

Introdução à Teoria da Informação

Prof. Derzu Omaia
Prof. Leonardo Vidal Batista
CI/DI/PPGI/UFPB
derzu@ci.ufpb.br

Teoria da Informação

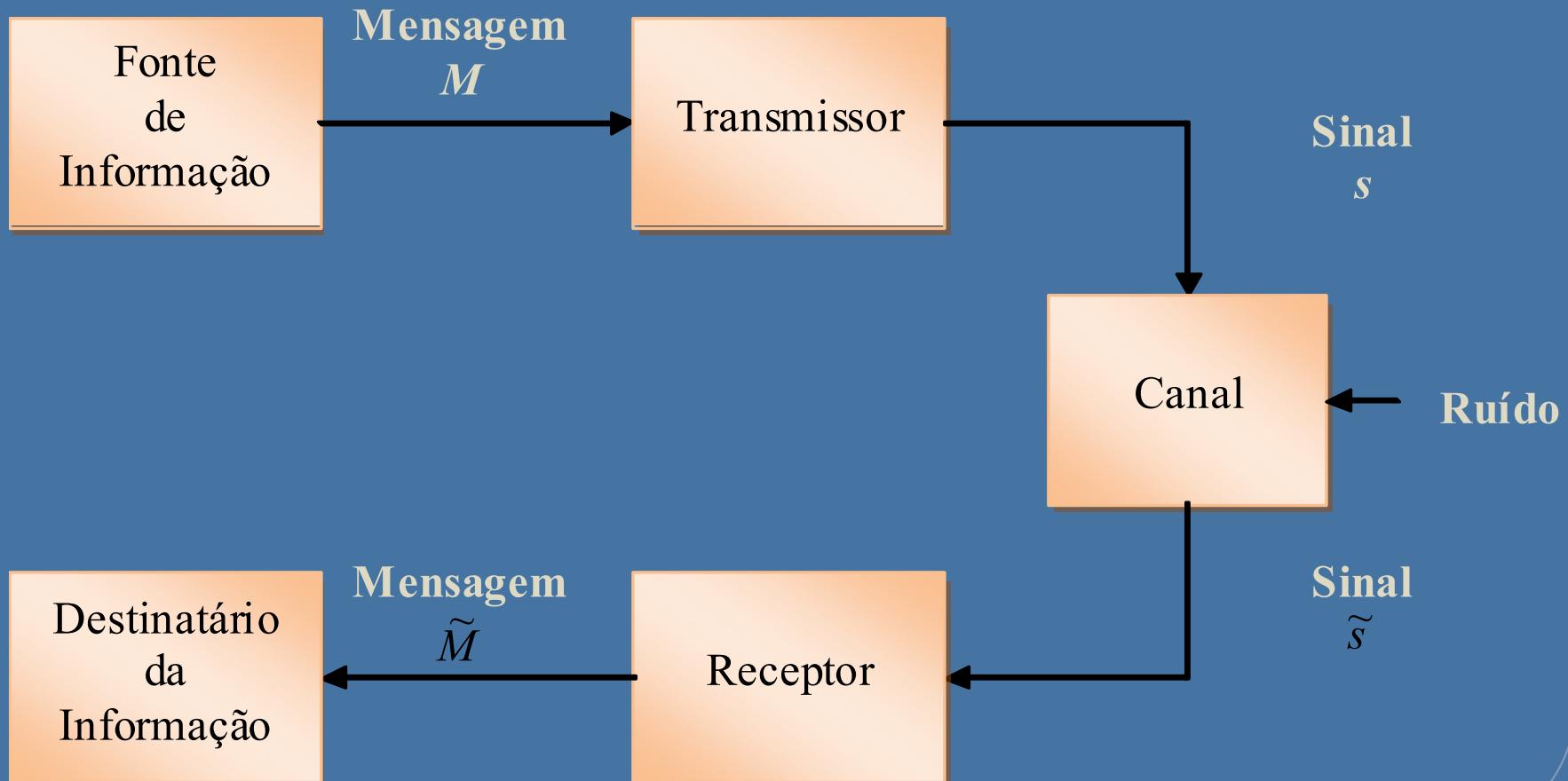
- Claude Shannon, 1948, “A Mathematical Theory of Communications”
- Área que estuda os processos de comunicação
- Alicerçada na conceituação formal de *informação* e da capacidade de transmissão de informação pelos canais de comunicação
- Compressão de Dados: aplicação da Teoria da Informação.

Claude Shannon

- 1916-2001
- Pai da Era da Comunicação Eletrônica
- Mestrado: lógica booleana aplicada à otimização de circuitos elétricos
- Pioneiro de IA, robótica, criptografia, etc



Sistemas de comunicação



Fonte de Informação

- Fonte de Informação produz mensagens = sequências de elementos selecionados de um conjunto (alfabeto)

$$A = \{a_0, a_1, a_2, \dots, a_{M-1}\}$$

- A: alfabeto da fonte
- a_i : símbolo
- M: cardinalidade ou tamanho do alfabeto

Questões

- Pode-se garantir comunicação perfeita?
- Pode-se garantir comunicação confiável?
- Comunicação equivale a "transmissão de dados"?

Respostas de Shannon:

- Não
- Sim. Pode-se projetar e construir sistemas de comunicação tão confiáveis quanto se deseje: 1 bit errado em 1 bilhão, 1 em 1 trilhão,...
- Não. Comunicação envolve transmissão de informação.
 - Compressores sem perdas reduzem a quantidade de dados em uma mensagem, preservando a informação

Informação

- Dados é relativo a forma como a mensagem está codificada.
- Mensagem é a sequência de símbolos de uma fonte.
- Informação é relativa, depende do conteúdo da mensagem e de como ele é previsto pelo receptor.

O Que É “Informação”?

- Informação (ou entropia) é uma medida associada à incerteza ou surpresa relacionada a um evento (ou relacionada ao próximo símbolo da mensagem)
- O processo de comunicação é inherentemente probabilístico
- Informação é relativa!
 - Depende do modelo preditivo de quem recebe a mensagem.

Relatividade da Informação

- 010101010101...
- Pedro Álvares Cabra...
- Marcela amou-me durante quinze meses e onze ...
- 14159265...
- 27033903618375114...

Informação e Incerteza

- Tarefa para a próxima aula: decorar letra por letra o livro “Os Sertões” de Euclides da Cunha
- Na próxima aula, farei apenas a leitura de “Os Sertões”
- Quem virá?
- Transmitirei informação?
- Ocorrerá comunicação?

Entropia da Língua Inglesa

- Jogo de Shannon:



Entropia da Língua Inglesa

- Jogo de Shannon:

- Cada letra tem um probabilidade baseada nas quantidade de tentativas.
- Exemplo:
 - <http://www.math.ucsd.edu/~crypto/java/ENTROPY/>
 - <https://mathweb.ucsd.edu/~crypto/java/ENTROPY/>

Teoria da Informação e Física

- A equação de entropia (informação) de Shannon é análoga à entropia da termodinâmica.



Modelagem e Compressão

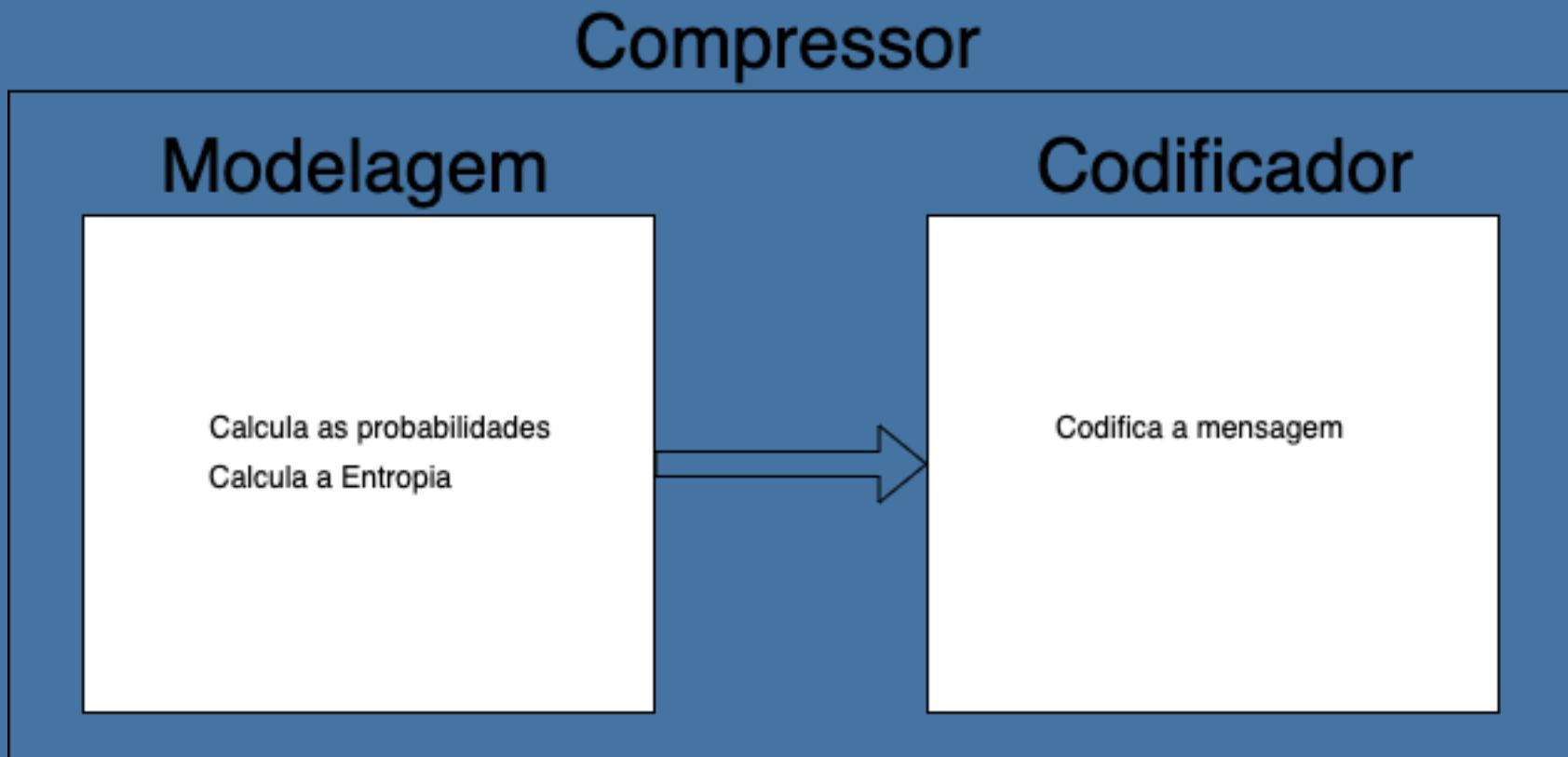
- Bom modelo prevê com precisão símbolos sucessivos da mensagem.
- Bom modelo => surpresa reduzida => entropia reduzida => grau de compressão elevado
- Grau de compressão elevado => entropia reduzida => surpresa reduzida => Bom modelo
- Bom modelo \Leftrightarrow Grau de compressão elevado
- Grau de compressão é uma medida da qualidade de um modelo

Entropia e Compressão

- Procurar construir modelos que reduzam a entropia:
 - Entropia elevada => grau de compressão reduzido
 - Entropia reduzida => grau de compressão elevado
- A entropia é um limite para o grau de compressão máxima que se pode obter.

Entropia e Compressão

- Forma geral:



Entropia da Língua Inglesa

- Qual é a entropia da Língua Inglesa?
- **O que é a língua inglesa?**
- Qual é a entropia da língua inglesa nos livros de Swift?

Entropia da Língua Inglesa

- Qual é a entropia da língua inglesa nos livros de Swift?
 - Por um especialista em Literatura Inglesa?
 - Por um engenheiro inglês?
 - Por um modelo computacional?
 - Por uma criança recém-alfabetizada em inglês?

Entropia da Língua Inglesa

- Shannon, 1952: entropia da "língua inglesa" escrita, percebida por americanos adultos ≈ 1 bit/símbolo
- Estudos posteriores confirmaram a estimativa
- Melhores modelos computacionais: aprox. 1,2 bit/símbolo
- Quando um modelo computacional tornar-se tão bom quanto um humano, o computador terá "entendido" os textos??

Entropia da Língua Inglesa

- Aplicações para um bom modelo computacional de um idioma:
 - OCR; conversão de voz para texto; atribuição de autoria etc.

Entropia de outras mensagens

- Língua portuguesa
- DNA
- IBOVESPA
- Dados sísmicos
- Imagens
- Vídeo
- Etc

Compressão para não Comprimir!

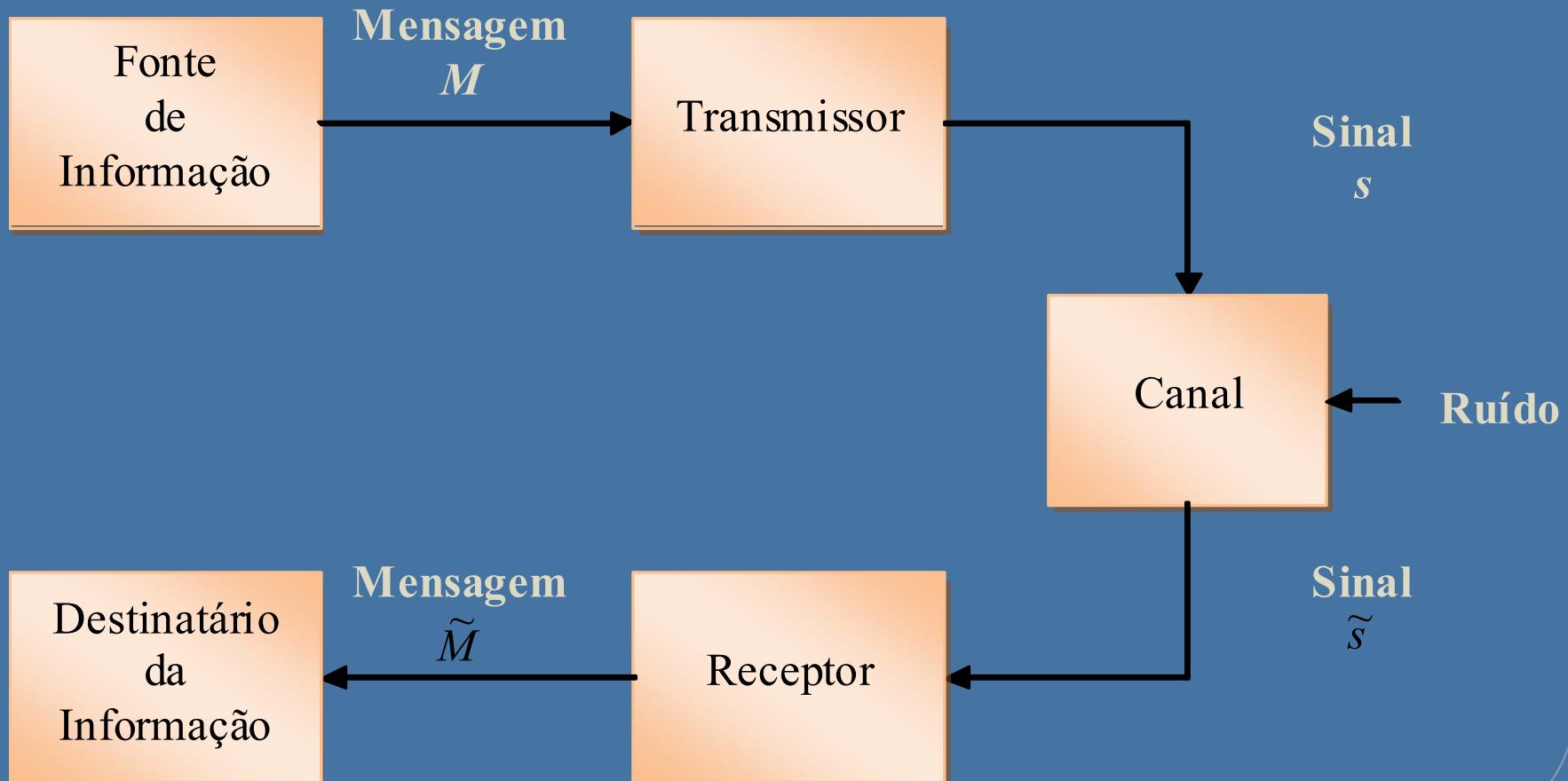
- Compressores modernos possuem excelente capacidade de modelagem estatística
- Modelos podem ser utilizados para outros fins que não compressão
 - Reconhecimento biométrico
 - Classificação de microorganismos
 - Segmentação e classificação de texturas
 - Atribuição de autoria
 - Etc
- Grau de compressão: qualidade do modelo!

Compressão para Comprimir!

