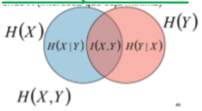
Informação: **i(a)** = $\log_2(1/P(a))$; **i(a,c)** = i(a) + i(c);

Y Determinística -> H(Y|X) = 0 Tdep
Invertível: H(X) = H(Y)

Propriedades de H(X):

Entropia: nº médio de bits para codificar uma fonte de informação
H(A) = - Sum($P(a_i)\log_2(P(a_i))$) = Sum($P(a_i)\log_2(1/P(a_i))$);
H(X)≥0(igual se p_i=1);
0<=**H(X)**<=log2(M), M - nº de elementos do alfabeto;
H(X) = log2(M) Se os eventos forem equi-prováveis;
H(X,Y) = H(X) + H(Y) se os acontecimentos forem independentes.
H(X,Y) = H(Y)+H(X|Y) = H(X)+H(Y|X)≤H(X)+H(Y) (agrupamentos);
H(X,Y)≤H(X);
H(Y|X) –incerteza remanescente após a observação de Y e conhecendo X
H(Y|X)≤H(Y) (igual se for idep.), a info de contexto reduz a entropia;
H(X|Y)≤H(X) (igual se for idep.);
H(X,Y)= - Sum(i)Sum(j)P(X=xi,Y=yj)log2(P(X=xi,Y=yj))
H(X|Y)= - Sum(i)Sum(j)P(X=xi,Y=yj)log2(P(X=xi|Y=yj))

Informação Mútua: diminui com o aumento da prob.



Independentes: **I(X;Y)**=0;
Total depend.: **I(X;Y)**=H(X)=H(Y)
I(X;Y)=H(Y)-H(Y|X)=H(X)-H(X|Y)≥0

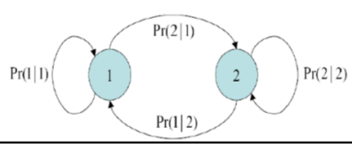
$$J = \sum_{i=1}^{n} P(x=i)\log_2 P(X=i) + \lambda\left(\sum_{i=1}^{n} P(x=i) - 1\right)$$

$$\log_2 P(x=i) = -\lambda - \frac{1}{\ln 2} \to P(x=i) = 2^{-\lambda} 2^{-\frac{1}{\ln 2}}$$

$$\frac{\partial J}{\partial \lambda} = \sum_{i=1}^{n} P(x=i) - 1 = 0 \to \sum_{i=1}^{n} P(v=i) = 1$$

$$\sum_{i=1}^{n} 2^{-\lambda} 2^{-\frac{1}{\ln 2}} = 1 \to 2^{-\lambda} 2^{-\frac{1}{\ln 2}} = \frac{1}{n} \to P(x=i) = 1/n$$

Modelo de Markov:



H=P(1)H(1)P(2)H(2)
H(1)= - P(1|1)$\log_2$P(1|1) – P(2|1)$\log_2$P(2|1)
H(2)= - P(2|2)$\log_2$P(2|2) – P(1|2)$\log_2$P(1|2)

$$\begin{cases} P(1) + P(2) = 1 \\ P(1).P(2|1) = P(2).P(1|2) \end{cases}$$

L ≥ H(X)
Huffman: H(X) ≤ **L** < H(X) + 1;
Agrupamento de símb (maj. bitrate): H(X) ≤ **L** < H(X) + 1/n;
Aritméticos: H(X) ≤ **L** < H(X) + 2; -> traduz linearmente
Agrupamento de simb (maj. bitrate): H(X) ≤ **L** < H(X) + 2/n;
D_{KL}(P,Q)=Sum P(X)log2(P(X)/Q(X));
Se P(X) = Q(X), D_{KL}(P,Q)≥0;

Árvores de Huffman: m = nº de elementos do alfabeto

Códigos Pré-acordados:
$2^e + r = m, 0 \le r \le 2^e$
Se $k \le 2r \ \to valor = k - 1 \to e + 1\ bits$
Se $k > 2r \ \to valor = k - r - 1 \to e\ bits$
Nº de nós = 2m-1
Bloco: conjunto de nós com o mesmo peso e que não são pai do nó.
Tx = C(NYT)|C(valor) ou Tx=0 ou Tx=1

Cíclicos(n,k)
1: r = X^(n-k)u(x)
2: b(x) = r/g(x) ->0-k

Códigos Aritméticos: ~ fano-elias

$$n \ge \frac{H(z)}{\log_2 Ly - Hnorm(X)}$$
$Hnorm(X) = H(X)/n$

- Calcular prob acumuladas $F(X_1)$,$F(X_2)$,…
- $l_0 = 0$; $u_0 = 0$;
- $l^i = l^{i-1} + (u^{i-1} - l^{i-1}) \cdot F(a_{k-1})\ [inferior]$
- $u^i = l^{i-1} + (u^{i-1} - l^{i-1}) \cdot F(a_k) [superior]$
- Está no intervalo [0,0.5[-> transmite 0 e expande: 2x;
- Está no intervalo [0.5,1[-> transmite 1 e expande: 2(x-0.5);
- Não está em nenhum dos intervalos: segue para a próx iteração
- Na ultima iteração: transmite o nº médio entre l e u:
  TAG = $\frac{l+u}{2}$

Um **código de prefixo** é instantâneo;
- É unicamente descodificável se um código não for sufixo de outro: $\sum_i^n D^{-l_i} \le 1$
- Para ser óptimo: $\sum_i^n D^{-l_i} = 1$

Propriedades dos restos:

(n+id) mod d = n mod d
($n_1 \pm n_2$)mod d = $n1$(mod d)± $n2$ ((mod d))mod d
($n_1 n_2$)mod d = $n1$(mod d) $n2$ ((mod d))mod d

$a^{\varphi(n)}$ mod n =1;**H(X,Y,Z)**=H(X)+H(Y|X)+H(Z|X,Y); **I(X;Y|Z)**=H(X|Z)-H(X|Z,Y)

$$\frac{\partial J}{\partial P(x=i)} = \log_2 P(x=i) + P(x=i)\frac{1}{P(x=i)}\frac{1}{\ln 2} + \lambda = 0$$

Não repúdio: mecanismo de garantia de segurança que impede uma entidade participante, numa dada operação, de negar essa participação.

