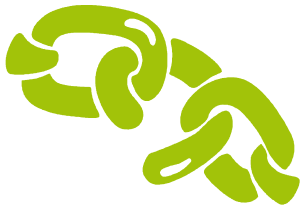
§5.4.1 二元哈夫曼编码

结论

★ 我们可以采用合并两个最小概率符号的方法，逐步地按这样的路线去编码：

$$S \to S' \to S'' \to \ldots \to 2\text{字母信源}$$

最后将2字母信源分配0、1符号；然后可逐步反推到原信源S，从而得到信源的最优编码。这种编码称做二元Huffman编码

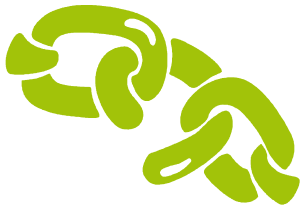


§5.4.1 二元哈夫曼编码

定理5.13

★ 二元Huffman码是最优码，即如果Huffman码的平均码长为 $\bar{l}_H$，而任何其他编码的平均码长为 $\bar{l}_C$，有：

$$\bar{l}_H \le \bar{l}_C$$



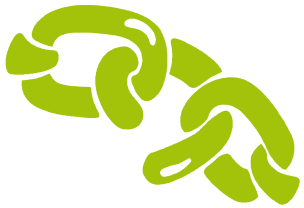
§5.4.1 二元哈夫曼编码

例5.7

一信源S的符号集A= $\{a_1, a_2, a_3, a_4, a_5\}$ ，

概率分别为：0.4，0.3，0.2，0.05，0.05；试

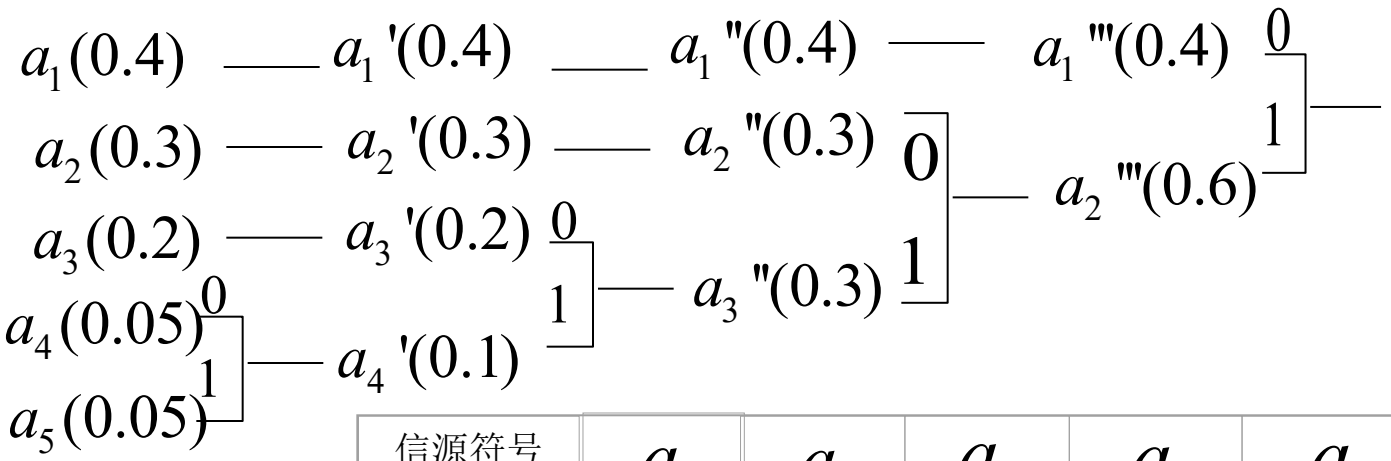
对信源符号进行二元Huffman编码



§5.4.1 二元哈夫曼编码

解：

依次做信源S′S″S‴，最后将0、1符号分配给S‴，如下图：



信源符号	a_1	a_2	a_3	a_4	a_5
码字	**0**	**10**	**110**	**1110**	**1111**

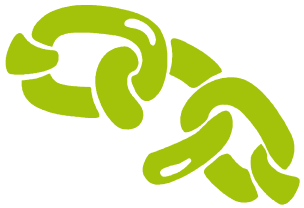
§5.4.1 二元哈夫曼编码

哈夫曼编码方法总结

★ (1) 将信源概率分布按大小依递减次序排列；
合并两概率最小者，得到信新源；
并分配 0,1 符号

★ (2) 新信源若包含两个以上符号返回（1），
否则到（3）

★ (3) 从最后一级向前按顺序写出每信源符号所
对应的码字



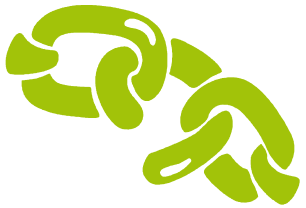
§5.2.1 二元哈夫曼编码

例5.8

一信源S的符号集A= $\{a_1, a_2, a_3, a_4, a_5\}$，

概率分别为：0.4，0.2，0.2，0.1，0.1；

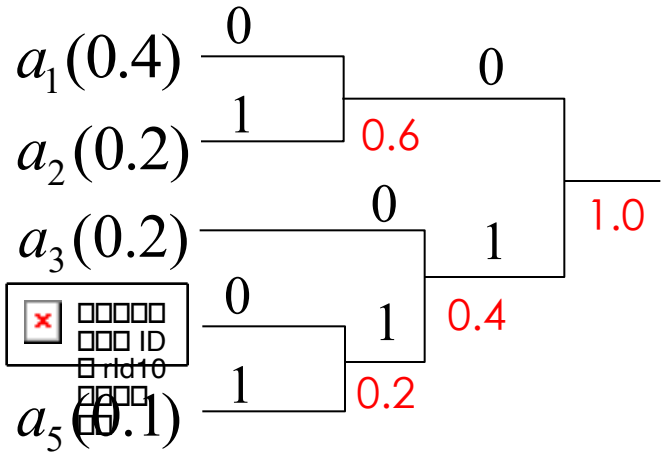
试对信源符号进行二元Huffman编码，并计算平均码长和编码效率



§5.4.1 二元哈夫曼编码

解1：

★ 编码：



a_1	a_2	a_3	a_4	a_5
00	**01**	**10**	**110**	**111**

§5.4.1 二元哈夫曼编码

解1：

$$\bar{l} = 0.4\times 2 + 0.2\times 2\times 2 + 0.1\times 3\times 2$$

$$= 2.2(\text{码元/信源符号})$$

$$H(S) = -0.4\log 0.4 - (0.2\log 0.2)\times 2$$

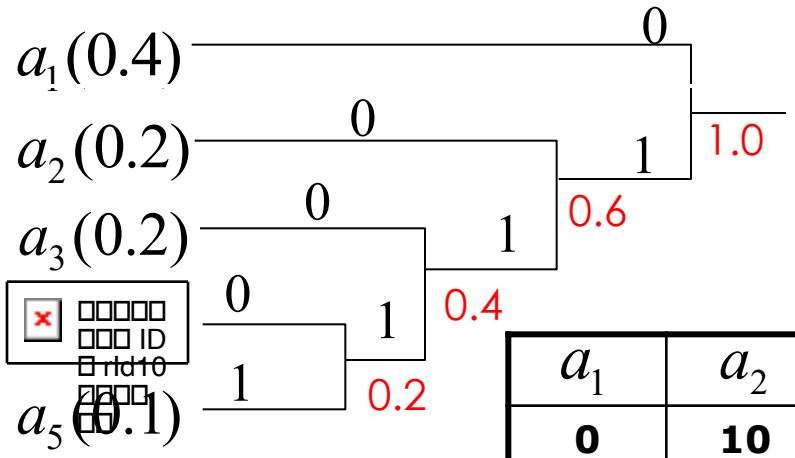
$$-(0.1\log 0.1)\times 2 = 2.122\text{比特 / 信源符号}$$

$$\eta = \frac{H(S)}{\bar{l}} = \frac{2.12}{2.2} = 0.965$$

§5.4.1 二元哈夫曼编码

解2：

★ 编码：



a_1	a_2	a_3	a_4	a_5
0	**10**	**110**	**1110**	**1111**

方法2

a_1	a_2	a_3	a_4	a_5
00	**01**	**10**	**110**	**111**

方法1

§5.4.1 二元哈夫曼编码

解2：

不变

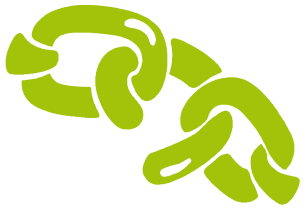
$$\bar{l} = 0.4\times1 + 0.2\times2 + 0.2\times3 + 0.1\times4\times2 = 2.2(\text{码元/信源符号})$$