- Tecnologia-chave em:

- DVDs
- TV digital
- MP3 (mostrar players)
- Stream de música (Deezer, Spotify)
- Stream de vídeo (YouTube, Netflix)
- Câmeras fotográficas e de vídeo
- Redes de comunicação
- Etc

Sistemas de comunicação

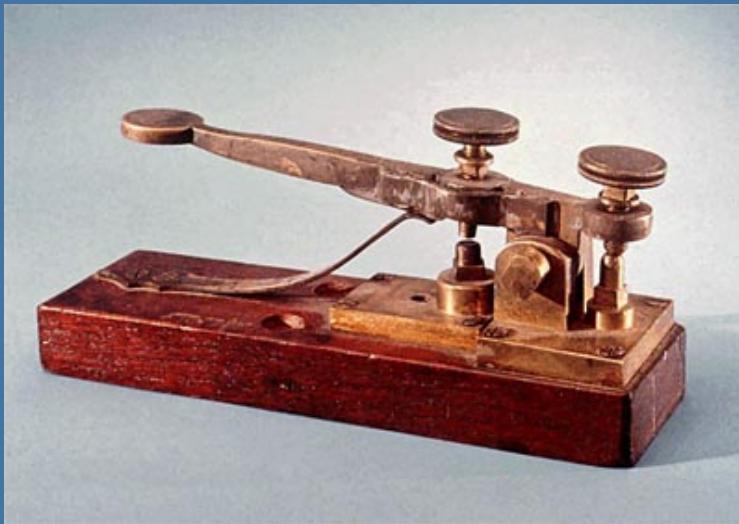


Códigos de Propósito Específico

- Códigos de transmissão: Morse, códigos corretores de erros, etc
- Códigos de criptografia
- Códigos de compressão: representação reduzida da informação
 - Códigos de compressão binários: menor número de bits possível

Códigos de Transmissão

- Equipamentos para transmissão de Morse:



Telegrafo



Lanterna



Tambor

- Ver Vídeo

Códigos de Transmissão

○ Tabela Código Morse:

- Proposto por Samuel Morse em 1835
- Traço: sinal longo
- Bola: sinal curto
- Sinais separados por espaço

Códigos de Transmissão

○ Tabela Código Morse:

A ● -
B - ● ● ●
C - ● - ●
D - ● ●
E ●
F ● ● - ●
G - - ●
H ● ● ● ●
I ● ●

J ● - - -
K - ● -
L ● - ● ●
M - -
N - ●
O - - -
P ● - - ●
Q - - ● -
R ● - ●

S ● ● ●
T -
U ● ● -
V ● ● ● -
W ● - -
X - ● ● -
Y - ● - -
Z - - ● ●

Códigos de Criptografia

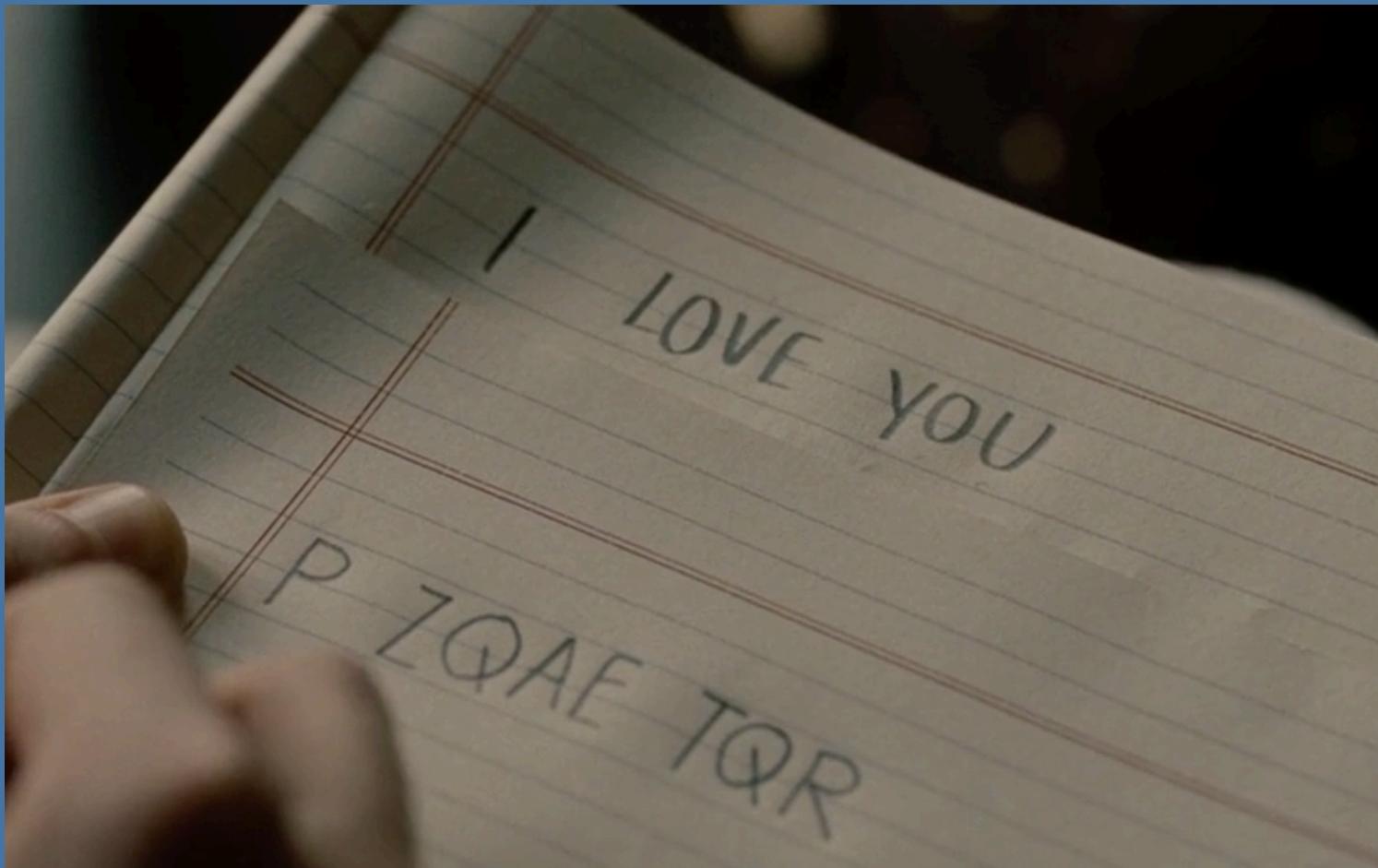
- Jogo da Imitação:
- Em 1939, a recém-criada agência de inteligência britânica MI6 recruta Alan Turing, um aluno da Universidade de Cambridge, para entender códigos nazistas, incluindo o "Enigma", que criptógrafos acreditavam ser inquebrável. A equipe de Turing, incluindo Joan Clarke, analisa as mensagens de "Enigma", enquanto ele constrói uma máquina para decifrá-las. Após desvendar as codificações, Turing se torna herói. Porém, em 1952, autoridades revelam sua homossexualidade, e a vida dele vira um pesadelo.

Códigos de Criptografia

- Exemplos Jogo da Imitação:
 - P ZQAE TQR
 - WII CSY MR XAS PSRK
AIIOW HIEVIWX JVMIRH
 - Caracteres deslocados por um determinado valor.

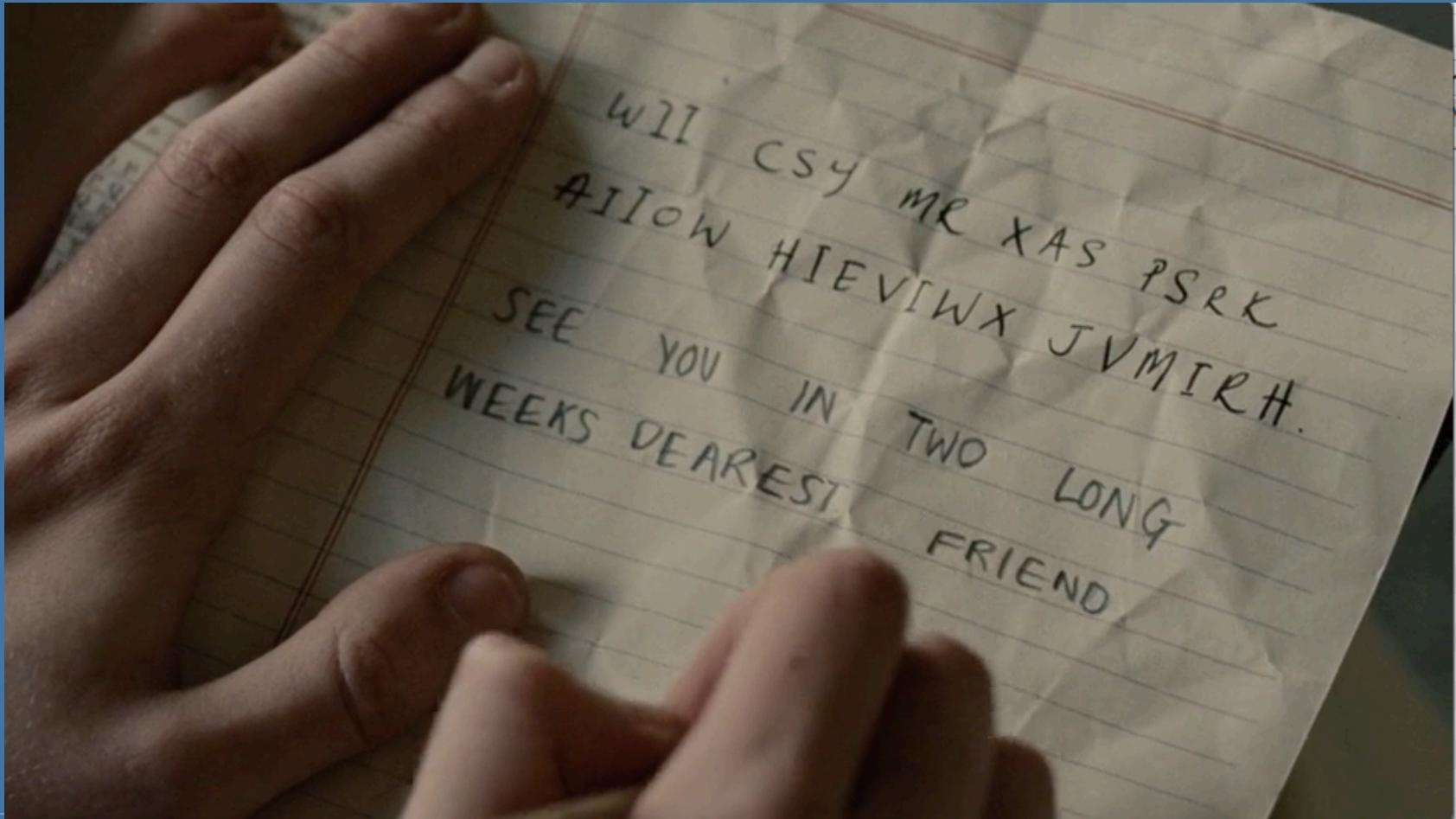
Códigos de Criptografia

- Exemplos:



Códigos de Criptografia

- Exemplos:



Codificação

- Codificação: representação da mensagem
- Palavra-código: representação de um símbolo ou sequência de símbolos
- Código de fonte: conjunto de palavras-código capaz de representar qualquer mensagem emitida pela fonte

Codificador e Decodificador

- Codificador: elemento (ser humano, circuito, programa, etc) que representa as mensagens geradas pela fonte empregando um código específico.
- Decodificador: elemento responsável por desfazer o mapeamento realizado por um codificador.

Fonte de Informação

- Fonte de Informação produz mensagens = sequências de elementos selecionados de um conjunto (alfabeto)

$$A = \{a_0, a_1, a_2, \dots, a_{M-1}\}$$

- A: alfabeto da fonte
- a_i : símbolo
- M: cardinalidade ou tamanho do alfabeto

Exemplos: fontes decimais e binárias

- **Fonte de dígitos decimais**

- $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Mensagem $M = 590126885122380546$

- **Fonte octal:**

- $A = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Mensagem $M = 6370171235130$

Códigos de Compressão

- Em termos de compressão, qual é o melhor código binário?

Decimal	Binário 1	Binário 2	Binário 3
0	0000_2	00_2	111111_2
1	0001_2	01_2	111110_2
2	0010_2	10_2	111101_2
3	0011_2	1100_2	111100_2
4	0100_2	1101_2	1110_2
5	0101_2	1110_2	1101_2
6	0110_2	111100_2	1100_2
7	0111_2	111101_2	10_2
8	1000_2	111110_2	01_2
9	1001_2	111111_2	00_2

Códigos de Compressão

- Resposta: Depende da frequência relativa (probabilidade) com que a fonte produz cada símbolo.
- Ou da frequência de cada símbolo para uma determinada mensagem.

Códigos de Compressão

- Binário l é a codificação padrão sem compressão.
- Regras para gerar a codificação padrão:
 - Todas as palavras códigos devem ser diferentes;
 - Todas as palavras código com mesmo tamanho;
 - O tamanho das palavras códigos deve ser o menor possível.

Exemplo 1: codificação-decodificação

- Codificar e decodificar:
- $A = \{R, G, B\}$
- $M = RGBRBRR$, usando $RGB1, RGB2, RGB3$

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha (R)	00	0	0
Cor verde (G)	01	1	10
Cor azul (B)	10	00	11

- $RGB1$
 - Codificar: 00011000100000 (14 bits)
 - Decodificar: $RGBRBRR$

Exemplo 1 (cont.)

- Codificar e decodificar:
- $A = \{R, G, B\}$
- $M = RGBRBRR$, usando RGB1, RGB2, RGB3

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha (R)	00	0	0
Cor verde (G)	01	1	10
Cor azul (B)	10	00	11

- RGB2
 - Codificar: 010000000 (9 bits)
 - Decodificar: RG ?

Exemplo 1 (cont.)

- Codificar e decodificar:
- $A = \{R, G, B\}$
- $M = RGBRBRR$, usando $RGB1, RGB2, RGB3$

Símbolo	Código RGB1	Código RGB2	Código RGB3
Cor vermelha (R)	00	0	0
Cor verde (G)	01	1	10
Cor azul (B)	10	00	11

- $RGB3$
 - Codificar: 0101101100 (10 bits)
 - Decodificar: $RGBRBRR$

Exemplo 1 (cont.)

- Comprimento (length) Médio:

- $L_m = \text{comprimento em bits} / \text{qtd. símbolos msg.}$

Exemplo 1 (resumo)

- Comprimento RGB1: 14bits
 - $Lm = 14/7 = 2\text{bits/símbolo}$
- Comprimento RGB2: 9bits
 - $Lm = 9/7 = 1,29\text{bits/símbolo}$
- Comprimento RGB3: 10bits
 - $Lm = 10/7 = 1,43\text{bits/símbolo}$

Códigos de Prefixo

- RGB1 e RGB3 são códigos de prefixo:
nenhuma palavra-código é prefixo de outra
 - Códigos unicamente decodificáveis.
 - Tabela do código morse.
- RGB2 não é um código de prefixo.
- RGB1 é sem compressão, codificação padrão.

Exemplo 2: comprimento médio

- Codificar mensagem W usando Bin2
- $W = \begin{matrix} 1 & 0 & 2 & 4 & 5 & 0 & 0 & 1 & 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 7 & 0 & 3 & 0 & 6 & 0 & 0 & 9 & 0 & 0 & 0 & 2 & 4 & 5 & 0 \\ 2 & 3 & 3 & 0 & 0 & 4 & 0 & 5 & 0 & 0 & 6 & 0 & 0 & 7 & 8 \\ 6 & 0 & 0 & 0 & 0 & 7 & 6 & 8 & 8 & 0 & 0 & 0 & 7 & 8 & 5 \end{matrix}$
- Alfabeto: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Comprimento da mensagem codificada, em bits?
- Comprimento médio, em bits/símbolo?

Binário 2
00_2
01_2
10_2
1100_2
1101_2
1110_2
111100_2
111101_2
111110_2
111111_2

Exemplo 2: comprimento médio

- $W = \begin{array}{cccccccccc} 1 & 0 & 2 & 4 & 5 & 0 & 0 & 1 & 0 & 2 \\ 7 & 0 & 3 & 0 & 6 & 0 & 0 & 9 & 0 & 0 \\ 2 & 3 & 3 & 0 & 0 & 4 & 0 & 5 & 0 & 0 \\ 6 & 0 & 0 & 0 & 7 & 6 & 8 & 8 & 0 & 0 \end{array} \begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 2 & 4 & 5 & 0 & 0 \\ 6 & 0 & 0 & 7 & 8 \\ 8 & 5 & 0 & 0 & 7 \end{array}$
- $W_c = \begin{array}{cccccccccccccccccccccccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}$
- Total de bits:
 - $L = 192$ bits
 - $L_m = 192/60 = 3,2$ bits/símbolo

Exemplo 2: comprimento médio

- $N = 60$ símbolos
- N_i : N° de ocorrências de a_i
 - $N_0 = 29; N_1 = 4; N_2 = 4; N_3 = 3;$
 $N_4 = 3; N_5 = 4; N_6 = 4; N_7 = 4;$
 $N_8 = 4; N_9 = 1$
- b_i : N° de bits da palavra-código associada a a_i
 - $b_0 = b_1 = b_2 = 2;$
 - $b_3 = b_4 = b_5 = 4;$
 - $b_6 = b_7 = b_8 = b_9 = 6$

Binário 2
00_2
01_2
10_2
1100_2
1101_2
1110_2
111100_2
111101_2
111110_2
111111_2

Exemplo 2: comprimento médio

- Comprimento:

$$l = \sum_{i=0}^{M-1} N_i b_i$$

- Comprimento médio:

$$\bar{l} = \frac{1}{N} \sum_{i=0}^{M-1} N_i b_i$$

$$A = \{a_0, a_1, \dots, a_{M-1}\}$$

N: Nº de símbolos da mensagem

N_i: Nº de ocorrências de a_i

b_i: Nº de bits da palavra-código associada a a_i

M: Cardinalidade alfabeto

Exemplo 2: comprimento médio

- Total de bits:

- $l = b_0N_0 + b_1N_1 + \dots + b_9N_9$
- $l = 29 \times 2 + 4 \times 2 + 4 \times 2 + 3 \times 4 + 3 \times 4 + 4 \times 4 + 4 \times 6 + 4 \times 6 + 4 \times 6 + 1 \times 6 = 192$ bits

- Número médio de bits por símbolo:

- $\bar{l} = (29 \times 2 + 4 \times 2 + 4 \times 2 + 3 \times 4 + 3 \times 4 + 4 \times 4 + 4 \times 6 + 4 \times 6 + 4 \times 6 + 1 \times 6) / 60$
 $= (29/60) \times 2 + (4/60) \times 2 + (4/60) \times 2 + (3/60) \times 4 + (3/60) \times 4 + (4/60) \times 4 + (4/60) \times 6 + (4/60) \times 6 + (4/60) \times 6 + (1/60) \times 6$
 $= 3,2$ bits/símbolo.

