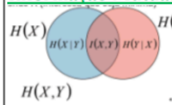
Informação: **i(a)** = $\log_2(1/P(a))$;
i(a,c) = i(a) + i(c);

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))

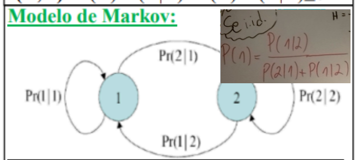
Informação Mútua: diminui com o aumento da prob.



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)= - $P(1|1)\log_2P(1|1)$ – $P(2|1)\log_2P(2|1)$
H(2)= - $P(2|2)\log_2P(2|2)$ – $P(1|2)\log_2P(1|2)$

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

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;

$$[x\log_2(x)]' = \log_2(x) + \frac{1}{\ln(x)}$$

Arvores 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

Aritmético Normal
se o intervalo atual:
-Estiver contido em [0,0.5[Tx = 0; remapear para E1(x) = 2x
-Estiver contido em [0,5,1[Tx = 1; remapear para E2(x) = 2(x-0.5)
-Estiver contido em [0.25,0.75[Tx = 10 se o próximo for E2 / Tx = 01 se o próximo for E1
-Para n E3 consecutivos, esperar pelo próximo E1/E2 e transmitir o bit correspondente seguido de n bits contrários

$$n \ge \frac{\log 2(|A_{cifra}|^{|cifra|})}{\log 2(|A_{cifra}|^{|cifra|}) - \frac{H(X)}{|mensagem|}}$$

Aritmético Inteiro

$$|Bits| = 2^{|String|}$$

$$F(a_k) = \frac{CumCounts(a_k)}{TotalCounts}$$

$$l_i = l_{i-1} + \lfloor (u_{i-1} - l_{i-1} + 1)\frac{CumCounts(a_{k-1})}{TotalCounts} \rfloor$$

$$u_i = l_{i-1} + \lfloor (u_{i-1} - l_{i-1} + 1)\frac{CumCounts(a_k)}{TotalCounts} \rfloor - 1$$

$E_3, E_2 \Leftrightarrow E_2, E_1$
$E_3, E_1 \Leftrightarrow E_1, E_2$
$E_3, E_3, E_3, E_2 \Leftrightarrow E_2, E_1, E_1, E_1$

E1/E2 - MSB igual em ambos: Tx = MSB -> shift left com 0 em L e com 1 em U
E3 - (MSB diferente em ambos + 2ºMSB diferente em ambos) -> NumE3++ ->
-> complementar 2ºMSB em ambos e shift tal como em E1/E2 ->
-> no próximo E1 ou E2 -> Tx = MSB | !MSB * NumE3 -> reset NumE3

Propriedades dos restos:

$(n+id) \bmod d = n \bmod d$
$(n_1 \pm n_2) \bmod d = ((n_1 \bmod d) \pm (n_2 \bmod d)) \bmod d$
$(n_1 n_2) \bmod d = ((n_1 \bmod d)(n_2 \bmod d)) \bmod d$

$a^{\varphi(n)}$ 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)

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

Teorema de Shannon – Segredo Perfeito:

Encriptador é perfeito
H(Y|XRZS)=0
R = RANDOM SEED
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!

x-mensagem
z-chave
y-mensagem encriptada

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 **ba**
{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 barray**ar#** (…)

O LZW é melhor, visto não ser preciso escrever os codificados sendo apenas necessário escrever os índices.
LZ8(indice+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

numero grande é provavelmente primo?
2^c mod p
c em potencia base 2 ->
ciclo mod (2^2^n-1)^2 mod p até n maximo ->
multiplicar os componentes de c

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

RSA:
2 números primos (p e q);
n = p*q ;
φ(n) = (p-1)(q-1);
expoente e < n em que mdc(e,(p-1)(q-1))=1 ; -usa public key
expoente de desencriptação ed mod(p-1)(q-1)=1 ; -usado para ñ repúdio
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^{2^2} mod 101 = 16 mod 101 = 16; 2^{2^3} mod 101 = 16^2 mod 101 = 54

C. seguro: quebra > valor inform e temp quebra > vida info
assimétrico: +computacional

Fontes naturais
– Ruído térmico
– Radioactividade
– Lançamento de moeda

Há sempre formas para obter informação:
- Captura de prisioneiros

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: $\sum_i^n D^{-l_i} = 1$

$$J(p)=\sum_{i=1}^{4} p(i)\log_2 p(i)+\alpha\left(\sum_{i=1}^{4} p(i)-1\right)+\beta\left(\sum_{i=1}^{4} c_i p(i)-2.5\right)$$
$$\frac{\partial}{\partial p(i)}J(p)=\left(\log_2 p(i)+\frac{1}{\ln 2}\right)+\alpha+\beta c_i=0$$
$$p(i)=2^{-\alpha}2^{-\beta c_i}2^{-\frac{1}{\ln 2}}$$
$$l=\left\lceil \log_2\frac{1}{P(x)}\right\rceil+1=\left\lceil \log_2\frac{1}{u^m-l^m}\right\rceil+1\ bits$$

contexto ordem n -> tabela contexto -1 (com todos os símbolos count 1 e cum_counts) -> tabela contexto 0 (apenas esc no início, vão sendo adicionados os simbs vistos pela primeira vez -> tabela contexto 1 (para cada contexto há uma subtabela com todos os símbolos que já apareceram após esse contexto, bem como esc em cada uma). Alg: ler tabela contexto n, se existe o contexto atual, codificamos com aritméticos (inteiros) usando cum_count/totalcount dessa subtabela -> caso não exista, codificamos ESC dessa tabela e tentamos encontrar contexto para n-=1;