但码长的方差改变了

$$\sigma^2 = E[(l_i - \bar{l})^2] = \sum_{i=1}^{n} p_i (l_i - \bar{l})^2 = \sum_{i=1}^{n} p_i l_i^2 - \bar{l}^2)^2 \quad \text{（式5.41）}$$

$$\sigma_1^2 = \sum_i p_i l_i^2 - (\bar{l})^2 = 0.4\times2^2 + 0.2\times2^2 + 0.2\times2^2 + 0.1\times3^2 + 0.1\times3^2 - 2.2^2 = 0.16$$

$$\sigma_2^2 = 0.4\times1 + 0.2\times2^2 + 0.2\times3^2 + 0.1\times4^2 + 0.1\times4^2 - 2.2^2 = 1.36$$

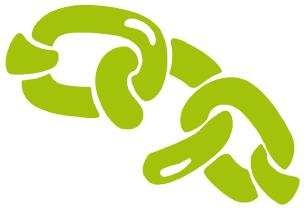


§5.4.1 二元哈夫曼编码

注：

当码长的方差小时，编码器所需缓冲器容量小。因此要尽量减小码长的方差。

方法是：在编码时，应使合并后的信源符号位于缩减信源符号尽可能高的位置上（均匀化合并次数）。



§5.4.1 二元哈夫曼编码

结论：

- ★ Huffman编码是最优码（或紧致码），是异前置码
- ★ 编码结果并不唯一，例如0、1可换，相同概率符号码字可换，但平均码长不变
- ★ 不一定达到编码定理下界，达下界条件为 $p_i = 2^{-l_i}$
- ★ 通常适用于多元信源，对于二元信源，必须采用合并符号的方法，才能得到较高的编码效率

§5.4.2 多元哈夫曼编码

★ 通过观察可知，要使编码的平均码长最短，对应的码树要构成满树是必要条件。对于r元哈霍夫曼编码，从第n阶的1个节点到n+1阶节点，增加的数目为r-1。因此，达到满树时，总的树叶数为：

码树图中每个中间节点后续的枝数为m时称满树；

$$s = r + (r-1)m \quad \text{（式5.42）}$$

m：非负整数

否则，就利用上式计算出大于q的最小正整数s。然后给信源增补零概率符号，使增补后的信源符号总数为s。编码后，去掉这些零概率符号所对应的码字，其余码字为所需码字。

§5.4.2 多元哈夫曼编码

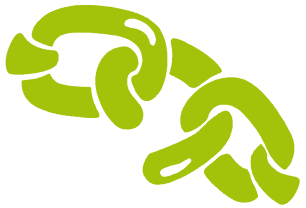
例5.9

一信源S的符号集A= $\{a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$

概率分别为：0.1,0.1,0.1,0.1,0.1,0.4,

0.05,0.05 ；试对信源符号进行3元哈夫曼编码

，并计算平均码长和编码效率



§5.4.2 多元哈夫曼编码

解：

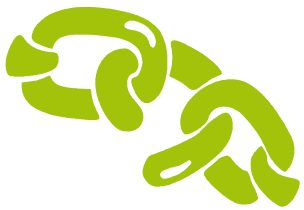
★

$$r=3 \Rightarrow \left.\begin{array}{l} s=3+2m \\ s>q=8 \end{array}\right\} \Rightarrow \begin{array}{l} m=3 \\ s=9 \end{array}$$

信源要增加1个零概率符号

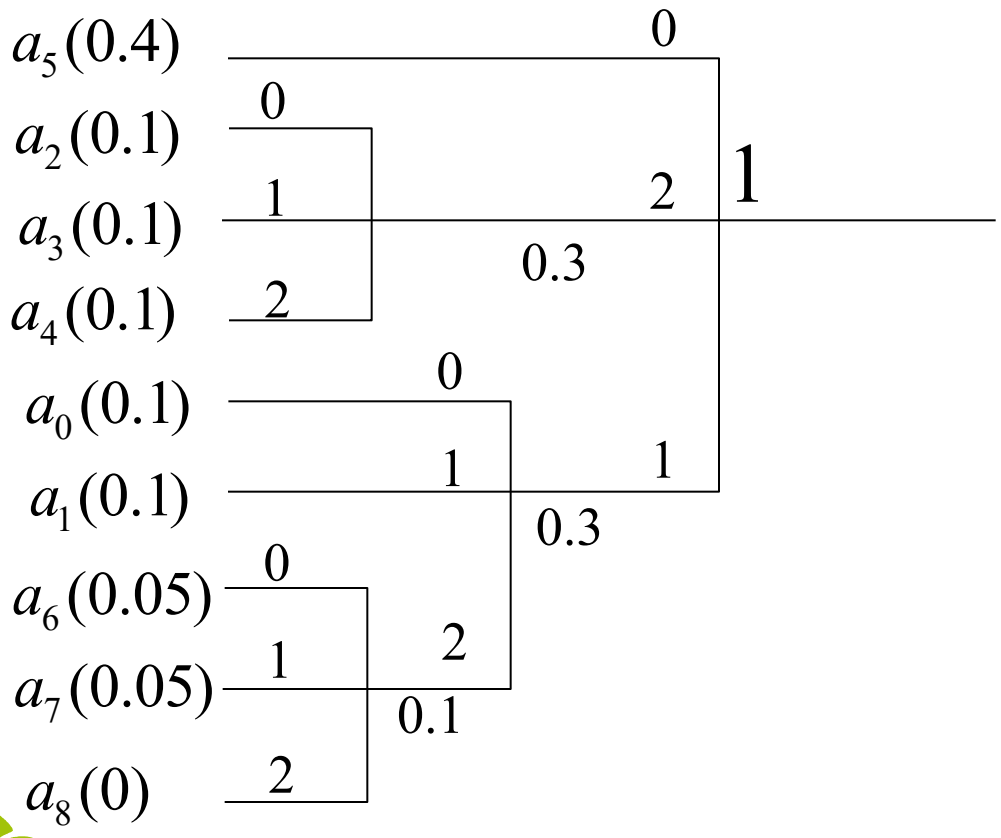
$$\left.\begin{array}{l} \bar{l}=1.7(\text{码元/信源符号}) \\ H(X)=2.622(\text{比特/符号}) \end{array}\right\}$$

$$\Rightarrow \eta=\frac{H}{R}=\frac{H}{\bar{l}\log r}=\frac{2.622}{1.7\times\log 3}=0.9729$$



§5.4.2 多元哈夫曼编码

解：



$a_0(0.4) \longrightarrow 10$

$a_1(0.1) \longrightarrow 11$

$a_2(0.1) \longrightarrow 20$

$a_3(0.1) \longrightarrow 21$

$a_4(0.05) \longrightarrow 22$

$a_5(0.05) \longrightarrow 0$

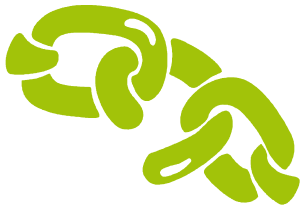
$a_6(0.05) \longrightarrow 120$

$a_7(0.05) \longrightarrow 121$

§5.4.3 哈夫曼决策树

内容

★ 如果有n个互斥随机事件，概率分别为p_i, 现用某种测试方法分步对所选择的目标事件进行识别，要求具有最小的决策平均次数，相当于对这些事件进行Huffman编码。