Exemplo 2: comprimento médio

- Para alfabeto $A = \{a_0, a_1, \dots a_{M-1}\}$
- N_i : N° de ocorrências de a_i
- b_i : N° de bits da palavra-código associada a a_i

$$\bar{l} = \frac{1}{N} \sum_{i=0}^{M-1} N_i b_i \Rightarrow \bar{l} = \sum_{i=0}^{M-1} \frac{N_i}{N} b_i ; P_i = \frac{N_i}{N} \Rightarrow$$

$$\bar{l} = \sum_{i=0}^{M-1} P_i b_i$$

Exemplo 2: comprimento médio

- Para alfabeto $A = \{a_0, a_1, \dots, a_{M-1}\}$
- P_i = estimativa da probabilidade de a_i
= freqüência relativa de $a_i = N_i/N$

$$\bar{l} = P_0 b_0 + P_1 b_1 + \dots + P_{M-1} b_{M-1}$$

$$\bar{l} = \sum_{i=0}^{M-1} P_i b_i$$

Exemplo 3: comprimento médio

- Codificar a mensagem W usando Bin3

- $b_0 = b_1 = b_2 = b_3 = 6;$
 $b_4 = b_5 = b_6 = 4;$
 $b_7 = b_8 = b_9 = 2$
- $\bar{l} = 29x6 + 4x6 + 4x6 + 3x6 + 3x4 + 4x4 + 4x4 + 4x2 + 4x2 + 2x1)/60 = 302/60 = 5,03 \text{ bits/símbolo}$

- Para compressão, qual é o melhor código, BIN2 ou BIN3?

Binário 3
111111 ₂
111110 ₂
111101 ₂
111100 ₂
1110 ₂
1101 ₂
1100 ₂
10 ₂
01 ₂
00 ₂

Exemplo 3: comprimento médio

- Codificar a mensagem W usando Bin1 ?

Binário 1
0000_2
0001_2
0010_2
0011_2
0100_2
0101_2
0110_2
0111_2
1000_2
1001_2

Comprimento médio

- Resumo:
- Binário1 = 4 bits/símbolo (sem compressão, cod. padrão)
- Binário2 = 3,2 bits/símbolo
- Binário3 = 5,03 bits/símbolo

- Há algum código de prefixo melhor?

Comprimento médio

Exemplo código de Huffman:

Símbolo	Código de Huffman	Comprimento Palavra-código
0	1	1
1	0111	4
2	0110	4
5	0101	4
6	0100	4
7	0011	4
8	0010	4
3	0001	4
4	00001	5
9	00000	5

Comprimento médio

- $I_{Huff} = 29 \times 1 + 4 \times 4 + 4 \times 4 + \dots + 1 \times 5 = 157$ bits
- Compr. Médio = $(29 \times 1 + 4 \times 4 + 4 \times 4 + \dots + 1 \times 5)/60 = 2,62$ bits/símbolo.

Comprimento médio

- Resumo:
- Binário1 = 4 bits/símbolo (sem compressão, cod. padrão)
- Binário2 = 3,2 bits/símbolo
- Binário3 = 5,03 bits/símbolo
- Huffman = 2,62 bits/símbolo

Problema Essencial da Compressão

- Dado um modelo probabilístico para a fonte, encontrar um código unicamente decodificável ótimo, ou seja, que represente a saída da fonte com o menor número de bits possível.
- Encontrar b_i de forma a minimizar (com restrição de decodificabilidade)

$$\bar{l} = \sum_{i=0}^{M-1} P_i b_i$$

Informação

- Solução: $b_i = \text{Informação } I_i$ associada ao símbolo a_i :

$$b_i = I_i = \log \frac{1}{P_i}$$

- Se \log_2 : informação em bits
- Se \log_3 : informação em trits
- Se \ln : informação em nats
- Se \log_{10} : informação em dígitos decimais

Entropia

$$H = \bar{l}_{\min} = \sum_{i=0}^{M-1} P_i \log_2 \frac{1}{P_i}$$

- Unidade (para base 2): bits/símbolo
- Entropia da fonte ou da mensagem?
- Shannon, 1948: entropia e compressão
- Huffman, 1952: codificação ótima de comprimento inteiro
- Pesquisadores da IBM, 1978: codificação ótima (codificação aritmética)

Entropia

$$H = \bar{l}_{\min} = \sum_{i=0}^{M-1} P_i \log_2 \frac{1}{P_i}$$

- Entropia de ordem 1
- Modelo não contextual semi-adaptativo ou estático.

Exemplo 4: Informação e entropia

• W = 1 0 2 4 5 0 0 1 0 2 1 1 0 0 0 7 0 3 0 6
0 0 9 0 0 0 2 4 5 0 2 3 3 0 0 4 0 5 0 0
6 0 0 7 8 6 0 0 0 0 7 6 8 8 0 0 0 7 8 5

$$P_0 = 29/60 \Rightarrow I_0 = \log_2 60/29 = 1,05 \text{ bit}$$

$$P_1 = P_2 = P_5 = P_6 = P_7 = P_8 = 4/60$$

$$\Rightarrow I_1 = I_2 = I_5 = I_6 = I_7 = I_8 = \log_2 60/4 = 3,91 \text{ bits}$$

$$P_3 = P_4 = 3/60 \Rightarrow I_3 = I_4 = \log_2 60/3 = 4,32 \text{ bits}$$

$$P_9 = 1/60 \Rightarrow I_9 = \log_2 60 = 5,91 \text{ bits}$$

$$H = (29/60)*1,05 + (4/60)*3,91 + \dots \\ + (1/60)*5,91 = 2,60 \text{ bits/símbolo}$$

Exemplo 4: Informação e entropia

- Comparar $H = 2,60$ bits/símbolo com codificação por Bin2 e Bin3 (3,2 bits/símbolo e 5,03 bits/símbolo)
- Código eficiente? Comparar sempre com a entropia!

Código de Huffman

- Código de comprimento inteiro ótimo
- Para mensagem dos exemplos anteriores:

Símbolo	Código de Huffman	Comprimento Palavra-código	Informação
0	1	1	1,05
1	0111	4	3,91
2	0110	4	3,91
5	0101	4	3,91
6	0100	4	3,91
7	0011	4	3,91
8	0010	4	3,91
3	0001	4	4,32
4	00001	5	4,32
9	00000	5	5,91

Código de Huffman

Decimal	Bin 1	Bin 2	Bin 3	Bin Huff
0	0000_2	00_2	111111_2	1_2
1	0001_2	01_2	111110_2	0111_2
2	0010_2	10_2	111101_2	0110_2
3	0011_2	1100_2	111100_2	0001_2
4	0100_2	1101_2	1110_2	00001_2
5	0101_2	1110_2	1101_2	0101_2
6	0110_2	111100_2	1100_2	0100_2
7	0111_2	111101_2	10_2	0011_2
8	1000_2	111110_2	01_2	0010_2
9	1001_2	111111_2	00_2	00000_2

Código de Huffman

- $I_{Huff} = 29 \times 1 + 4 \times 4 + 4 \times 4 + \dots + 1 \times 5 = 157$ bits
- Compr. Médio = $(29 \times 1 + 4 \times 4 + 4 \times 4 + \dots + 1 \times 5)/60 = 2,62$ bits/símbolo.
- Entropia $H = 2,60$
- Eficácia?

Comprimento médio e RC

- $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $M = W$
- Binário1 (sem compressão, cod. padrão)
 - $L_m = 4$ bits/símbolo
- Binário2:
 - $L_m = 3,2$ bits/símbolo
- Binário3:
 - $L_m = 5,03$ bits/símbolo
- Huffman:
 - $L_m = 2,62$ bits/símbolo
- Entropia:
 - $L_m = 2,60$ bits/símbolo

Razão de Compressão

- Razão de Compressão (RC):
 - $RC = \text{Comprimento sem compressão} / \text{Comprimento com compressão}$
- Exemplo:
 - $RC = 4/2,62 = 1,53 \ (1,53:1)$

Comprimento médio e RC

- $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $M = W$
- Binário 1 (sem compressão, cod. padrão)
 - $Lm = 4$ bits/símbolo
 - $RC = 4/4 = [1:1]$
- Binário 2:
 - $Lm = 3,2$ bits/símbolo
 - $RC = 4/3,2 = [1,25:1]$
- Binário 3:
 - $Lm = 5,03$ bits/símbolo
 - $RC = 4/5,03 = [0,79:1]$
- Huffman:
 - $Lm = 2,62$ bits/símbolo
 - $RC = 4/2,62 = [1,53:1]$
- Entropia:
 - $Lm = 2,60$ bits/símbolo
 - $RC = 4/2,60 = [1,54:1]$

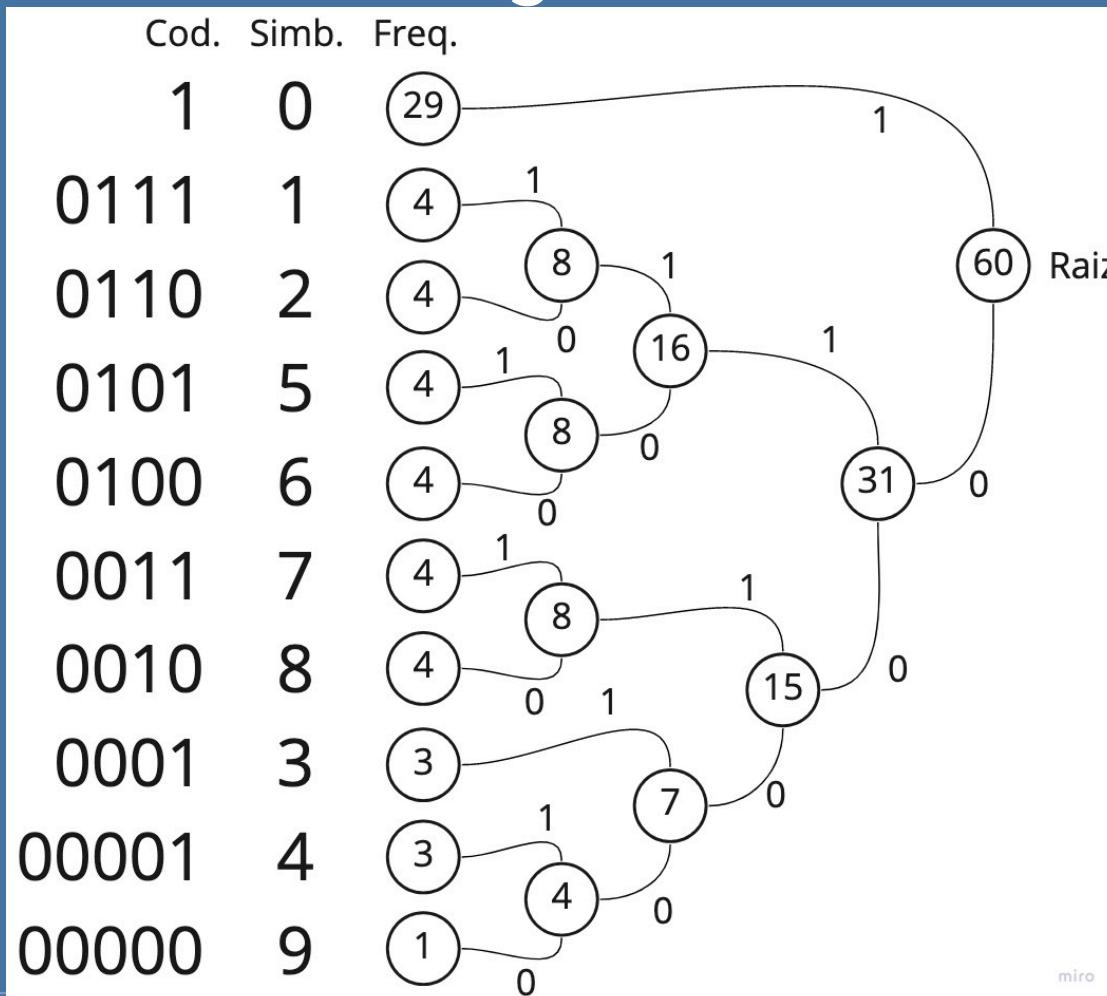
Código de Huffman

○ Árvores de Huffman:

- Nós livres são nós sem ascendentes.
- Cada símbolo do alfabeto gera um nó folha, com peso igual a probabilidade ou frequência.
- Enquanto não chegar na raiz:
 - Ordenar (decrescente) os nós livres de acordo seu peso.
 - Unir os 2 nós livres de menor peso. Gerando outro com a soma desses pesos.