Teorema de Shannon – Segredo Perfeito:

x-mensagem
z-chave
y-mensagem encriptada

Encriptador é perfeito
H(Y|XRZS)=0
Desencriptador é perfeito
H(X|YRZ)=0
H(X)=H(YZR); H(X)=H(X|Y) -> a cifra não deve conter info da sms
Sistema inquebrável: O segredo será perfeito se os dados (R,Y) observados pelo inimigo forem estatisticamente independentes da mensagem (o que impede a sua descodificação a partir dos dados observados!)
H(X|YR)=H(X) = H(Z|YR)+H(X|ZYR) (é 0) ≤ H(Z)
H(X|YR) ≤ H(XZ|YR)
H(Z|Y_1,Y_2,…,Y_N) deve ser máximo;->significa independentes cifra e key
A incerteza da chave secreta tem que ser pelo menos igual à incerteza da mensagem!

LZ78:

LZ77: $\lceil \log_2 Ns \rceil + \lceil \log_2 Ns + NL \rceil + \lceil \log_2 A \rceil$

EX: pipapipapipo
Dicionário vazio à partida
A cada ocorrência: <i, c(a)>, i-indice, c(a)-código a

codificador	índice	entrada
<0, c(p)>	1	p
<0, c(i)>	2	i
<1, c(a)>	3	pa
<1, c(i)>	4	pi
<3,c(p)>	5	Pap (…)

Códigos dicionário mais eficientes que aritméticos
Dicio não agrupamento

LZ77 (dimensão janela 30, look-ahead buffer 15 (próximos N(15) símb a serem codificados, search-buffer 15 (últimos N(15) símb codificados))
{0,0,código(b)} c= 2 **b**
{0,0,código(a)} c= 1 b**a**
{0,0,código(r)} c= 4 ba**r**
{1,1,código(a)} c= 2 ba**rr**
{0,0,código(y)} c= 2 barra**y**
{5,2,código(#)} c= 2 ba**rrayar#** (…)

O LZW é melhor, visto não ser preciso escrever os codificados sendo apenas necessário escrever os índices.
LZ8(índice+caracter) LZ77(padroes loc

LZW: **EX:** barrayar#bar#

simbolo	indice
a	**1**
b	**2**
#	**3**
r	**4**
y	**5**
ba	6
ar	7

(entradas iniciais) Iteração1:
- Padrão máximo=“b”;
- <enviar 2>
- criar entrada “ba” com índice 6
- Continuar analise no padrão “a”

Algoritmo de Euclides:
mdc(30030,257)
30030 = 257x116 + 218
257 = 218x1 + 39
218 = 39x5 + 23
39 = 23x1 + 16
(…)
7 = 2x3 + **1 (mdc)**
2 = 1x2 + 0

GIF:
2^{b+1} entradas iniciais;
Quando dicionário cheio:
• Duplicar espaço disponível
• Até atingir um máximo de 4096 entrad
– Uma vez atingidas as 4096 entradas
o dicionário passa a ser estático -> RESI

Entre 0-255 – literal, 256 fim do bloco
257-258:<comprimento,dist recuar>;
BFINAL: 1 se ultimo bloco, 0 o resto
BTYPE 00 s/comp, 01-huff fix, 10 huf dinâmico, 11 reservado
códigos huff: literal/comp e dist recuar
HLIT-cod literais/compr – 257-286
HDIST #codigos de distancia (1-32)
HCLEN (códigos de comprimento de código) (4-19)
HCLEN+4*3 (códigos pela ordem)

RSA:
2 números primos (p e q);
n = p*q ;
$\varphi(n)$ = (p-1)(q-1);
expoente e < n em que mdc(e,(p-1)(q-1))=1 ;
expoente de desencriptação ed mod(p-1)(q-1)=1 ;
E(m) = m^e mod n ; D(c) = c^d mod n
Chave: pública (n,e), privada (p,q,d).
Nºs primos menores que n: n/ln(n);
Nº provavelmente primo: a^{p-1}mod p =1
- p-1 = 2^x+2^y+(…);
- $2^{2^\wedge 2}$ mod 101 = 16 mod 101 = 16; $2^{2^\wedge 3}$ mod 101 = 16^2 mod 101 = 54

Algoritmo de Euclides estendido:
172x + 190y = mdc(172,190) **x0 = 0** ; **x1 = 1** ; **y0 = 1** ; **y1 = 0**
- Algoritmo de Euclides (**k**=nº da iteração, começa em 1);
- **q** é o factor a multiplicar (k=1 -> X_2)
- X_k = - $q_{k-1}X_{k-1}$ + X_{k-2}
- Y_k = - $q_{k-1}Y_{k-1}$ + Y_{k-2}
- Calcular para todos os **k**.

Cifra = Chave XOR Mensagem
Mensagem = Chave XOR Cifra

Capacidade do canal:
- **C(Q)** = max I(X;Y) = 1-h_2(p) (canal simétrico binário);
- h_2(p) = - (1-p) $\log_2$(1-p) – p $\log_2$(p);
- **R(Pb)** = C/(1-2H(Pb));

DES 64 bits:
Gera 16 chav
Permutação tabela usada 56bits
2^n * 2^n + 2^n = 2^(2n) + 2^n