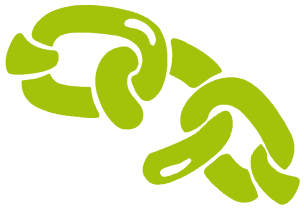
§5.4.3 哈夫曼决策树

例5.3

例如，甲手中有4张纸牌，点数分别为1、2、3、4，要求乙猜：乙可以向甲提问题，甲只能用是或否来回答。求乙平均最少问几个问题可以猜到纸牌的点数和相应的策略。

(1)1、2、3、4的概率均为1/4的决策树；

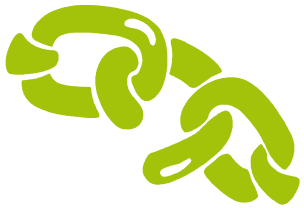
(2)1、2、3、4的概率分别为1/2、1/4、1/8、1/8的决策树。



§5.4.3 哈夫曼决策树

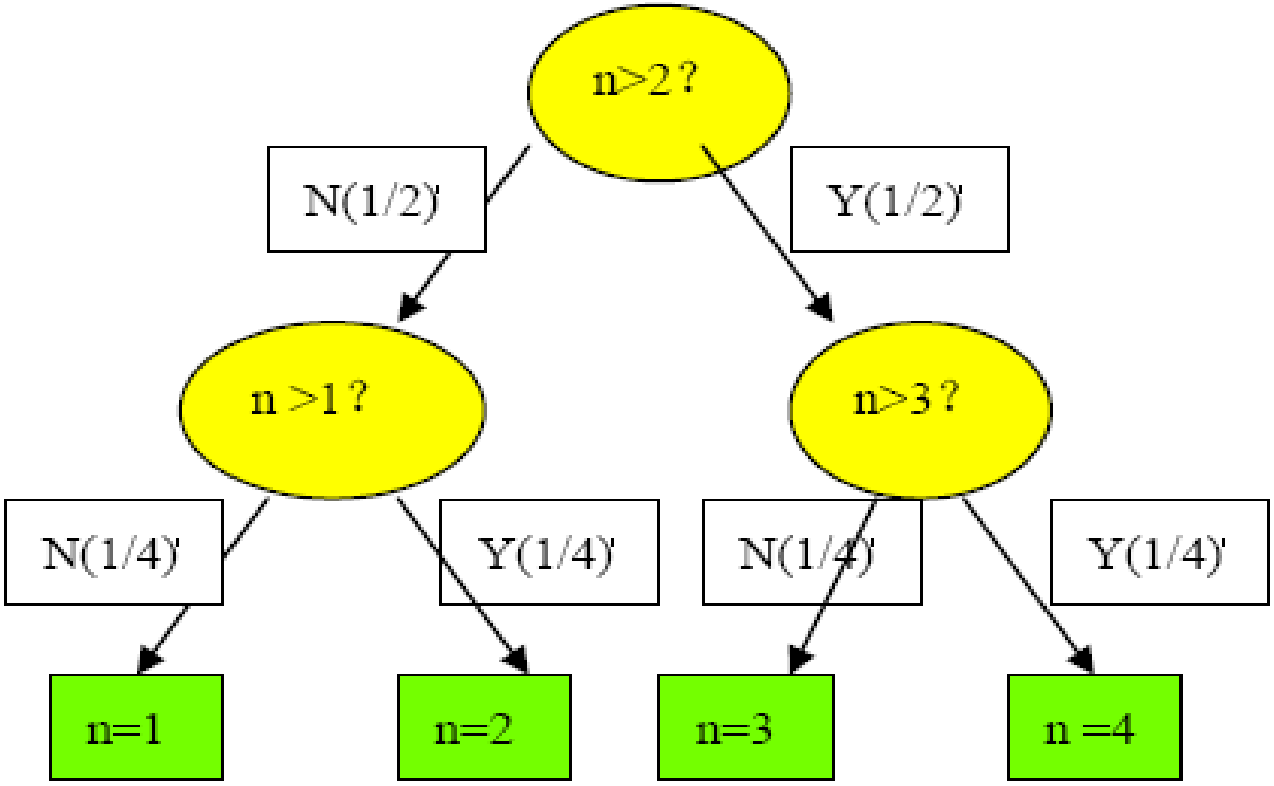
解：

- ★ 首先Huffman编码
- ★ 根据Huffman编码码树形成决策树
- ★ 每步决策结果应该与节点分支的概率匹配



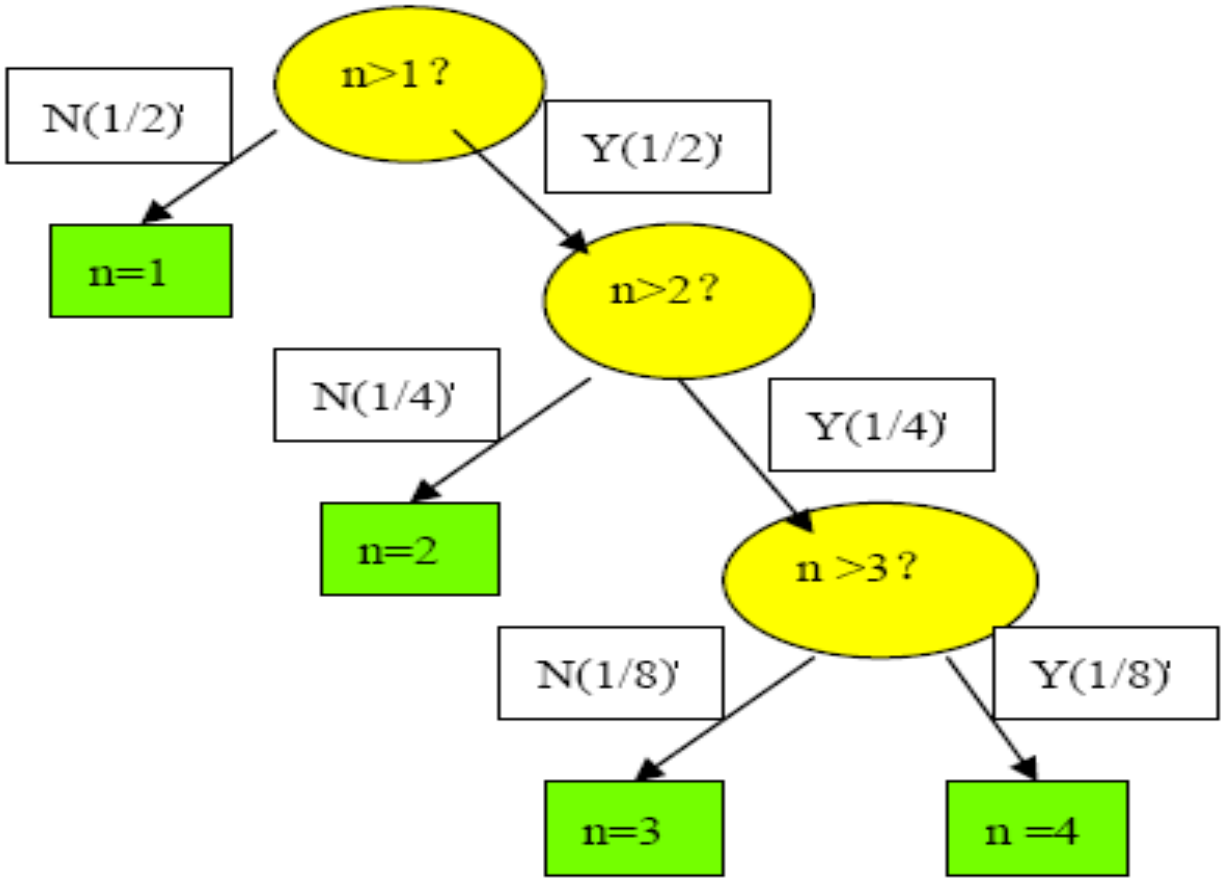
§5.4.3 哈夫曼决策树

解：



§5.4.3 哈夫曼决策树

解：



§5.4.4 规范哈夫曼编码

当信源的符号数很大时，常规Huffman编码所需的存储量就很大，译码速度也很慢。为了满足低译码复杂度和低存储量的要求而提出规范Huffman编码，这种编码是二元编码，特别适用于大字母表和快速译码要求的场合。