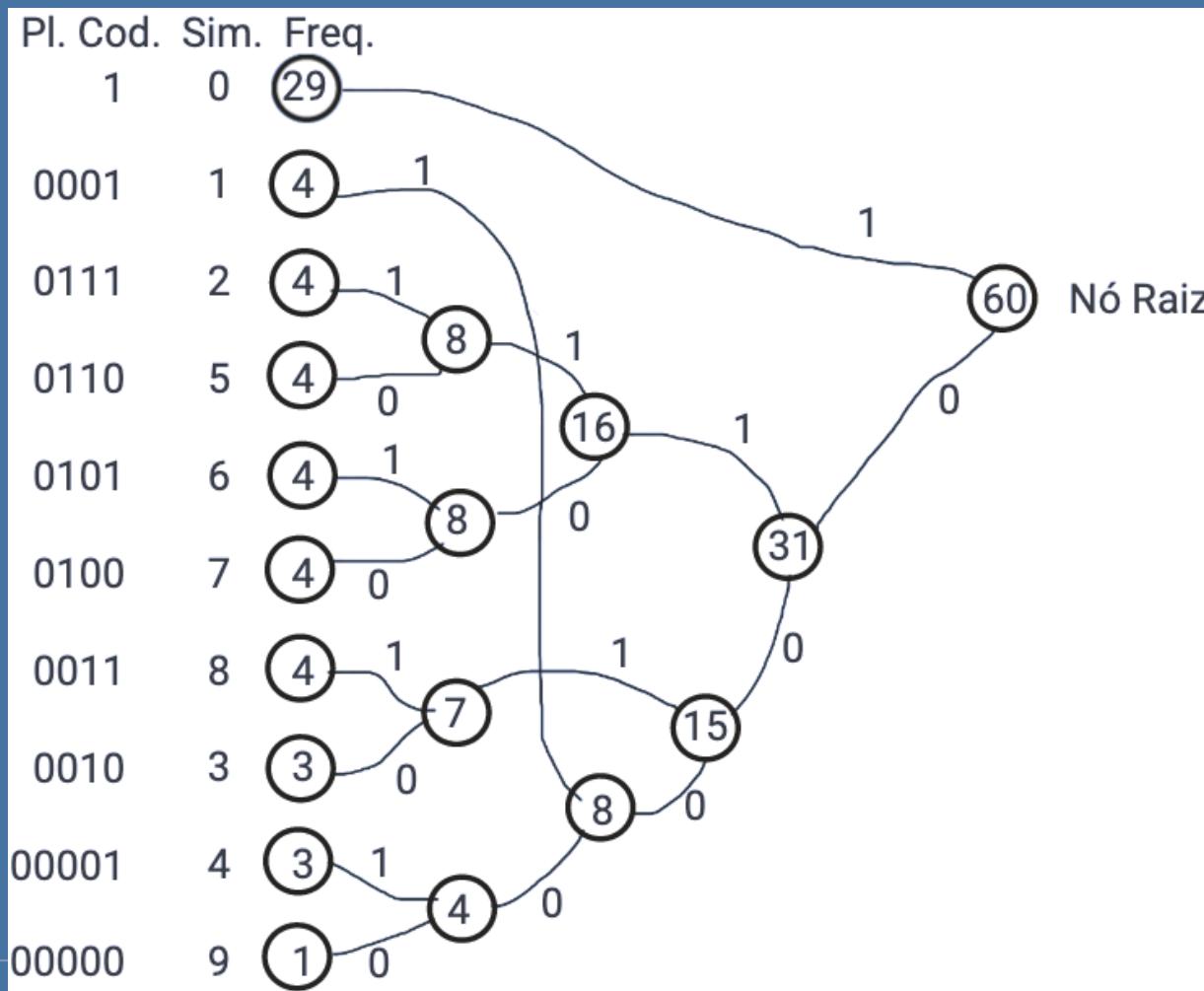
Código de Huffman

Para a mensagem W



Código de Huffman

- Para a mensagem W, versão 2



Código de Huffman

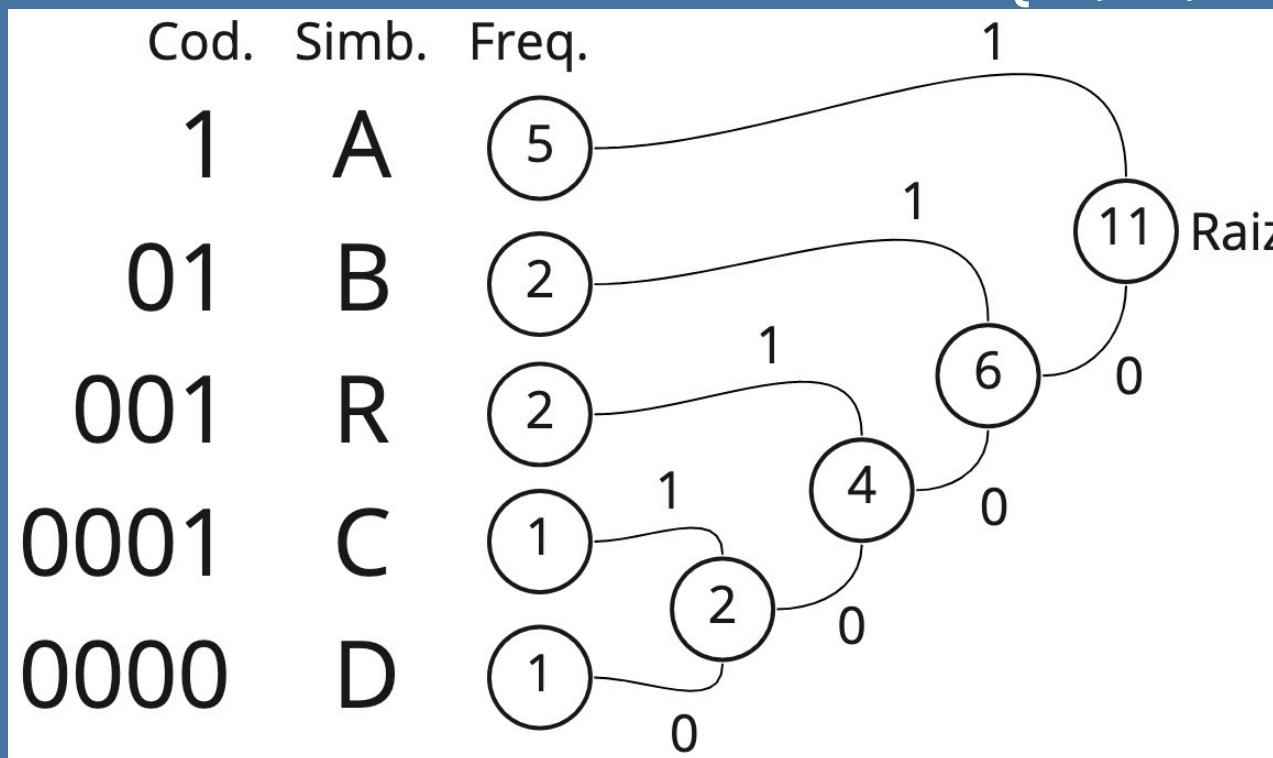
- $M = W; A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $Lm = 2,62 \text{ bits/símbolo}$
- $H = 2,60 \text{ bits/símbolo}$
- $RC = 4/2,62 = 1,53$

Código de Huffman

- M = ABRACADABRA
- A = {A, B, C, D, R}
 - Gerar o código de Huffman
 - Comprimento médio

Código de Huffman

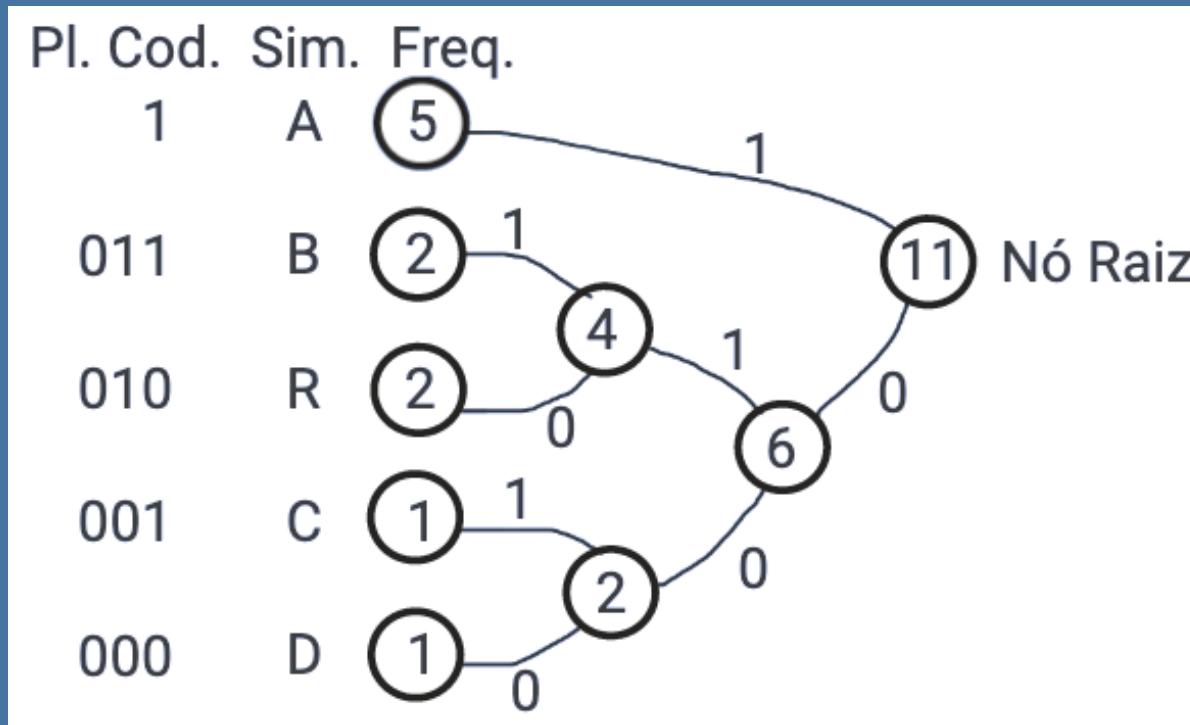
- $M = \text{ABRACADABRA}$ $A = \{\text{A, B, C, D, R}\}$



- $L = 5*1 + 2*2 + 2*3 + 1*4 + 1*4 = 23\text{bits}$
- $L_m = 23/11 = 2,09 \text{ bits/símbolo}$

Código de Huffman

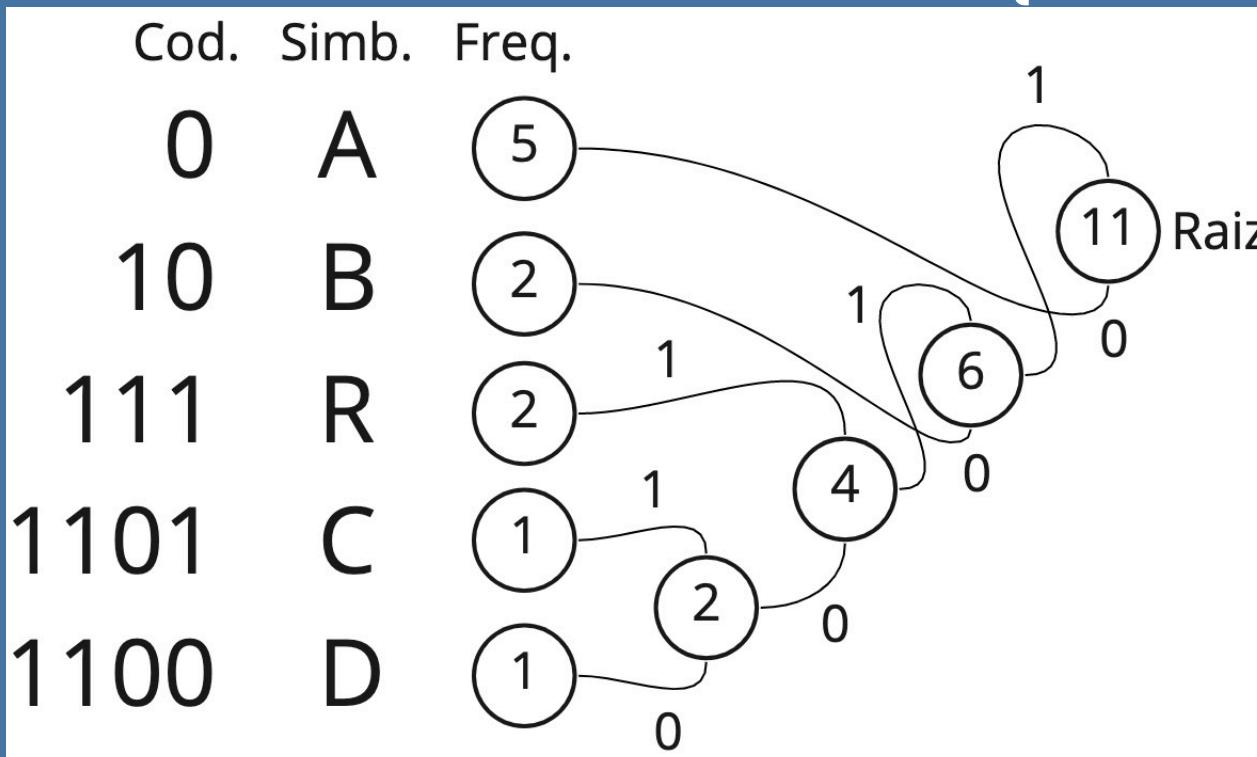
- $M = \text{ABRACADABRA}$ $A = \{\text{A, B, C, D, R}\}$



- $L = 5*1 + 2*3 + 2*3 + 1*3 + 1*3 = 23\text{bits}$
- $L_m = 23/11 = 2,09 \text{ bits/símbolo}$

Código de Huffman

- $M = \text{ABRACADABRA}$ $A = \{\text{A, B, C, D, R}\}$



- Trocou a posição devido a ordenação decrescente:

Código de Huffman

- $M = \text{ABRACADABRA}$ $A = \{A, B, C, D, R\}$
- $Lm = 2,09$ bits/símbolo
- $RC = 3/2,09 = [1,43:1]$
- $H = 2,04$

Código de Huffman

- Outras árvores de Huffman?
- Como quebrar empates?
- Que informação passar ao decodificador para permitir decodificação?
 - Tabela de códigos ou
 - Árvore ou
 - Contadores
- Quando o Huffman se torna ineficaz?
- O que ocorre se codificarmos por Huffman uma mensagem já codificada por Huffman?

Código de Huffman

- Quando o Huffman se torna ineficaz?
 - $P_i \gg 0,5$
- Redundância do código de Huffman

$$R_{Huff} = \bar{l}_{Huff} - H$$

- Pode-se mostrar que

$$R_{Huff} \leq 1$$

- Ou seja

$$\bar{l}_{Huff} \leq H + 1$$

Código de Huffman - Exemplo

- $A = \{a, b, c, d, r\}$
- $P(a) = 0,99; P(b)=P(c)=P(d)=0,003 , P(r) = 0,001$
- $I(a) = 0,0145; I(b) = I(c) = I(d) = 8,3817; I(r) = 9,9668$
- Comprimento médio de Huffman, entropia, eficácia, RC?

$$\bar{l}_{Huff} = 0,99 \times 1 + 3 \times 0,003 \times 3 + 0,001 \times 3 = 1,02 \text{ bits/símbolo}$$

$$RC = 3/1,02 = 2,94:1$$

$$H = 0,0998 \text{ bits/símbolo.}$$

Código de Huffman - Exemplo

$$R_{Huff} = 0,9202 \text{ bit/símbolo}$$

$$\bar{l}_{Huff} / H = 10,22$$

- $RC_{\max} = 3/0,0998 = 30,06:1$
- $RC_{\max} = 10,22 \times RC_{Huff}$
- O que fazer para aplicar o código de Huffman eficientemente nestes casos?
 - RLE, por exemplo.
- Huffman para alfabetos binários?

Modelos Probabilísticos

- Equiprobabilidade, $|A|$ (cardinalidade): entropia?
 - Equiprobabilidade, $|A| = 2^k$: entropia?
 - Equiprobabilidade \Leftrightarrow Entropia máxima
 - $H = \log_2 |A|$
 - Se $|A| = 2^k$, $H = k$.
- $$H = \sum_{i=0}^{M-1} P_i \log_2 \frac{1}{P_i}$$

Modelos Probabilísticos

- Equiprobabilidad, $|A| = 2^k$: entropia?

$$H = \sum_{i=0}^{M-1} P_i \log_2 \frac{1}{P_i}$$

- $P_i = 1/M$; $M = 2^k$
- $H = M (1/M \log_2(M))$
- $H = \log_2(M)$
- $H = \log_2(2^k)$
- $H = k$

Modelos Probabilísticos

- $P(a_i) = 1$: entropia?
- $P(a_i) = 1 \Leftrightarrow$ entropia mínima, $H = 0$.

Modelos Probabilísticos

- Estáticos: probabilidades estáticas durante todo o processo de codificação.
Probabilidade da Fonte.
- Semiadaptativos: probabilidades estimadas para cada mensagem a ser codificada, e mantidas estáticas durante a codificação.
- Adaptativos: probabilidades atualizadas durante a codificação.

Modelos Probabilísticos

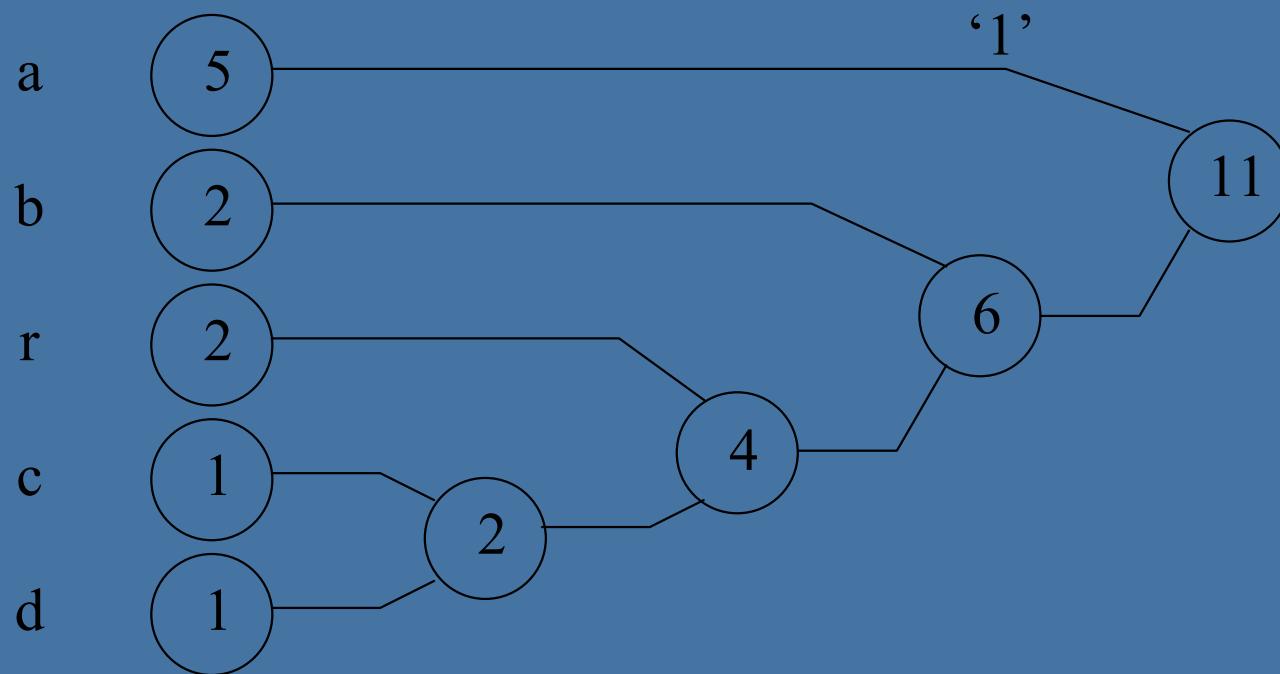
- Não-contextuais: símbolos independentes, probabilidades não condicionais.
- Contextuais: símbolos dependentes, probabilidades condicionais.

Huffman Adaptativo Decremento

- Huffman não contextual adaptativo de **decremento**
- W = abracadabra
- Inicia-se como no Huffman não-contextual semiadaptativo visto em aula.