编码基本方法：

编码的基本方法是，将长度相同的码字编成一组，每组的码字用连续整数的二进制代码表示，因此码字以连续的存储器地址存储，这样可以用很少的数据重建Huffman树的结构，加快了编译码速度。

§5.4.4 规范哈夫曼编码

编码具体方法：

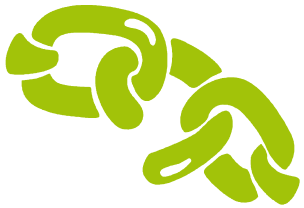
首先通过常规Huffman编码得到每个信源符号对应码字的码长，然后规范Huffman编码算法如下：

（1）先从长度为 l_1 的码组开始，将 l_1 个“0”分配给组内第一个码字；

（2）同组的其他码字为其前面码字的代码值加1；

（3）长度为 l_{k+1} 码组的第1个码字为长度为 l_k 码组中最后一个码字的二进制代码加1，并在后面补 $l_{k+1}-l_k$ 个“0”；

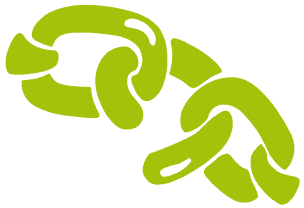
（4）步骤（2）、（3）不断重复，直到所有长度码组分配到码字，就得到规范Huffman编码。



§5.4.4 规范哈夫曼编码

例5.10

一信源符号集A={a,b,c,d,e,f,g,h,i,j}，概率分别为：0.24，0.26，0.11，0.12，0.13，0.14,试将信源符号编成规范Huffman编码。

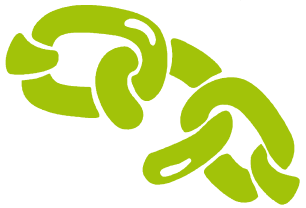


§5.4.4 规范哈夫曼编码

解：

通过常规Huffman码树得到的码字共有4种长度。
按长度分为4组：
第1组有1个码字，码长为1；
第2组有2个码字，码长为3……。

规范Huffman码字按下面方法确定：
第1组的码字确定为0，
第2组的第一个码字通过0+1后面再补2个“0”成为100，
另一个为（100+001=）101……。



§5.4.4 规范哈夫曼编码

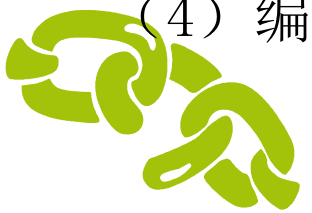
解：

Huffman树码字和规范Huffman码字
如表中第一和二行所示。

信源符号	a	b	c	d	e	f	g	h	i	j
Huffman树码字	1	001	011	0101	01001	01000	00011	00010	00001	00000
规范Huffman码字	0	100	101	1100	11010	11011	11100	11101	11110	11111

注：

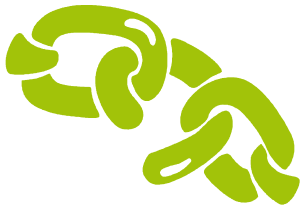
（1）规范Huffman码与通过构造Huffman码树得到的编码长度相同
（2）规范Huffman码未必可以通过Huffman码树直接得到
（3）规范Huffman码是异前置码
（4）编码器传送的编码器信息只包含每一种长度的第一个码字。



§5.4.4 规范哈夫曼编码

例5.11

某二元Huffman编码码字长度为：(2, 2, 2, 3, 5, 5, 5, 5)，试编成规范Huffman码。



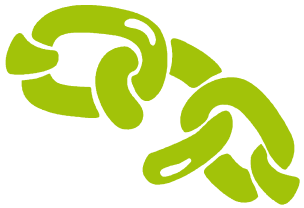
§5.4.4 规范哈夫曼编码

解：

按规范Huffman编码算法，

所有码字为：

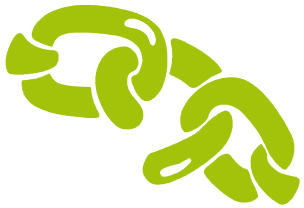
00, 01, 10, 110, 11100, 11101, 11110, 11111。



§5.4.4 规范哈夫曼编码

例5.11（续）

设译码器输入的规范Huffman编码序列为11011 101 000 1100 ，试对该序列进行译码。

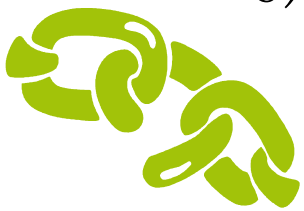


§5.4.4 规范哈夫曼编码

解：

4种长度的第一个码字的集合为
{0，100，1100，11010}。

1) 接收分组为1，因为1>0, 输入下两位:接收分组为110
2) 因为110>100，输入下一位，接收分组为1101;
3) 因为1101>1100，输入下一位，接收分组为11011;
4) 因为11010<11011<11111，判为码字：11011
5) 继续输入未判决的符号。接收分组为1，
   因为1>0, 输入下两位，接收分组为101。
6) 因为101>100，输入下一位，接收分组为1010，
   因为1010<1100，，接收分组为101，
   而101-100=001，判为码字：101。



§5.4.5 马氏源编码

马氏源可以采用按状态编码和多个符号合并编码

按状态编码：

根据马氏源的特性，当前发出的符号所含信息量取决于当前的状态。这个信息量可能很大也可能很小。例如，一个马氏源包含3个状态{a，b，c}，每个状态代表一个输出符号，状态转移矩阵如下：

$$\begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 0 & 1 & 0 \end{bmatrix}$$

下一个字母b.c出现等概

包含的信息量最大

下一个字母必然出现b，信息量为0

§5.4.5 马氏源编码

按状态编码过程：

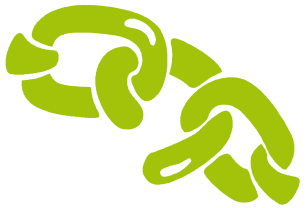
1. 给定一个初始状态 S_0

2. 对每个状态 S，根据转移概率 $p(a_i \mid s=j), i=0,1,...,q-1$

进行最优编码，例如 Huffman编码.

3. 设 $C_j(j=0,1,..,J-1)$ 为对应的码表, 其中规定

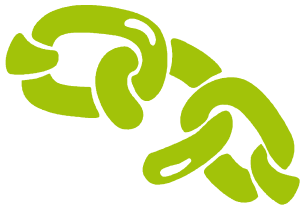
信源符号 a_i 和码字 $y_i^{(j)}$ 的对应关系，记为 $C_j(a_i, y_i^{(j)})$



§5.4.5 马氏源编码

编码过程：

1. 给定一信源序列 $x_0x_1\cdots x_n\cdots$，设初始状态 s_0
2. 用C_{s_0}码表，查出$x_0=a_{i_0}$对应的码字$y_{i_0}^{(s_0)}$作为编码器输出，同时根据 s_0,x_0得到下一个状态 s_1
3. 如此重复，直到处理完最后一个信源符号 x_n
4. 编码器输出为 $y_{i_0}^{(s_0)},y_{i_1}^{(s_1)},\ldots,y_{i_n}^{(s_n)}$



§5.4.5 马氏源编码

译码过程：

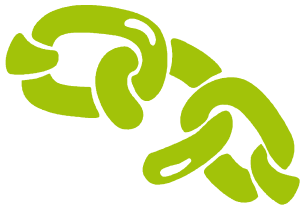
1. 根据译码器初始状态 s_0，用 C_{s_0} 码表查出其中的码字与序列 $b_0b_1...b_m$ 的前缀的相同部分，设 $b_0b_1...b_{k_0} = y_{i_0}^{(s_0)}$，则 $y_{i_0}^{(s_0)}$ 对应的 a_{i_0} 为译码器的输出
2. 根据 s_0 和 a_{i_0} 确定下一个状态，设为 s_1，则找到 C_{s_1} 码表中的码字与序列 $b_{k_0+1}b_{k_0+2}...b_m$ 中的前缀相同的部分，设 $b_{k_0+1}b_{k_0+2}...b_{k_1} = y_{i_1}^{(s_1)}$，则 $y_{i_1}^{(s_1)}$ 对应的为译码器的输出 a_{i_1}
3. 如此重复，直到最后一个序列符号处理完。

§5.4.5 马氏源编码

例5.3

对状态转移矩阵如下的马氏源进行哈夫曼编码，并计算编码效率。

$$\begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 0 & 1 & 0 \end{bmatrix} \quad (式5.43)$$



§5.4.5 马氏源编码

解：

★ 在3个状态下的Huffman编码如下

编码 / 符号 / 状态	a	b	c
a	——	0	1
b	1 0	0	11
c	——	——	——

★ 先求平稳分布 $[\pi_a \quad \pi_b \quad \pi_c]\begin{bmatrix} 0 & 1/2 & 1/2 \\ 1/4 & 1/2 & 1/4 \\ 0 & 1 & 0 \end{bmatrix}=[\pi_a \quad \pi_b \quad \pi_c]$

§5.4.5 马氏源编码

解：

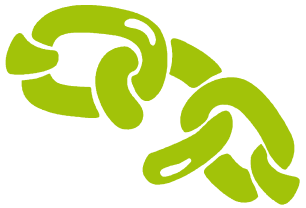
$$\pi_a + \pi_b + \pi_c = 1$$

得到 $[\pi_a \pi_b \pi_c] = \left[\frac{2}{13}\frac{8}{13}\frac{3}{13}\right]$

平均码长 $\bar{l} = \sum_i \pi_i \bar{l}_i$，$\pi_i$ 为平稳分布的概率，$\bar{l}_i$ 为在每一个状态编码的平均码长

$$\bar{l}_a = 1,\ \bar{l}_b = \frac{1}{2}\times 1 + \frac{1}{4}\times 2 \times 2 = 1.5,\ \bar{l}_c = 0$$

$$\bar{l} = \frac{2}{13}\times 1 + \frac{8}{13}\times 1.5 = \frac{14}{13}$$



§5.4.5 马氏源编码

解：

信源的熵 $H_\infty(X)=\frac{2}{13}\times 1+\frac{8}{13}\times 1.5=\frac{14}{13}$ 比特/符号

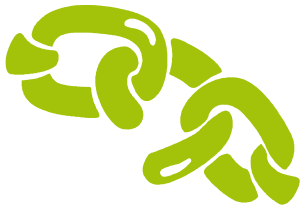
编码效率 $\eta=\frac{H_\infty(X)}{\bar{l}}=\frac{14/13}{14/13}=1$

如果利用平稳分布编码结果为： a:11， b:0， c:10

$$\bar{l}=\frac{2}{13}\times 2+\frac{8}{13}\times 1+\frac{3}{13}\times 2=\frac{18}{13}$$

$$\eta=\frac{H_\infty(X)}{\bar{l}}=\frac{14/13}{18/13}=\frac{7}{9}=78\%$$

状态编码比利用平稳分布编码效率高



§5.4.6 香农码

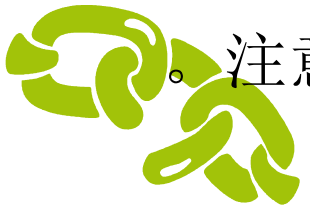
定义：$$l_i = \log_r(1/p_i)$$ （式5.44）

按式(5.44)选取码长的分组码称为香农码

编码过程：

①根据信源符号已知概率分布按式(5.29)选取每个符号的码字长度；

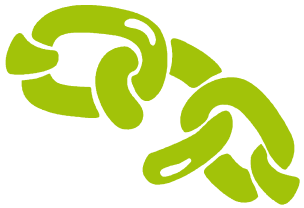
②作二元编码码树，选取阶数与码长相同的树叶作为码字。注意：在一般情况下，香农码的码树不是满树。



§5.4.6 香农码

例5.13

一信源S的符号集A={a_1， a_2， a_3， a_4}，概率分别为：9/24, 7/24, 1/4, 1/12；试将信源符号编成香农码和二元Huffman编码，并分别计算平均码长。



§5.4.6 香农码

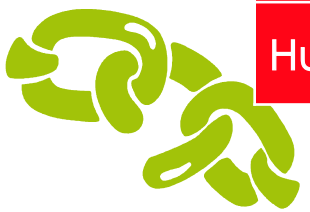
解：

设信源符号对应码字 c_1,c_2,c_3,c_4，码长分别为 l_1,l_2,l_3,l_4，对于香农码，根据上述公式，得：

$$l_1=\lceil -\log_2(9/24)\rceil=2$$

同理可得 l_2,l_3,l_4；香农码和二元Huffman码的码字、码长和平均码长如表所示。

	c_1	c_2	c_3	c_4	l_1	l_2	l_3	l_4	平均码长
香农码	11	10	01	0000	2	2	2	4	13/6
Huffman码	0	10	111	110	1	2	3	3	47/24



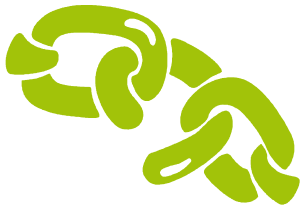
§5.5 几种实用的编码技术

5.5 几种实用的信源编码方法

5.5.1 算术编码

5.5.2 游程编码

5.5.3 LZ编码

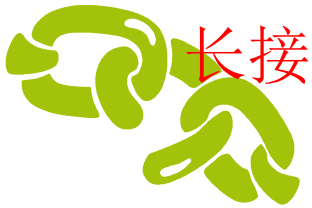


§5.5.1 算术编码

★ 算术编码是一种非分组码，非常适用于符号位数少的信源。编码时，信源符号序列连续地进入编码器，通过编码器的运算得到连续的编码器输出。

★ 算术编码是将一条信源序列映射成一条码序列，这样的码序列有时也称为码字。

算术编码的实质就是，将一条信源序列映射到[0，1）区间中的一个子区间（这种映射是一一对应的关系，以保证唯一译码），然后取这个子区间内的一点作为码字，只要码长选择合适，就可以保证唯一可译。而且当信源序列长度足够大时，每信源符号的平均码长接近信源的熵。



§5.4.6 香农码

定义：$$l_i = \log_r(1/p_i)$$ （式5.44）

按式(5.44)选取码长的分组码称为香农码

编码过程：

①根据信源符号已知概率分布按式(5.29)选取每个符号的码字长度；

②作二元编码码树，选取阶数与码长相同的树叶作为码字。

注意：在一般情况下，香农码的码树不是满树。

§5.5.1 算术编码

积累概率及计算：

积累概率是算术编码中的基本概念，包含单符号积累概率和序列积累概率。

设信源X的符号集 $A=\{a_1,a_2,\cdots,a_n\}$，对应的概率分别为 $p_1,p_2,\cdots,p_n$。定义单信源符号 a_k 的积累概率为：

$$P(a_k)=\sum_{i=1}^{k-1}p_i \qquad \text{（式5.47）}$$

其中： $P(a_1)=0$

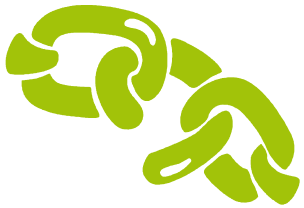
区间[0，1）分成 n个子区间，第 k 个符号对应第k个子区间，且子区间是左闭右开的

§5.5.1 算术编码

信源积累概率：

首先要对同长度序列采用字典序进行排序。对于前面的符号集A ，将各a_i的序号作为其取值，就有 $a_1 < a_2 < \cdots < a_n$ 。设序列 $x_1^m = x_1 x_2 \cdots x_m$，定义序列 x_1^m 的积累概率为：

$$P(x_1^m) = \sum_{\tilde{x}_1^m < x_1^m} p(\tilde{x}_1^m) \qquad \text{(式5.48)}$$