Huffman Adaptativo Decremento

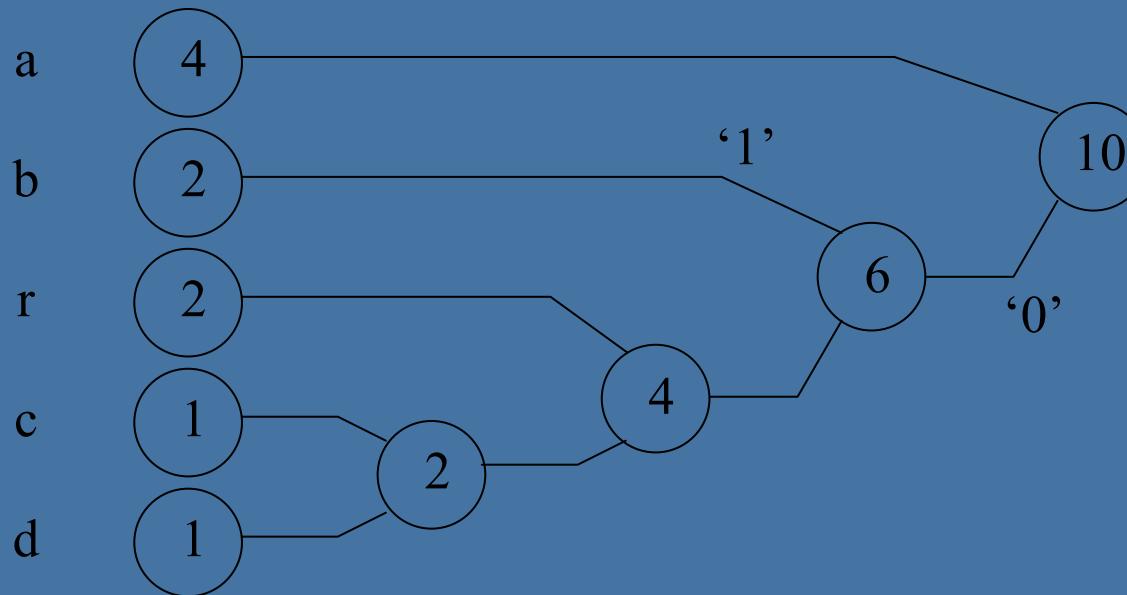
- Inicia-se como no Huffman não-contextual semiadaptativo
- ‘a’ codificado com palavra-código ‘1’



Huffman Adaptativo Decremento

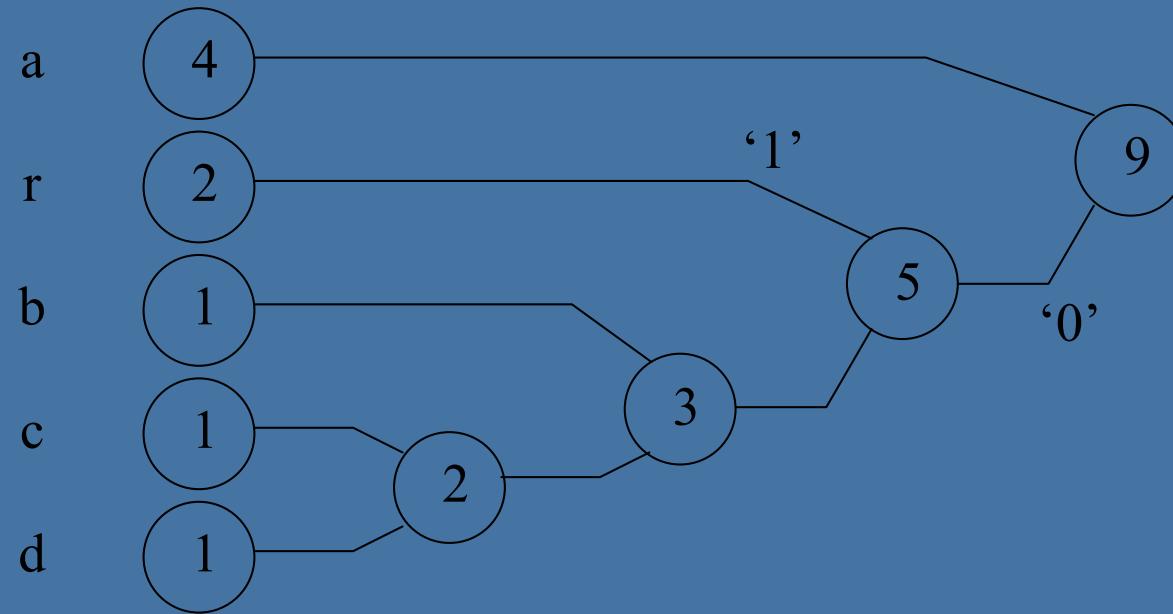
- $I_0(a) = \log_2 (11/5) = 1,14$ bit
- As estimativas de probabilidades são alteradas para refletir a ocorrência do símbolo ‘a’
- Atualização do modelo e reconstrução da árvore para codificação do próximo símbolo:

Huffman Adaptativo Decremento



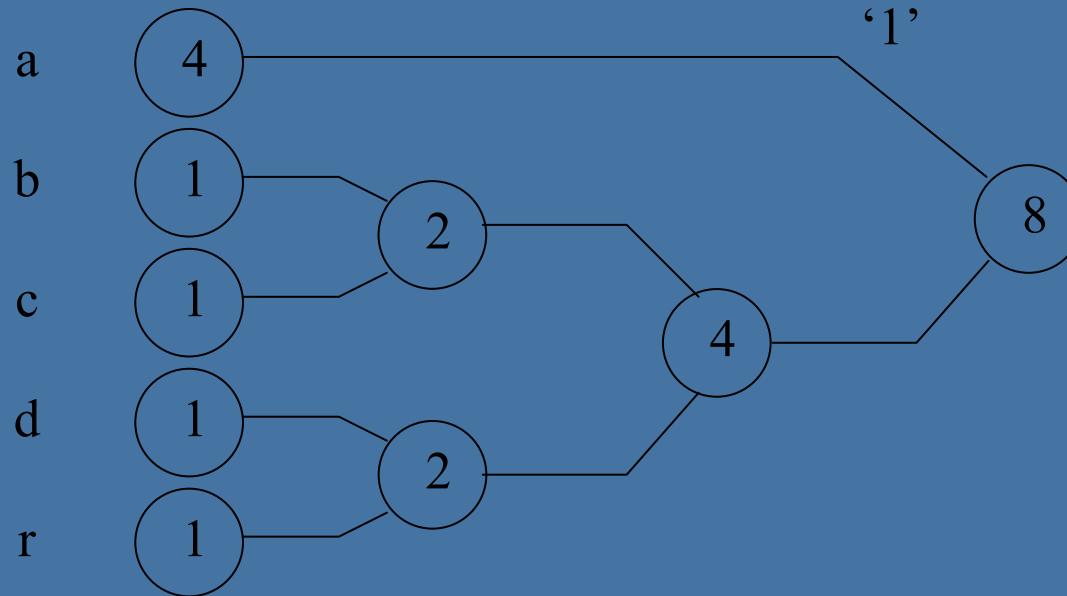
- 'b' codificado com palavra-código '01'
- $I_1(b) = \log_2 (10/2) = 2,32$ bits.
- Probabilidades alteradas pela ocorrência de 'b'.

Huffman Adaptativo Decremento



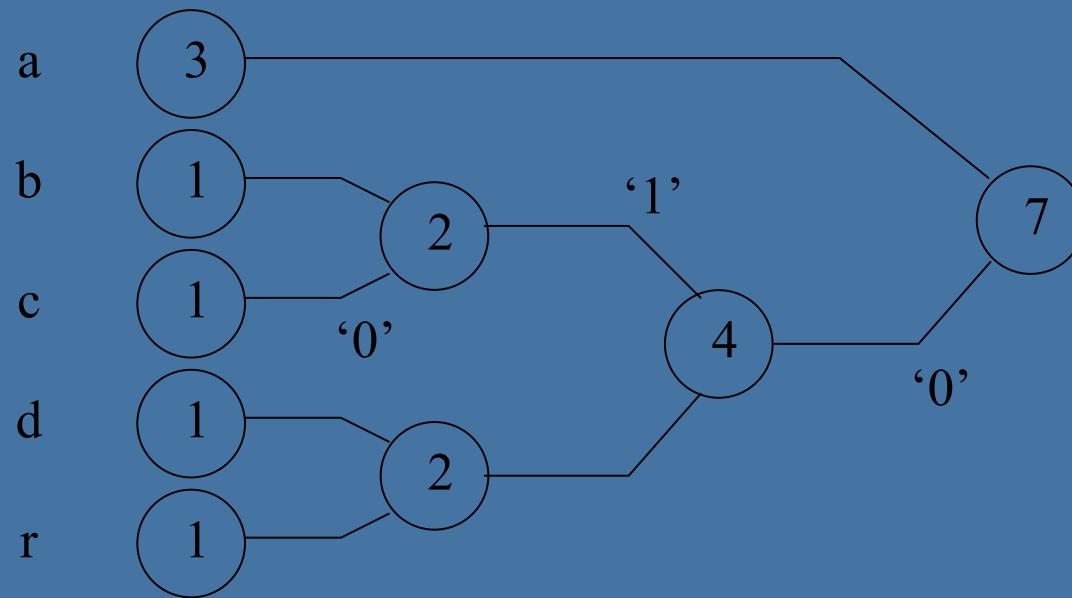
- 'r' codificado com '01' (= código de 'b' no passo anterior!!!)
- $I_2(r) = \log_2 (9/2) = 2,17$ bits.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



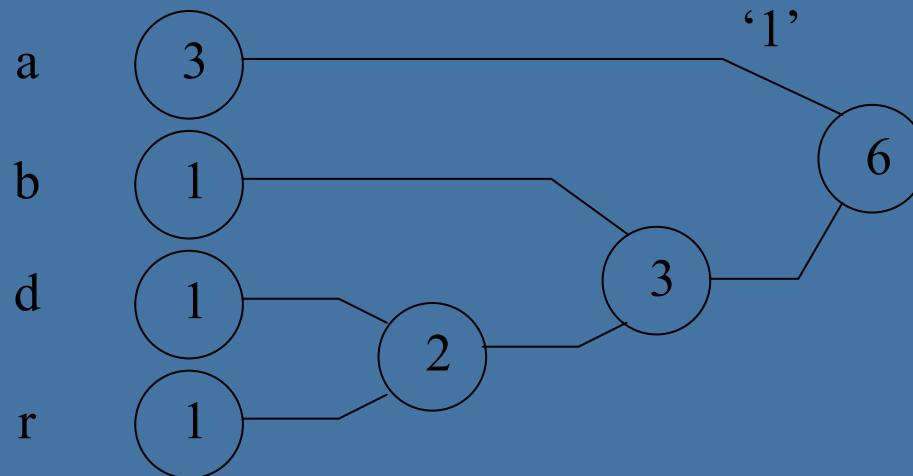
- 'a' codificado com '1'
- $I_3(a) = \log_2 (8/4) = 1$ bit.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



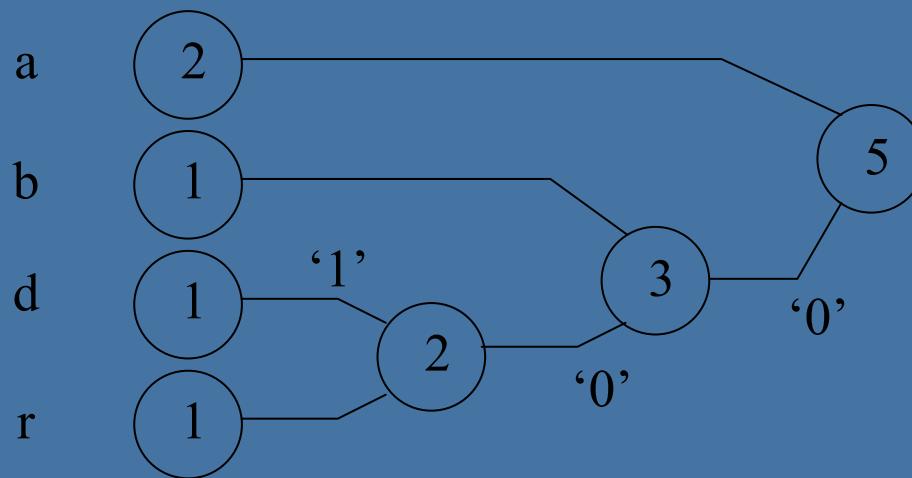
- 'c' codificado com '010'
- $I_4(c) = \log_2 (7/1) = 2,8$ bit.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



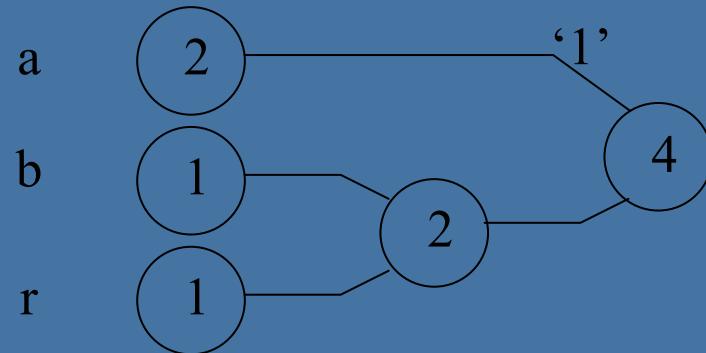
- 'a' codificado com '1'
- $I_5(a) = \log_2 (6/3) = 1$ bit.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



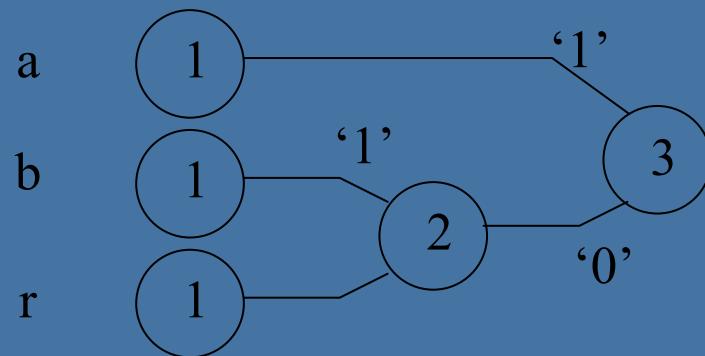
- ‘d’ codificado com ‘001’
- $I_6(d) = \log_2 (5/1) = 2,32$ bits.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



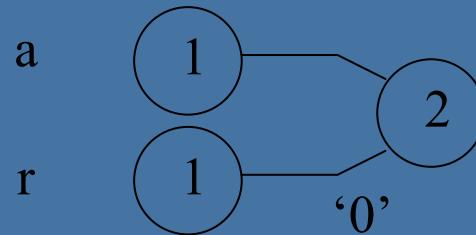
- 'a' codificado com '1'
- $I_7(a) = \log_2 (4/2) = 1$ bit.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



- 'b' codificado com '01'
- $I_8(b) = \log_2 (3/1) = 1,59$ bit.
- Probabilidades e árvore alteradas

Huffman Adaptativo Decremento



- 'r' codificado com '0'
- $I_9(r) = \log_2 (2/1) = 1$ bit.
- Probabilidades e árvore alteradas.

Huffman Adaptativo Decremento

a 1

- ‘a’ codificado com 0 bit
- $I_{10}(a) = \log_2 (1/1) = 0$ bit
- Comprimento médio do código: $17/11 = 1,55$
- Entropia?

Huffman Adaptativo Decremento

- Codificação: W = ABRACADABRA

simb.	cod.	prob.	Inf.
-------	------	-------	------

A	1	5/11	1,14
B	01	2/10	2,32
R	01	2/9	2,17
A	1	4/8	1
C	010	1/7	2,8
A	1	3/6	1
D	001	1/5	2,32
A	1	2/4	1
B	01	1/3	1,59
R	0	1/2	1
A	-	1	0

$W_C = 10101101010011010$

Huffman Adaptativo Decremento

○ Decodificação: $w_c = 10101101010011010$

Cabeçalho: $A = \{A:5, B:2, C:1, D:1, R:2\}$

Cod. Simb.

1	A
01	B
01	R
1	A
010	C
1	A
001	D
1	A
01	B
0	R
	A

Huffman Adaptativo Decremento

- Entropia é a média das informações.

$$H = \frac{1}{N} \sum_{i=0}^{N-1} \log_2 \frac{1}{P_i}$$

- Entropia de ordem 1 (não contextual)
- Modelo adaptativo, semi-adaptativo ou estático.
- N é o número de símbolos da mensagem.

Huffman Adaptativo Decremento

- Entropia é a media das informações.
- Entropia = 1,49

Entropia e Comprimento Médio:

- Comprimento médio para M=abracadabra:
 - Semi-adaptativo:
 - Huffman: 2,09
 - Entropia: 2,04
 - Redundância de Huffman: 0,05
 - Adaptativo de Decremento:
 - Huffman: 1,55
 - Entropia: 1,49
 - Redundância de Huffman: 0,06

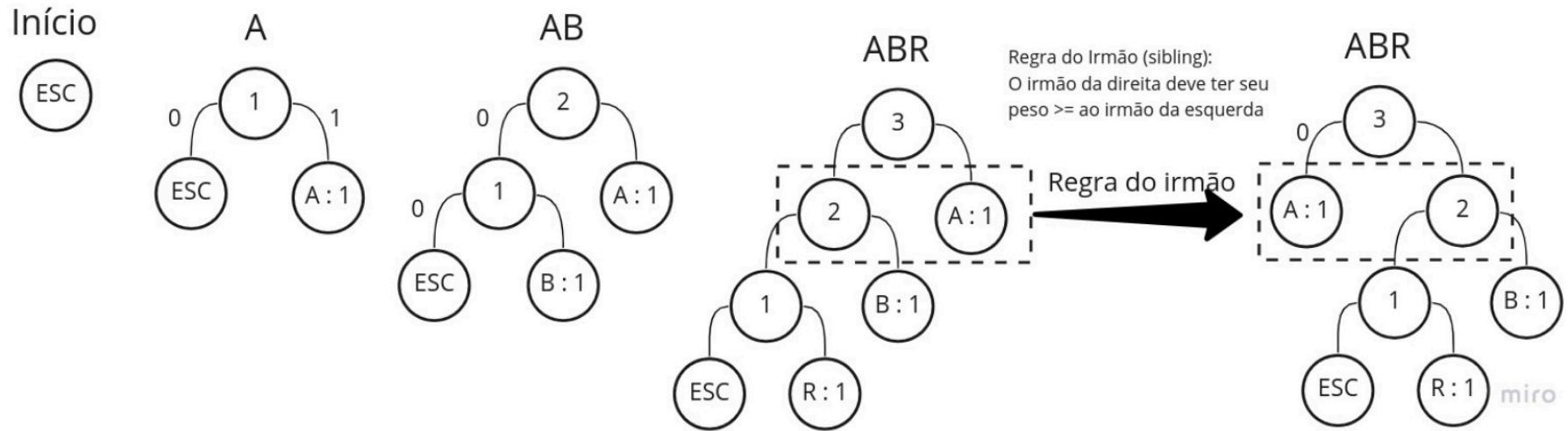
Huffman Adaptativo Incremento

- Exemplo: Huffman adaptativo de **incremento** para abracadabra.
- Regra do Irmão (sibling):
 - O irmão da direita deve ter seu peso \geq ao irmão da esquerda
- Regra do ancestral:
 - todos os nós de nível superior tem que ter peso \geq que os nós de nível inferior.