§5.5.1 算术编码

总结：

★ $P(x_1^m)$把区间[0，1）分成 n^m个子区间，序列 x_1^m对应的子区间$I(x_1^m)$满足

$$I(x_1^m)=[P(x_1^m),P(x_1^m)+p(x_1^m))$$ （式5.49）

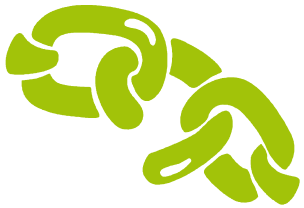
★ 同长度序列的各子区间有如下特点：

① $I(x_1^m)$的宽度等于$p(x_1^m)$；②各子区间互不相交，且它们的并构成[0，1）区间；③子区间 $I(x_1^m)$与符号 x_1^m有一一对应的关系。

§5.5.1 算术编码

总结：

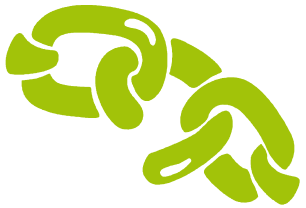
★ 可以取子区间$I(x_1^m)$内的任意一点作为 x_1^m的编码，这样的编码是唯一可译的。



§5.5.1 算术编码

例5.16

设独立信源X的符号集为符号集A={a_1， a_2， a_3}，概率分别为1/2，1/3，1/6；求序列 $a_1a_3a_2$的积累概率 $P(a_1a_3a_2)$和对应的区间 $I(a_1a_3a_2)$。



§5.5.1 算术编码

解：

$$P(a_1a_3a_2)=P(a_1a_3)+p(a_1a_3)P(a_2)=P(a_1)+p(a_1)P(a_3)+p(a_1)p(a_3)P(a_2)$$
$$=0+(1/2)\times(1/3+1/6)+(1/2)\times(1/6)\times(1/2)=7/24$$

$$L_3=P(a_1a_3a_2)=7/24$$

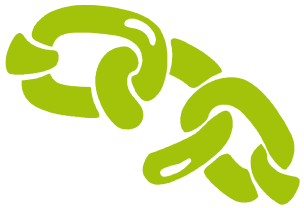
$$H_3=P(a_1a_3a_2)+p(a_1a_3a_2)=\frac{7}{24}+\frac{1}{2}\times\frac{1}{6}\times\frac{1}{3}=\frac{23}{72}$$

$$I(a_1a_3a_2)=[7/24,23/72)$$

§5.5.1 算术编码

算术编码算法：

算术编码的基本任务是将输入序列转换成码序列，每当编码器输入一个信源符号，就进行区间更新，直至最后得到整条序列所对应的子区间，再将表示该子区间内的一点作为码字输出。



§5.5.1 算术编码

算术编码编码算法：

（1）初始化：$j=0, L_j=0, H_j=1$，$\Delta_j = H_j - L_j$，

输入序列长度为m；

（2） 读信源符号 ，按区间计算图进行区间更新，

j=j+1 ；

（3） 若j ≤ m-1 返回（2），否则继续；

（4） 将子区间 $[L_m, H_m)$ 内的一个二进制小数作为

编码器输出的码字c。

§5.5.1 算术编码

算术编码唯一可译条件：

为使编码是唯一可译，该二进制小数c应该具有足够的长度，以便处于子区间$[L_m, H_m)$内。可以证明，如果c小数点后选取的位数l满足：

$$l = \lceil -\log \Delta_m \rceil = \lceil -\log(H_m - L_m) \rceil$$

就可以实现唯一译码。当编码器对最后一个符号 x_m进行编码后，将序列积累概率转换成二进制小数，取小数点后*l*位，若后面有尾数就进位，小数点保留的序列就是编码输出

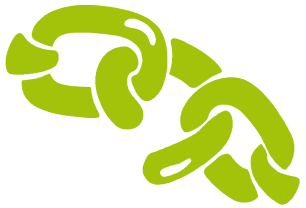
§5.5.1 算术编码

例5.16

设有二元独立序列 $x_1^4 = 1011$，符号概率 $p_0 = 1/4, p_1 = 3/4$，

（1）直接求序列积累概率对其进行算术编码；

（2）完成对序列的整个算术编码过程实现编码。



§5.5.1 算术编码

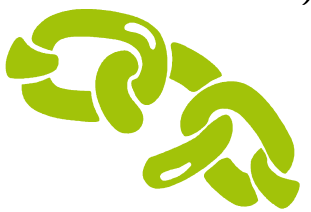
解：

（1）因为以11开头的序列都排在1011的后面，所以根据积累概率的定义有：

$$P(1011)=1-p(11)-p(1011)=1-(3/4)^2-(3/4)^3\times 1/4=85/256$$

码长取为：$l=\lceil \log_2(1/p(1011)\rceil=\lceil -\log_2((3/4)^3\times 1/4)\rceil=4;$

在 $P(1011)$的二进小数0.01010101中取小数点后面的前4位。因后面有尾数，所以再进位到第4位，得到小数0.0110。码字取小数点后面的部分，尾零去掉，得码字 c=011。

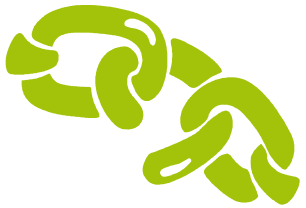


§5.5.1 算术编码

解：

（2）编码过程如表所示，这里有0、1两个符号

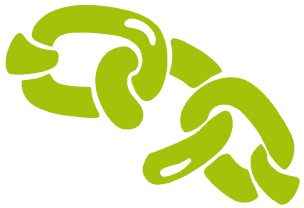
$l_0 = 0,\ h_0 = l_1 = 1/4,\ h_1 = 1$子区间初始化为$[0, 1)$；每输入一个符号就进行区间更新，直到最后一个符号输入后，区间为$[85/256, 7/16)$，区间宽度为$27/256$；类似前面的方法得到码字为c=011。



§5.5.1 算术编码

解：

j	a_j	L_j	H_j	Δ_j	c
0		0.	1	1.	
1	1	1/4=0+1×1/4	1=0+1	3/4=1-1/4	
2	0	1/4=1/4	7/16=1/4+(3/4)×1/4	3/16=7/16-1/4	
3	1	19/64=1/4+(3/16)×1/4	7/16=1/4+(3/16)	9/64=7/16-19/64	
4	1	85/256=19/64+(9/64)×1/4	7/16=19/64+(9/64)	27/256=7/16-85/256	011



§5.5.1 算术编码

算术编码译码算法：

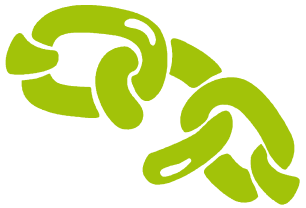
（1）初始化：$j=0, L_0=0, H_0=1$，$\Delta_0=1$，序列长度m；

（2）将接收序列转换成码字c；

（3）若归一化码值 $(c-L_j)/\Delta_j \in I_i, i=1,\cdots,n$ 输出符号 a_i；

（4）按式前面图中进行区间更新，$j=j+1$；

（5）若j ≤m-1 ，则返回（3）；否则，结束。



§5.5.1 算术编码

例5.17

设将例5.16中编成的码字进行译码

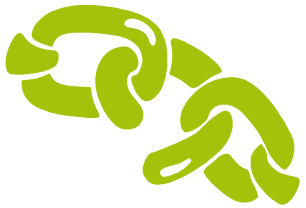
解：

译码器输入为小数$(0.011)_2=3/8$。因为是0、1二元编码，当译第j个符号时，$\gamma=L_{j-1}+P(x=1)\Delta_{j-1}$ $(P(x=1)=1/4)$将区间I_j分成两个子区间，其中$[L_j,\gamma)$和$[\gamma,H_j)$分别对应序列$x_1^{j-1}\cdot 0$和$x_1^{j-1}\cdot 1$，所以γ可作为判决门限；若$c>\gamma$，判定$x_j=1$，否则，判定$x_j=0$。根据比较结果，输出信源符号$a_j=1$，然后进行区间更新，直到译码结束，译码过程如表所示。

§5.5.1 算术编码

解：

j	比较	a_j	L_j	H_j	Δ_j
0	-	-	0.	1	1.
1	3/8>1/4	1	1/4=0+1×1/4	1=0+1	3/4=1-1/4
2	3/8<7/16	0	1/4=1/4	7/16=1/4+(3/4)×1/4	3/16=7/16-1/4
3	3/8>19/64	1	19/64=1/4+(3/16)×1/4	7/16=1/4+(3/16)	9/64=7/16-19/64
4	3/8>85/256	1	85/256=19/64+(9/64)×1/4	7/16=19/64+(9/64)	27/256=7/16-85/256



§5.5.1 算术编码

技术难点：

及时性、精度问题、运算量过大、进位差错

优点：

灵活性、最优性

缺点：

运算量大、差错传播

§5.5.2 游程编码

★ 游程编码是适用于二元信源的有效编码方法，可用于黑、白二值文件的传真压缩。

★ 连“0”子序列称为0游程，连“1”子序列称为1游程，这些子序列的长度称为游程长度，0游程和1游程总是相间分布的。

★ 任意一条二元序列都可用交替出现的0游程和1游程的长度按原顺序构成的序列来表示，这种序列称为游程长度序列，简称游程序列。

★ 把原序列变成游程序列的变换叫做游程变换，这是一种一一对应的变换。

§5.5.2 游程编码

★ 游程编码实际上由两部分组成：游程变换加熵编码。游程变换并不进行压缩，只是使后面的熵编码更容易进行压缩。黑、白图像传真压缩技术中使用的游程编码由游程变换和修正Huffman编码（MH码）结合而成，方法如下：

①游程长度在0～63：直接查找相应的黑或白结尾码作为码字；
②游程长度在64～1728：用黑或白组成码和结尾码的组合作为码字；
③游程长度在1792～2560：黑白游程用一附加组成码的码字；
④规定每行从白游程开始（长度可以为零），每行用一个结束码（EOL）终止；
⑤用于传输时，每页数据之前加一个结束码，每页尾部连续用6个结束码。

§5.5.3 LZ编码

★ LZ编码是一种通用编码，很多这类压缩算法的变种，统称为LZ编码。LZ编码不利用信源的统计特性，而采用基于字典的编码技术，其共同特点是，实现简单，而且渐进码率接近信源的熵，算法快速而高效，基本思路如下：

①把信源序列分成长度不完全相同的字符串，也称作词组，对每个词组逐一进行编码。

②当对某词组编码时，就到字典中搜索该词组。在字典中找到这个词组，称作匹配。

③如果发生匹配，就将该词组在字典中标号作为它的编码；

④如果没有匹配，就直接输出该词组的原始未压缩的形式。

因此，编码器的输出文件是由标号和原始词组构成。

§5.5.3 LZ编码

LZ77算法分类：

滑动窗LZ算法（SWLZ）、固定数据库LZ算法（FDLZ）

LZ77算法输出标号构成：

偏移、匹配长度、匹配段观察缓冲器中的下一个符号

（改进后的LZSS输出标号由两部分组成：偏移和匹配长度；

如果无匹配，那么编码器发送下一个符号的未压缩代码）

LZ77算法特点：

优点：特别适合于一次压缩和多次解压的场合。

缺点：窗长有限、观察缓冲器的大小有限。

§5.5.3 LZ编码

LZ78算法简介：

（1）采用由碰到的输入文本中的字符串所构成的字典

（2）先将信源序列分成一系列以前未出现而且最短的字符串或词组

LZ78算法构成：

编码输出的标号由两部分组成：

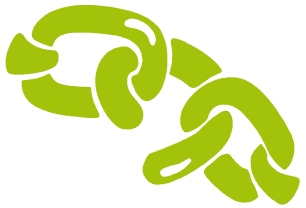
一是字典指针；二是尾字符的编码，标号不含匹配长度。

（一个标号对应一个字符串）

§5.5.3 LZ编码

LZW算法简介：

主要特点是删除了LZ78标号中的第二部分，标号中仅包含字典指针。编码器按一定的规则将信源序列分成序号连续的词组，构成字典的元素，并发送每个词组前缀的地址（字典指针）。译码器利用相同的规则构建字典，根据接收到的前缀地址重建每个词组，从而恢复信源序列。



§5.5.3 LZ编码

LZW算法编码：

编码器将以上述原则划分所得到的词组作为字典元素，用有序对$<n, a_i>$表示，其中n为词组前缀在字典中的地址；a_i为词组的尾符号。只有第一次出现的新词组才存到字典中。这样，这些有序对就构成一个链接表。字典中每一个元素都分配一个地址，使得元素与地址有一一对应的关系。此外还要建立一个初始化字典，信源符号作为初始字典的元素。编码器的输出就是词组前缀在字典中的地址n。

§5.5.3 LZ编码

LZW算法译码：

译码器必须建立与编码器相同的字典才能对编码序列进行译码

工作过原理如下：

① 接收任何码字时都必须建立新的字典元素；

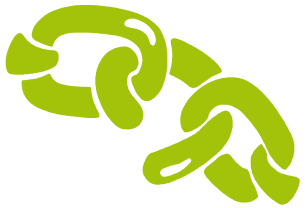
② 新的字典元素的指针n与接收码字的n相同；

③ 确定字典元素的方法：设当接收码字为n_t时，地址指针为m，那么对应的字典元素为< n_t, ?> ，因为当时还未收到关于信源符号的信息。

§5.5.3 LZ编码

LZW算法译码：

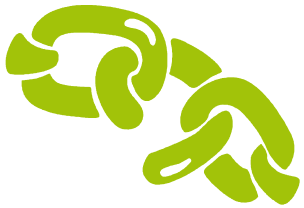
④而当接收码字为n_{t+1}时，地址指针为m+1，那么对应的的字典元素为< n_{t+1}, ?>。因为< n_t, ?> 和n_{t+1}, ?>对应着两个连接的词组， n_{t+1}地址词组的第1个符号就是< n_t, ?>对应词组的尾符号。而通过查字典可以找到n_{t+1}地址词组的第1个符号。这个符号就是< n_t, ?>中的“？”。因此译码要延迟一个词组的时间。



§5.5.3 LZ编码

例5.18

一个二元信源输出序列为：110 001 011 001 011 100 011 11……，建编码字典并确定编码器输出序列。



§5.5.3 LZ编码

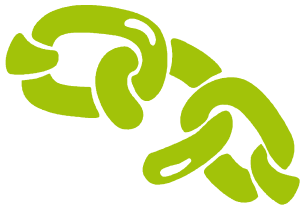
解：

编码过程				译码过程			
信源词组	编码存储器地址M	编码字典元素	发送码字	译码存储器地址m	译码部分字典元素	译码完整字典元素	译码输出
空	0	<0，null>	-	0	-	<0,null>	-
-	1	<0,0>	-	1	-	<0,0>	-
1	2	<0,1>	-	2	-	<0,1>	-
11	3	<2,1>	2	3	<2,? >	<2,1>	1
10	4	<2,0>	2	4	<2,? >	<2,0>	1
00	5	<1,0>	1	5	<1,? >	<1,0>	0
001	6	<5,1>	5	6	<5,? >	<5,1>	0 0
101	7	<4,1>	4	7	<4,? >	<4,1>	1 0
110	8	<3,0>	3	8	<3,? >	<3,0>	1 1
0010	9	<6,0>	6	9	<6,? >	<6,0>	0 0 1
01	10	<1,1>	1	10	<1,? >	<1,1>	0
111	11	<3,1>	3	11	<3,? >	<3,1>	1 1
100	12	<4,0>	4	12	<4,? >	<4,0>	1 0
0011	13	<6,1>	6	13	<6,? >		0 0 1
111…	14						-