Huffman Adaptativo Incremento

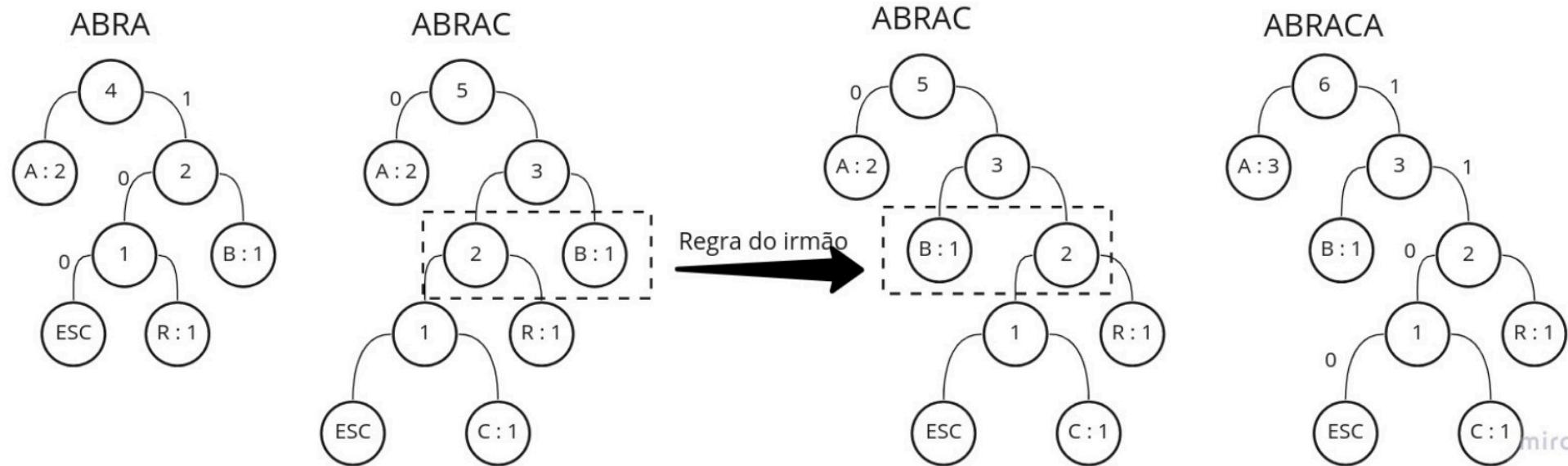
- Árvore começa vazia:

M = ABRACADABRA A = {A, B, C, D, R}



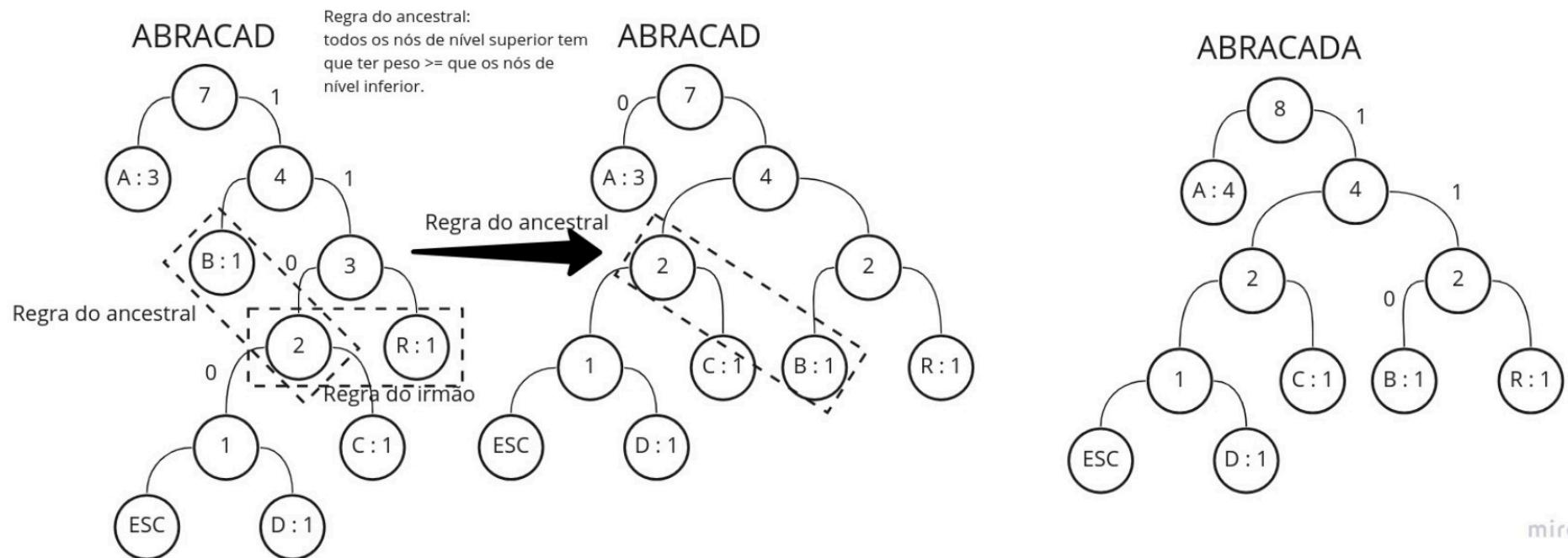
Huffman Adaptativo Incremento

Parte 2:



Huffman Adaptativo Incremento

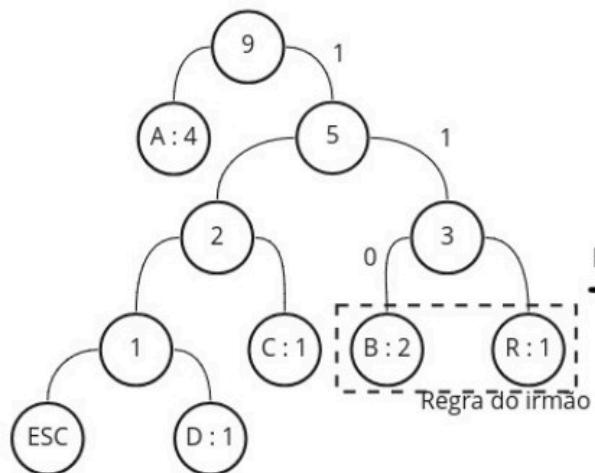
Parte 3:



Huffman Adaptativo Incremento

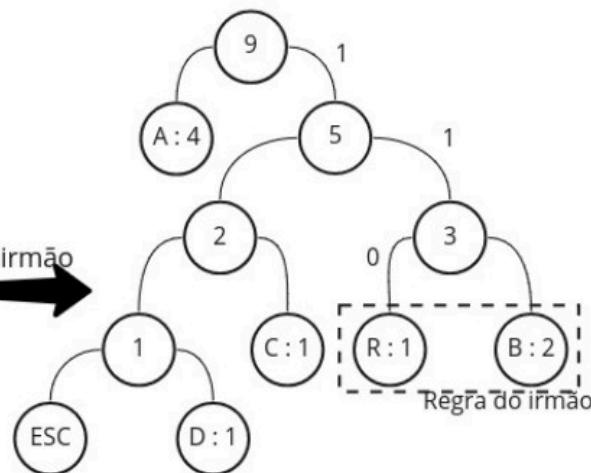
Parte 4:

ABRACADAB

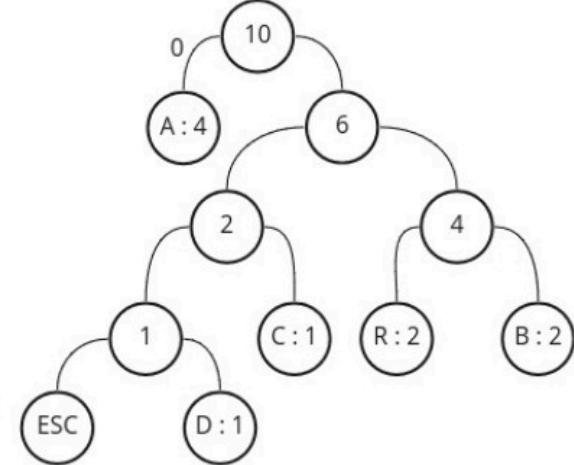


Regra do irmão

ABRACADAB



ABRACADABR



Huffman Adaptativo Incremento

- W = ABRACADABRA
- A = {A, B, C, D, R}
- Cod. Padrão
- A 000 B 00 C 00 C 0 D -
- B 001 C 01 D 01 D 1
- C 010 D 10 R 10
- D 011 R 11
- R 100

Huffman Adaptativo Incremento

- Codificação:

simb.	cod.	prob.	Inf.
ESC	-	1	0
A	000	1/5	2,32
ESC	0	1/2	1
B	00	1/4	2
ESC	00	1/3	1,59
R	10	1/3	1,59
A	0	1/3	1,59
ESC	100	1/5	2,321
C	0	1/2	1
A	0	2/5	1,321
ESC	1100	1/7	2,807
D	-	1	0
A	0	3/7	1,222
B	110	1/8	3
R	110	1/9	3,169
A	0	4/10	1,321

WC = 0000000010010000110001101100

Huffman Adaptativo Incremento

Comprimento médio: 2,55

Entropia: 2,39

Huffman Adaptativo Incremento

○ Decodificação:

WC = 000 0 00 00 10 0 10000110001101100

A = {A, B, C, D, R}

Simb. bits

ESC -

A 000

ESC 0

B 00

ESC 00

R 10

A 0

Huffman Adaptativo

- Diferenças Decremento e Incremento:
 - Decremento precisa da lista de frequências/probabilidades para decodificação.
 - Incremento usa mais bytes devido ao símbolo ESC e da tabela de codificação padrão.

Entropia e Comprimento Médio:

- Comprimento médio para $M=\text{abracadabra}:$
 - Semi-adaptativo:
 - Huffman: 2,09
 - Entropia: 2,04
 - Adaptativo de Decremento:
 - Huffman: 1,55
 - Entropia: 1,49
 - Adaptativo de Incremento:
 - Huffman: 2,55
 - Entropia: 2,39

Modelos Contextuais

- Se uma fonte gera símbolos que dependem dos valores em um contexto de tamanho L , um modelo de Markov de ordem k , com $k \leq L$, deve ser superior a um modelo de ordem $k-1$ e, consequentemente, conduz a uma redução na entropia.
- Empregar modelos de ordem elevada para aumentar a compressão
- Problemas práticos limitam a ordem

Modelos Contextuais

- Número de contextos diferentes para modelo de ordem k e alfabeto com cardinalidade $|A|$?
 - $|A|^k$ contextos diferentes
- Número de contextos diferentes cresce exponencialmente com k
- Quantos contadores devem ser passados ao decodificador utilizando um modelo semi-adaptativo?
 - $|A|^{k+1}$

Modelos Contextuais

- Se $|A| = 256$ e $k = 4$: aproximadamente um trilhão de contextos diversos ($256^5 = 1$ TB).
- Problemas práticos:
 - memória, tempo de processamento, estimativas de probabilidades, transmissão para o decodificador das probabilidades condicionais.
- Usar modelos adaptativos de incremento elimina apenas o último dos problemas citados!
- Na prática, para $M = 256$, $k = 0, 1, \dots, 10$

Modelos Contextuais

- Fonte S com $A = \{a_0, a_1, \dots, a_{M-1}\}$
- Elemento x gerado por S é uma variável aleatória
- Modelos não-contextuais: $P(x = a_i)$
- Modelos contextuais: $P(x_n = a_i | x_{n-1}, x_{n-2}, \dots)$
- Modelo de Markov de k -ésima ordem:
 $P(x_n = a_i | x_{n-1}, x_{n-2}, \dots, x_{n-k})$
- Modelos de memória finita
- $k=0$, sem contexto; $k \geq 1$ contextual.

Modelos Contextuais

- ◎ $\mathbf{x}^k = x_{n-1}, x_{n-2}, \dots, x_{n-k}$
- ◎ Define-se informação condicional associada a a_i , dado um contexto \mathbf{x}^k como:

$$I(a_i | \mathbf{x}^k) = \log_2 \frac{1}{P(x_n = a_i | \mathbf{x}^k)}$$

Exemplo:

- $A = \{0,1\}$, Mensagem:

1010101010101010101010101010101010101
0101010101010101010101010101

- Entropia ?

- Modelo não-contextual

- $H = 1 \text{ bit/símbolo}$ (1 bit/bit)

- Modelo contextual de ordem 1

- H aproximadamente 0 bit/símbolo

Modelos De Markov de Ordem 1

- Fonte com alfabeto $A = \{a_0, a_1, \dots, a_{M-1}\}$.
- Bigrama: par ordenado de símbolos consecutivos gerados pela fonte.
- O bigrama (a_i, a_j) é diferente do bigrama (a_j, a_i) , para i diferente de j .
- Experimento probabilístico: observar um bigrama da mensagem.
- Eventos de interesse:
 - A_i : a_i é o 1º elemento do bigrama
 - B_i : a_i é o 2º elemento do bigrama

Modelos De Markov de Ordem 1

- Probabilidade de ocorrência simultânea dos eventos A_i e B_j :

$$P(A_i, B_j) = P(B_j, A_i)$$

- Probabilidade de ocorrência de B_j dada a ocorrência de A_i :

$$P(B_j \mid A_i) \neq P(A_i \mid B_j)$$

Modelos De Markov de Ordem 1

- Regra da probabilidade condicional:

$$P(B_j \mid A_i) = \frac{P(A_i, B_j)}{P(A_i)}$$

Modelos De Markov de Ordem 1

- Teorema de Bayes:

$$P(B_i | A_j) = \frac{P(A_j | B_i)P(B_i)}{P(A_j)}$$

Modelos De Markov de Ordem 1

- Exemplo
- Mensagem W =
abracadabraabraabarcabrab.
- $A = \{a, b, c, d, r\}$.
- Evento B_2 : símbolo c na 2^a posição de um bigrama
- Evento A_0 : símbolo a na 1^a posição de um bigrama
- Calcular: $P(A_i)$, $P(B_j)$, $P(A_i, B_j)$, $P(B_j | A_i)$, $P(A_i | B_j)$,
 $P(B_j | A_i)$

Modelos De Markov de Ordem 1

- W: 26 símbolos, 25 bigramas (ab, br, ra, ...ba).
- A_0 ocorre 11 vezes na mensagem
- B_2 ocorre 2 vezes.
- Estimando-se as probabilidades pela freqüência relativa, tem-se:

Modelos De Markov de Ordem 1

- A_0 ocorreu 11 vezes em 25 bigramas:
 $P(A_0) = 11/25$
- B_2 ocorreu 2 vezes em 25 bigramas: $P(B_2) = 2/25$
- A_0 ocorreu 11 vezes; destas, em uma B_2 ocorreu: $P(B_2 | A_0) = 1/11$
- B_2 ocorreu 2 vezes; destas, em uma A_0 ocorreu: $P(A_0 | B_2) = 1/2$
- A_0 e B_2 ocorreram 1 vez nos 25 bigramas observados: $P(A_0, B_2) = P(B_2, A_0) = 1/25$

Modelos De Markov de Ordem 1

- ◎ Observe que:

$$P(B_2 \mid A_0) = \frac{P(A_0, B_2)}{P(A_0)} = \frac{1/25}{11/25} = \frac{1}{11}$$

$$P(A_0 \mid B_2) = \frac{P(A_0, B_2)}{P(B_2)} = \frac{1/25}{2/25} = \frac{1}{2}$$

Modelos De Markov de Ordem 1

- Distribuição de probabilidades conjuntas:

1º símbolo do bigrama	2º símbolo do bigrama				
	a	b	c	d	r
a	2/25	6/25	1/25	1/25	1/25
b	2/25	0	0	0	4/25
c	2/25	0	0	0	0
d	1/25	0	0	0	0
r	4/25	0	1/25	0	0

- Observe que o somatório de todas as células da tabela resulta em 1

Modelos De Markov de Ordem 1

- Distribuição de probabilidades condicionais $P(B_j | A_i)$

1º símbolo do bigrama	2º símbolo do bigrama				
	a	b	c	d	r
a	2/11	6/11	1/11	1/11	1/11
b	2/6	0	0	0	4/6
c	2/2	0	0	0	0
d	1/1	0	0	0	0
r	4/5	0	1/5	0	0