§5.5.3 LZ编码

解：

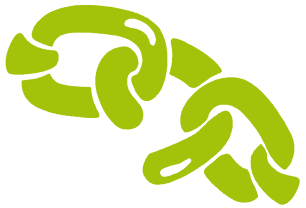
编码过程如表左边4列所示。编码开始时：
第1个词组为1,但1在初始化字典中已经存在所以不存入字典；
第2个词组为11，建字典元素为<2,1>，
此处2为符号1的字典地址，输出码字2；
依次类推…，得到所有字典元素，
最后输出序列为：2 2 1 5 4 3 6 1 3 4 6。



§5.5.3 LZ编码

例5.18（续）

试将LZ译码器输入序列2 2 1 5 4 3 6 1 3 4 6进行译码。



§5.5.3 LZ编码

解：

编码过程				译码过程			
信源词组	编码存储器地址M	编码字典元素	发送码字	译码存储器地址m	译码部分字典元素	译码完整字典元素	译码输出
空	0	<0，null>	-	0	-	<0,null>	-
-	1	<0,0>	-	1	-	<0,0>	-
1	2	<0,1>	-	2	-	<0,1>	-
11	3	<2,1>	2	3	<2,? >	<2,1>	1
10	4	<2,0>	2	4	<2,? >	<2,0>	1
00	5	<1,0>	1	5	<1,? >	<1,0>	0
001	6	<5,1>	5	6	<5,? >	<5,1>	0 0
101	7	<4,1>	4	7	<4,? >	<4,1>	1 0
110	8	<3,0>	3	8	<3,? >	<3,0>	1 1
0010	9	<6,0>	6	9	<6,? >	<6,0>	0 0 1
01	10	<1,1>	1	10	<1,? >	<1,1>	0
111	11	<3,1>	3	11	<3,? >	<3,1>	1 1
100	12	<4,0>	4	12	<4,? >	<4,0>	1 0
0011	13	<6,1>	6	13	<6,? >		0 0 1
111…	14						-

§5.5.3 LZ编码

解：

译码过程如表的右边4列所示。译码开始时，
n=2，m=3，部分字典元素为<2,?>，因为n=2表示词组前缀地址，对应字典元素为<0,1>，所以输出1，到下一步；n=2，表明在地址m=2的词组第1个符号是前面<2,?>中的？，所以<2,?>= <2,1>，……
接收码字为n=6，m=9，部分字典元素为<6,?>，m=6内容为<5,1>，m=5内容为<1,0>，m=1内容为<0,0>，所以前缀n=6内容为001，作为当前输出…，
译码输出为：110 001 011 001 011 100 01。

§5.5.3 LZ编码

LZ编码的优点：

①编码包括字符串的搜索和匹配操作，无数值运算；

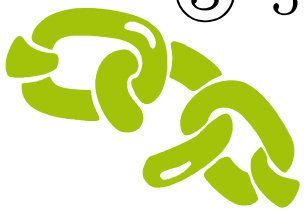
②译码简单。

LZ编码的应用：

① UNIX计算机系统中广泛使用的文件压缩程序compress；

② GIF图像压缩；

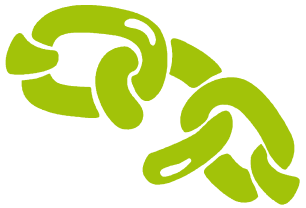
③ 3.V.42bis协议。



本章小结

★ 信源编码的主要目的是提高信息传输的有效性，分为如下几类

- ❖ 概率匹配编码（信源符号概率已知）
  - 分组码： 定长码， 变长码
  - 非分组码
- ❖ 通用编码（信源符号概率未知）



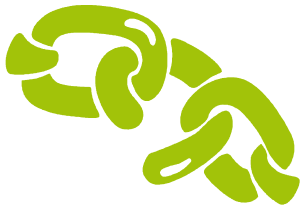
§5.2.3 定长码信源编码定理

★ 信源序列渐近均分特性

- ❖ 典型序列的概率$p(\vec{x})=2^{-N[H(X)\pm\delta]}$，个数$N_G\approx 2^{NH(X)}$，
- ❖ 当序列长度N足够大时，有$\left|\frac{1}{N}\log p(\vec{x})+H(X)\right|<\delta$

★ 唯一可译码必须满足Kraft不等式

$$\sum_{i=1}^{n} r^{-l_i}\le 1$$



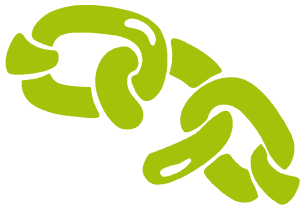
本章小结

★ 无失真信源编码定理（香农第一定理）

若对信源X的N次扩展源 X^N 进行编码，当N足够大时，总能找到唯一可译的r进编码，使得X的平均码长任意接近信源的熵 $H_r(X)$

$$\bar{l} \geq H_r(X), \bar{l} \to H_r(X)$$

$$R \geq H_\infty(X), R \to H_\infty(X)$$



本章小结

★ 关于信源编码定理的另一种描述

只要编码后，信息传输速率不大于无噪声信道的容量，就可实现无失真信源编码。

$$\eta = \frac{H}{\overline{l}\log r} = \frac{R_c}{C} \le 1$$

$R_c = \dfrac{H}{\overline{l}}$ 为编码信道信息传输速率

$C = \log r$ 为无噪信道的容量

本章小结

★ 编码序列的特性

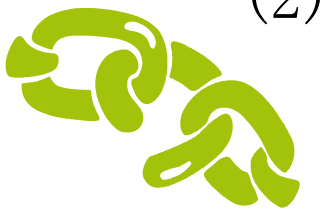
R的最大限制： $R \le \log r$

$R = \log r \Rightarrow$编码符号独立且编码符号等概率

★ 无失真信源编码所采取的主要措施

(1) 概率匹配（Huffman编码等）使编码符号等概率

(2) 解除相关性，使信源变成无记忆

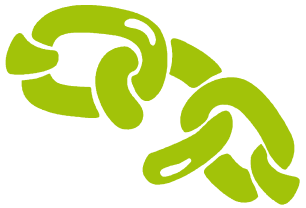


本章小结

无失真信源编码限制：

★ 典型序列个数估计 $N_G = 2^{NH(X)}$，若 $H(X) = \log_2 n$ 则 $N_G = n^N$，即每个序列都是典型序列。要实现无失真，必须有 $r^l \geq n^N$。与无编码情况一样。

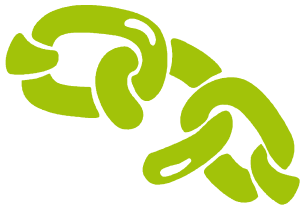
★ 当信源的熵接近 $\log_2 n$ 时，无失真信源编码的意义不大；此时信源冗余度 $r = 1 - \eta = 1 - \dfrac{H(X)}{\log_2 n} = 0$，没有压缩的余地。



本章小结

信源压缩编码下限：

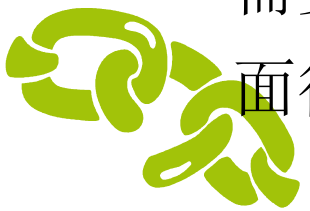
- 不采用信源编码时每信源符号的码长为 $\log n / \log r = \log_r n$
- 而通过压缩编码后的平均码长会减小，但大于等于 $H(X)/\log r = H_r(X)$
- 压缩编码的目的就是尽量降低传送每个信源符号时所需的比特数，而信源的熵 $H_r(X)$为无失真压缩码长的下限。



本章小结

几种重要编码：

- ★ Huffman编码：最优编码，需要知道信源的概率分布，对于小符号集信源不适合
- ★ 算术编码：性能优良，特别适用于二元信源的非分组熵编码，有广泛应用
- ★ 游程编码：针对有记忆信源的有效编码方法，用于黑白图像传真压缩
- ★ LZ编码： 一种通用信源编码，算法简单，不需要知道信源概率分布，在计算机文件压缩方面得到广泛应用



谢谢！