- O somatório de todas as células de cada linha da tabela resulta em 1

Exemplo: modelo de ordem 1

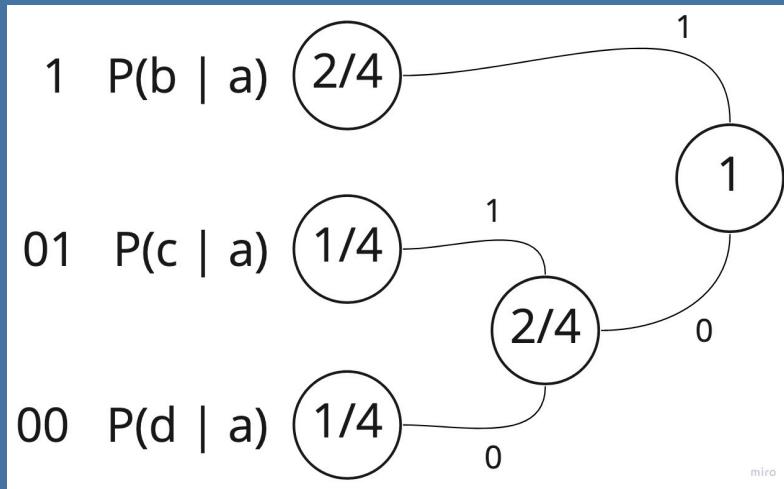
- Huffman semi-adaptativo contextual de ordem 1
- Mensagem: abracadabra

Exemplo: modelo de ordem 1

- O símbolo a ocorre 5 vezes na mensagem, mas apenas em 4 é contexto de outro.
- Quatro ocorrências do contexto a:
 - em 2 o símbolo subsequente é b
 - em 1 o símbolo subsequente é c
 - em 1 o símbolo subsequente é d
- Portanto,
 - $P(b | a) = 2/4 \rightarrow I(b | a) = 1$
 - $P(c | a) = 1/4 \rightarrow I(c | a) = 2$
 - $P(d | a) = 1/4 \rightarrow I(d | a) = 2$

Exemplo: modelo de ordem 1

- ◎ Gera uma árvore para cada contexto.
- Contexto a:



Exemplo: modelo de ordem 1

- Árvore de Huffman apenas com a raiz
- Por um raciocínio análogo:
 - $P(r | b) = 1 \rightarrow I(r | b) = 0$
 - $P(a | c) = 1 \rightarrow I(a | c) = 0$
 - $P(a | d) = 1 \rightarrow I(a | d) = 0$
 - $P(a | r) = 1 \rightarrow I(a | r) = 0$

Exemplo: modelo de ordem 1

- Saída da codificação Huffman contextual:

simb. cod. prob inf.

A	1	5/11	1,13	(não contextual)
B	1	2/4	1	$P(B A)$
R	-	1	0	$P(R B)$
A	-	1	0	$P(A R)$
C	01	1/4	2	$P(C A)$
A	-	1	0	$P(A C)$
D	00	1/4	2	$P(D A)$
A	-	1	0	$P(A D)$
B	1	2/4	1	$P(B A)$
R	-	1	0	$P(R B)$
A	-	1	0	$P(A R)$

Exemplo: modelo de ordem 1

- Primeiro símbolo: não recodificar, ou codificar usando um modelo não contextual.
- $Lm1 = 6/10 = 0,6$ (sem o primeiro símbolo)
- $Lm2 = 7/11 = 0,63$ (com todos os símbolos)
- $RC1 = 3/0,6 = [5:1]$
- $RC2 = 3/0,63 = [4,76:1]$
- $H1 = 6/10 = 0,6$
- $H2 = 7,13/11 = 0,64$

Exemplo: modelo de ordem 1

- Modelos contextuais tendem a produzir muitos valores de informação condicional bem inferiores a 1, mas não nulos, e o Huffman se torna ineficiente

Entropia e Comprimento Médio:

- Comprimento médio para M=abracadabra:
 - Semi-adaptativo:
 - Huffman: 2,09
 - Entropia: 2,04
 - Adaptativo de Decremento:
 - Huffman: 1,55
 - Entropia: 1,49
 - Adaptativo de Incremento:
 - Huffman: 2,55
 - Entropia: 2,39
 - Contextual semi-adaptativo:
 - Huffman1: 0,6 Huffman2: 0,63
 - Entropia1: 0,6 Entropia2: 0,64

Exemplo 2: modelo de ordem 1

- Exemplo: Huffman semi-adaptativo contextual de ordem 1
- Mensagem: abracadabraabraabarcabrabab

Entropia de Ordem 2

- Markov de ordem 1: entropia de ordem 2
- $I(B_j|A_i) = \log_2[1/P(B_j|A_i)]$: informação associada a a_j dado o contexto a_i
- Mensagem W = abracadabraabraabarcabrab, A = {a, b, c, d, r}
- $H_2 = [I(b|a) + I(r|b) + I(a|r) ... + I(a|b)]/25$
- $H_2 = [2I(a|a) + 6I(b|a) + ... + 1I(c|r)]/25$

Entropia de Ordem 2

- Markov de ordem 1: entropia de ordem 2
- $I(B_j | A_i) = \log_2 [1/P(B_j | A_i)]$: informação associada a a_j dado o contexto a_i
- N_{ij} : nº de ocorrências de a_j no contexto a_i = nº de ocorrências de $a_i a_j$
- N_b : total de bigramas em uma mensagem

$$H_2 = -\frac{1}{N_b} \times [N_{00} \log_2 P(B_0 | A_0) + N_{01} \log_2 P(B_1 | A_0) + N_{02} \log_2 P(B_2 | A_0) + \dots + N_{M-1,M-1} \log_2 P(B_{M-1} | A_{M-1})]$$

Entropia de Ordem 2

- ◎ Estimando probabilidades por freqüência relativa:

$$H_2 = -[P(A_0, B_0) \log_2 P(B_0 | A_0) + P(A_0, B_1) \log_2 P(B_1 | A_0) + \dots + P(A_{M-1}, B_{M-1}) \log_2 P(B_{M-1} | A_{M-1})]$$

$$H_2 = - \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(A_i, B_j) \log_2 P(B_j | A_i)$$

Entropia de Ordem 2

○ M=abracadabra

$$\begin{array}{lll} P(b|a) = 2/4 \rightarrow I(b|a) = 1 & P(a,b) = 2/10 \\ P(c|a) = 1/4 \rightarrow I(c|a) = 2 & P(a,c) = 1/10 \\ P(d|a) = 1/4 \rightarrow I(d|a) = 2 & P(a,d) = 1/10 \\ P(r|b) = 1 \rightarrow I(r|b) = 0 & P(b,r) = 2/10 \\ P(a|c) = 1 \rightarrow I(a|c) = 0 & P(c,a) = 1/10 \\ P(a|d) = 1 \rightarrow I(a|d) = 0 & P(d,a) = 1/10 \\ P(a|r) = 1 \rightarrow I(a|r) = 0 & P(r,a) = 2/10 \end{array}$$

$$H_2 = - \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(A_i, B_j) \log_2 P(B_j | A_i)$$

$$H_2 = 0,6$$

Entropia de Ordem 2 – Média Simb.

- Entropia é a média das informações.

$$H = \frac{1}{N} \sum_{i=0}^{N-1} \log_2 \frac{1}{P(B_j | A_i)}$$

- Entropia de ordem 2
- Modelo adaptativo, semi-adaptativo ou estático.
- N é o número de bigramas da mensagem.

Entropia de Ordem N

- $K = N-1$
- Contexto v_i : k símbolos consecutivos
- Alfabeto $A = \{a_0, a_1, \dots, a_{M-1}\}$: $C = M^k$ contextos distintos
 - $v_0 = a_0 \ a_0 \ a_0 \dots a_0$
 - $v_1 = a_0 \ a_0 \ a_0 \dots a_1$
 - ...
 - $v_{C-1} = a_{M-1} \ a_{M-1} \ a_{M-1} \dots a_{M-1}$

Entropia de Ordem N

- Eventos A_i e B_i
- A_i : o contexto v_i ocorre no início de um N-grama
- B_i : o símbolo a_i ocorre na última posição de um N-grama
- Valem as mesmas equações de probabilidades condicionais anteriormente apresentadas.

Entropia de Ordem N

- ◎ Entropia de ordem N

$$H_N = - \sum_{i=0}^{C-1} \sum_{j=0}^{M-1} P(A_i, B_j) \log_2 P(B_j | A_i)$$

- ◎ Para Modelos semi-adaptativos ou estáticos

Exemplo: modelo contextual adaptativo

- Exemplo:
- Huffman adaptativo contextual de ordem 1.
- PPM
- Mensagem = abracadabra

Modelos Probabilísticos

- Família LZ (contextuais adaptativos)
- Família PPM (contextuais adaptativos)
- Equivalência entre compressores baseados em dicionários e compressores estatísticos

PPM

- Prediction by Partial Matching
- Cleary, J.G. and Witten, I.H. (1984)
“Data compression using adaptive coding and partial string matching,” IEEE Transactions on Communications, 32(4), 396-402

Exemplo: PPM

- $W = \text{abracadabra}$
- Contexto máximo $K = 2$
- $A = \{a, b, c, d, r\}$

Exemplo: PPM

- ◎ Modelo inicial (vazio) :

K = 2				K = 1				K = 0				K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.	1/ A = 1/5

Exemplo: PPM

- ◎ Modelo inicial (vazio) :



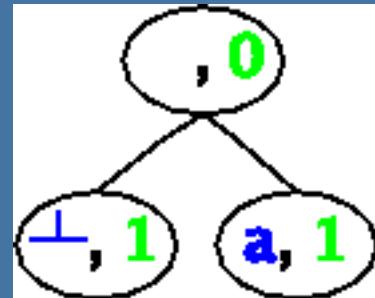
Exemplo: PPM

- abracadabra: ‘a’ codificado em K = -1 com $P('a') = 1/5$
 - Modelo atualizado:

K = 2				K = 1				K = 0				K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.	
								a	1	1/2	1/4	
								esc	1	1/2		

Exemplo: PPM

- abracadabra: ‘a’ codificado em $K = -1$ com $P('a') = 1/5$
- Modelo atualizado:



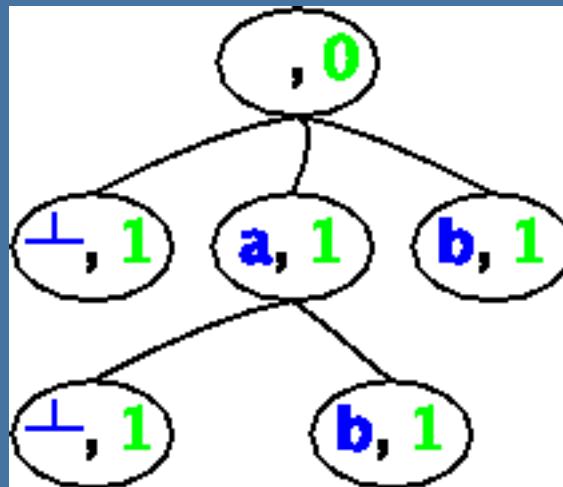
Exemplo: PPM

- **abracadabra**: ‘esc’ em $K = 0$ codificado com prob. $\frac{1}{2}$; e ‘b’ em $K = -1$ com prob. $\frac{1}{4}$
- Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
				a	b	1	1/2	a	1	1/4	1/3
					esc	1	1/2	b	1	1/4	
								esc	2	2/4	

Exemplo: PPM

- **abracadabra**: ‘esc’ em $K = 0$ codificado com prob. $\frac{1}{2}$; e ‘b’ em $K = -1$ com prob. $\frac{1}{4}$
- Modelo atualizado:



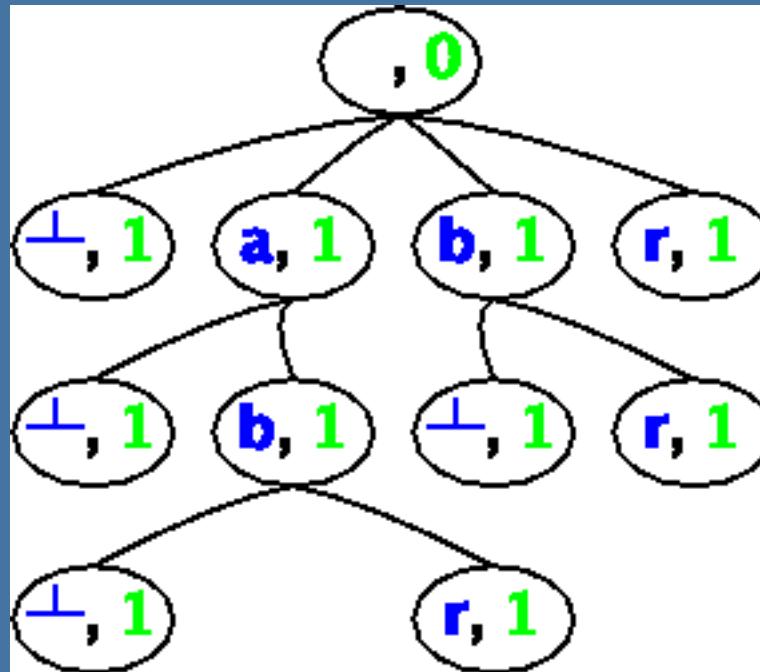
Exemplo: PPM

- abracadabra: ‘esc’ codificado em $K = 0$ com prob. $\frac{1}{2}$; ‘r’ em $K = -1$ com prob. $1/3$.
- Modelo atualizado:

K = 2				K = 1				K = 0				K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.	
ab	r	1	1/2	a	b	1	1/2	a	1	1/6	1/2	
	esc	1	1/2		esc	1	1/2	b	1	1/6		
				b	r	1	1/2	r	1	1/6		
					esc	1	1/2	esc	3	3/6		

Exemplo: PPM

- abracadabra: ‘esc’ codificado em $K = 0$ com prob. $\frac{1}{2}$; ‘r’ em $K = -1$ com prob. $1/3$.
- Modelo atualizado:



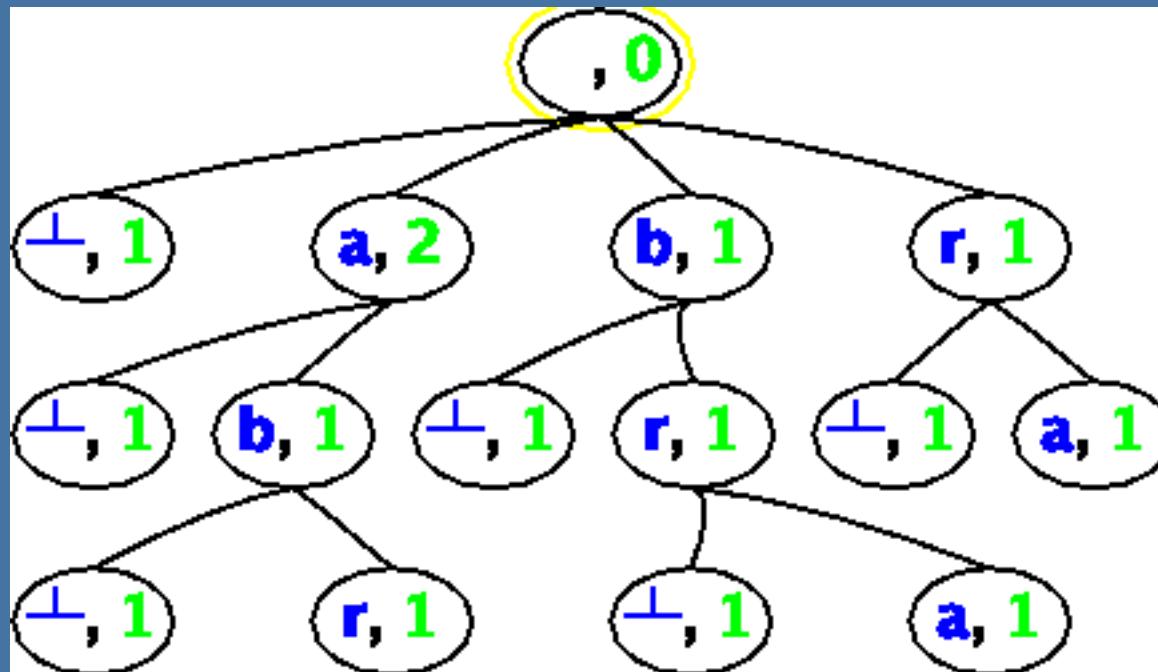
Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 1/6
- Modelo atualizado:

K = 2				K = 1				K = 0				K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.	
ab	r	1	1/2	a	b	1	1/2	a	2	2/7	1/2	
	esc	1	1/2		esc	1	1/2	b	1	1/7		
br	a	1	1/2	b	r	1	1/2	r	1	1/7	1/2	
	esc	1	1/2		esc	1	1/2	esc	3	3/7		
				r	a	1	1/2					
				esc		1	1/2					

Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 1/6
- Modelo atualizado:



Exemplo: PPM

- abracadabra: a->'esc' codificado com prob. 1/2. No chaveamento para $K = 0$, o símbolo a ser codificado não pode ser 'b', pois ele teria sido codificado no contexto 'a'. Assim, 'b' é excluído da coluna $K = 0$, e as probabilidades nesta coluna são recalculadas, apenas para a codificação do símbolo. 'esc' em $K = 0$ é codificado com prob. 3/6, e 'c' na coluna $K = -1$ com prob. 1/2).
- Modelo atualizado:

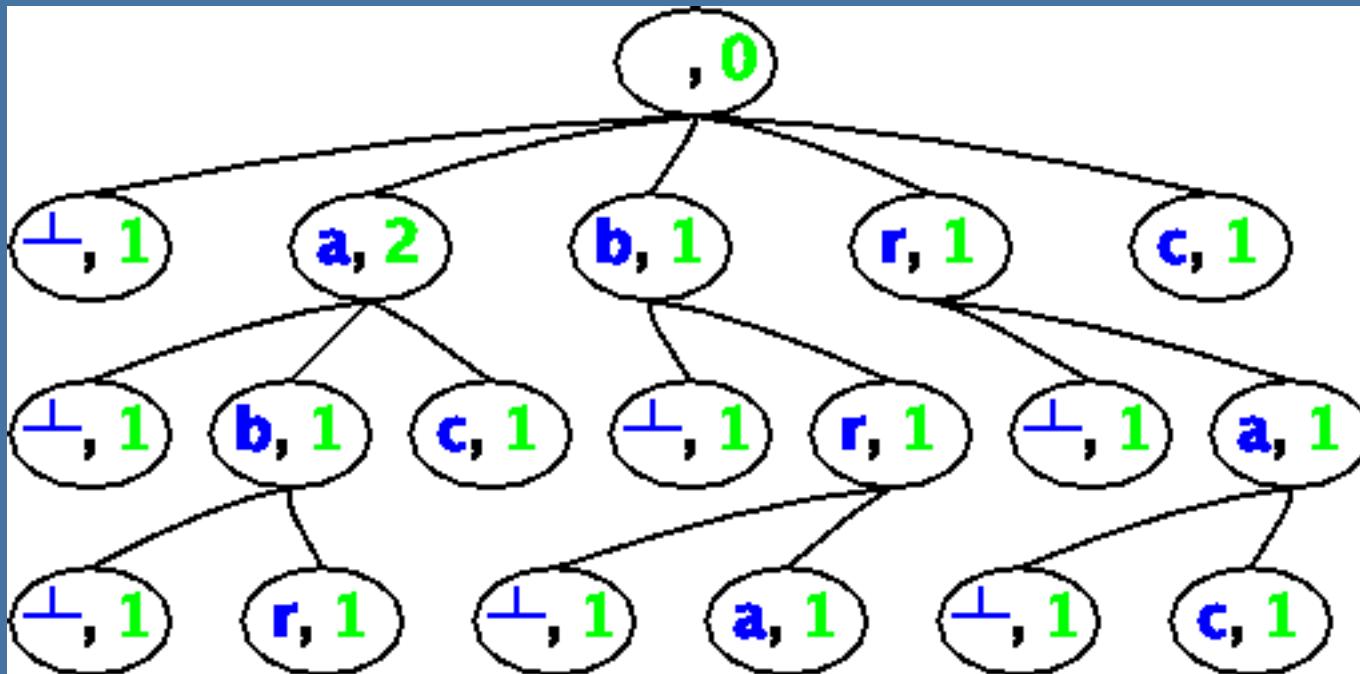
Exemplo: PPM

- abracadabra:
- Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/4	a	2	2/9	1
	esc	1	1/2		c	1	1/4	b	1	1/9	
br	a	1	1/2		esc	2	2/4				
	esc	1	1/2	b	r	1	1/2	c	1	1/9	
ra	c	1	1/2		esc	1	1/2	r	1	1/9	
	esc	1	1/2	r	a	1	1/2	esc	4	4/9	
					esc	1	1/2				

Exemplo: PPM

- abracadabra:
- Modelo atualizado:



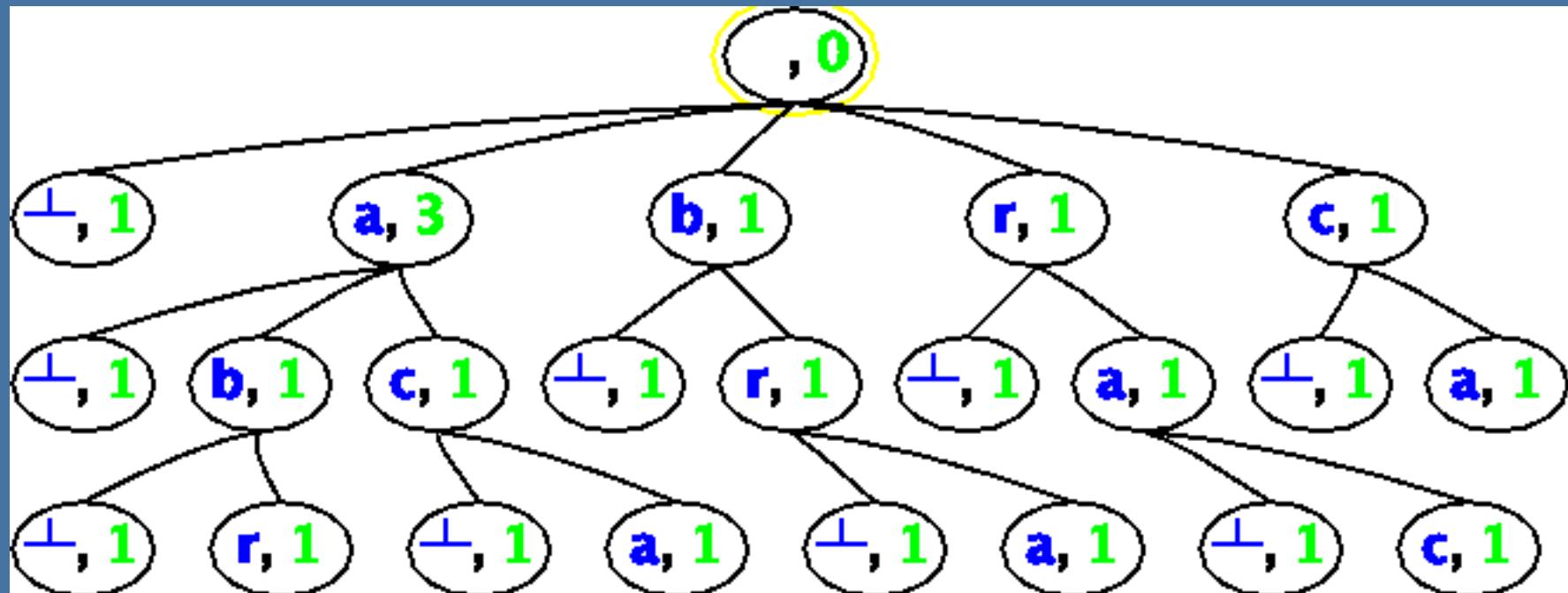
Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 2/9.
- Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/4	a	3	3/10	1
	esc	1	1/2		c	1	1/4	b	1	1/10	
ac	a	1	1/2		esc	2	2/4				
	esc	1	1/2	b	r	1	1/2	c	1	1/10	
br	a	1	1/2		esc	1	1/2	r	1	1/10	
	esc	1	1/2	c	a	1	1/2	esc	4	4/10	
ra	c	1	1/2		esc	1	1/2				
	esc	1	1/2	r	a	1	1/2				
					esc	1	1/2				

Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 2/9.
 - Modelo atualizado:



Exemplo: PPM

- abracadabra: a->‘esc’ em $K = 1$ codificado com prob. $2/4$; ‘esc’ em $K = 0$ com prob. $4/8$ (exclusão de ‘b’ e ‘c’) ; ‘d’ em $K = -1$ com prob. 1.
- Não pode mais ocorrer ‘esc’ na coluna $K = 0$, e a coluna $K = -1$ não será mais utilizada.
- Modelo atualizado:

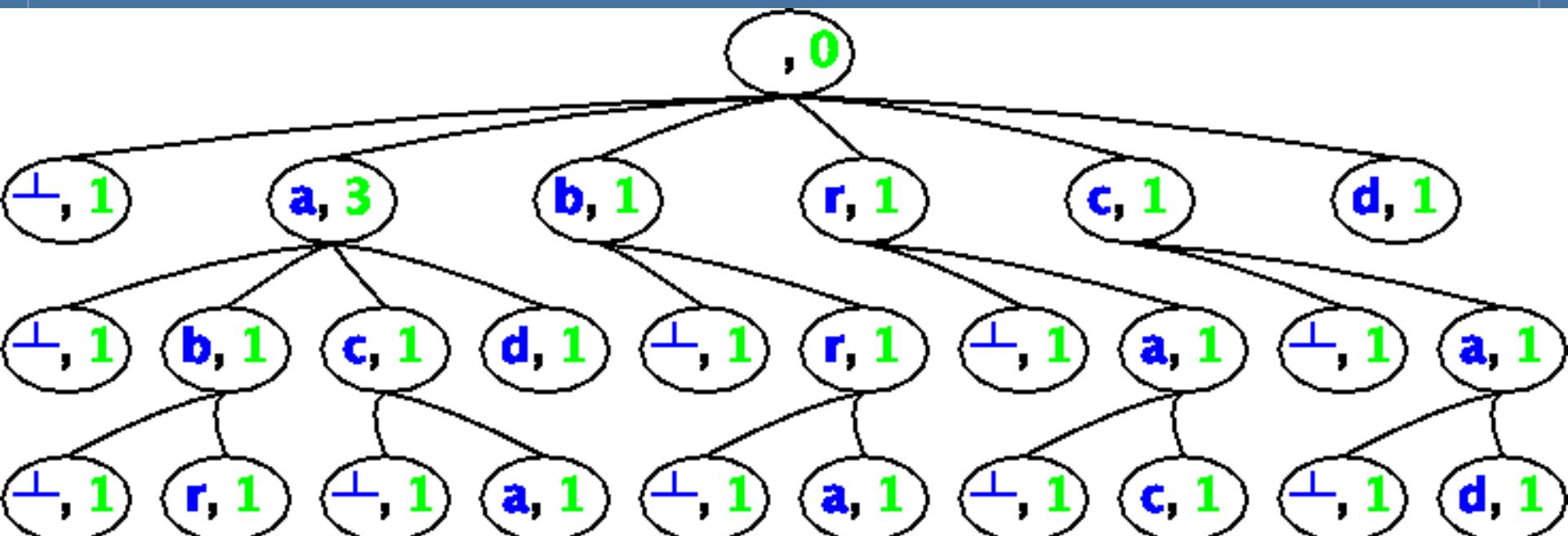
Exemplo: PPM

- abracadabra:
- Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	1	1/6	a	3	3/7	-
	esc	1	1/2		c	1	1/6	b	1	1/7	
ac	a	1	1/2		d	1	1/6				
	esc	1	1/2		esc	3	3/6	c	1	1/7	
br	a	1	1/2	b	r	1	1/2	d	1	1/7	
	esc	1	1/2		esc	1	1/2	r	1	1/7	
ca	d	1	1/2	c	a	1	1/2				
	esc	1	1/2		esc	1	1/2				
ra	c	1	1/2	r	a	1	1/2				
	esc	1	1/2		esc	1	1/2				

Exemplo: PPM

- abracadabra:
- Modelo atualizado:



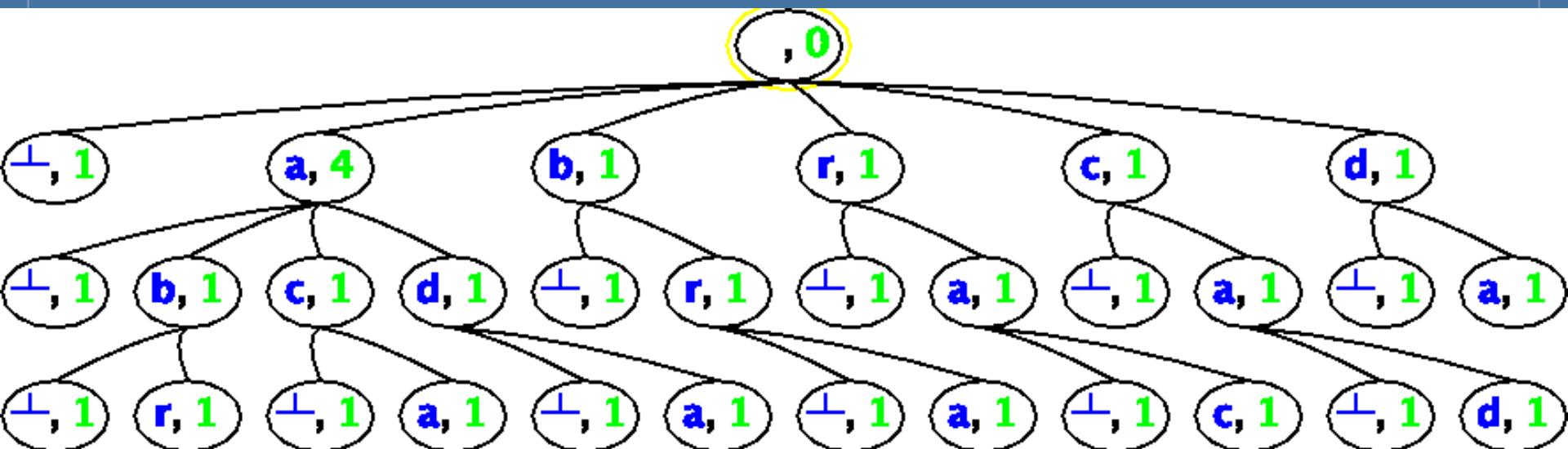
Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 3/7. Modelo atualizado:

K = 2				K = 1				K = 0				K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.	
ab	r	1	1/2	a	b	1	1/6	a	4	4/8	-	
	esc	1	1/2		c	1	1/6	b	1	1/8		
ac	a	1	1/2		d	1	1/6					
	esc	1	1/2		esc	3	3/6	c	1	1/8		
ad	a	1	1/2	b	r	1	1/2	d	1	1/8		
	esc	1	1/2		esc	1	1/2	r	1	1/8		
br	a	1	1/2	c	a	1	1/2					
	esc	1	1/2		esc	1	1/2					
ca	d	1	1/2	d	a	1	1/2					
	esc	1	1/2		esc	1	1/2					
ra	c	1	1/2	r	a	1	1/2					
	esc	1	1/2		esc	1	1/2					

Exemplo: PPM

- abracadabra: ‘a’ em K = 0 codificado com prob. 3/7. Modelo atualizado:



Exemplo: PPM

- abracadabra: ‘b’ no contexto ‘a’ em $K = 1$ codificado com prob. $1/6$.
- Modelo atualizado:

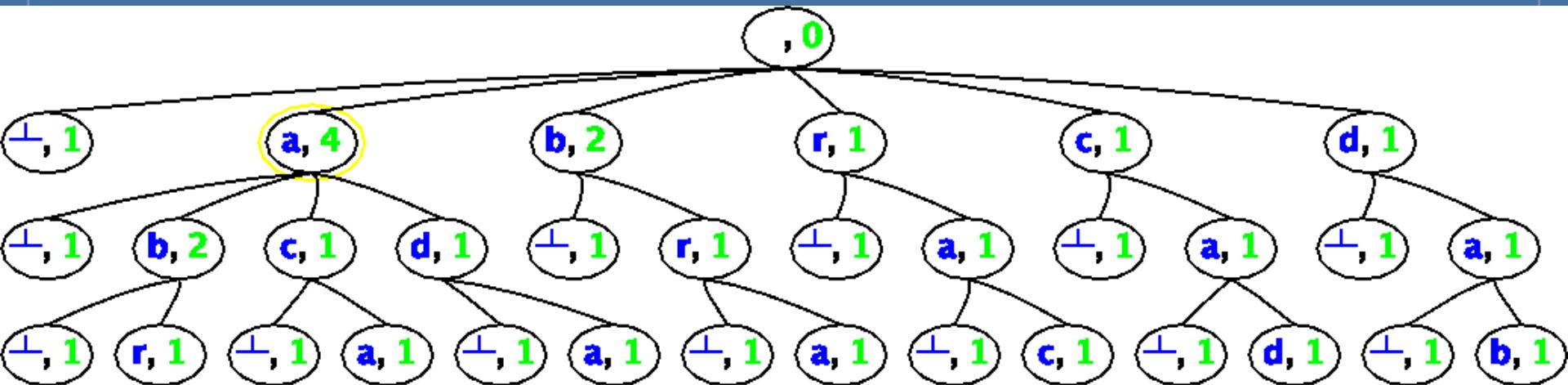
Exemplo: PPM

○ abracadabra: Modelo atualizado:

K = 2				K = 1				K = 0			K = -1 (Equip.)
Contexto	Simb.	Cont.	Prob.	Contexto	Simb.	Cont.	Prob.	Simb.	Cont.	Prob.	Prob.
ab	r	1	1/2	a	b	2	2/7	a	4	4/9	-
	esc	1	1/2		c	1	1/7	b	2	2/9	
					d	1	1/7				
					esc	3	3/7	c	1	1/9	
ac	a	1	1/2	b	r	1	1/2	d	1	1/9	
	esc	1	1/2		esc	1	1/2	r	1	1/9	
ad	a	1	1/2	c	a	1	1/2				
	esc	1	1/2		esc	1	1/2				
br	a	1	1/2	d	a	1	1/2				
	esc	1	1/2		esc	1	1/2				
ca	d	1	1/2	r	a	1	1/2				
	esc	1	1/2		esc	1	1/2				
da	b	1	1/2								
	esc	1	1/2								
ra	c	1	1/2								
	esc	1	1/2								

Exemplo: PPM

- abracadabra: Modelo atualizado:



Exemplo: PPM

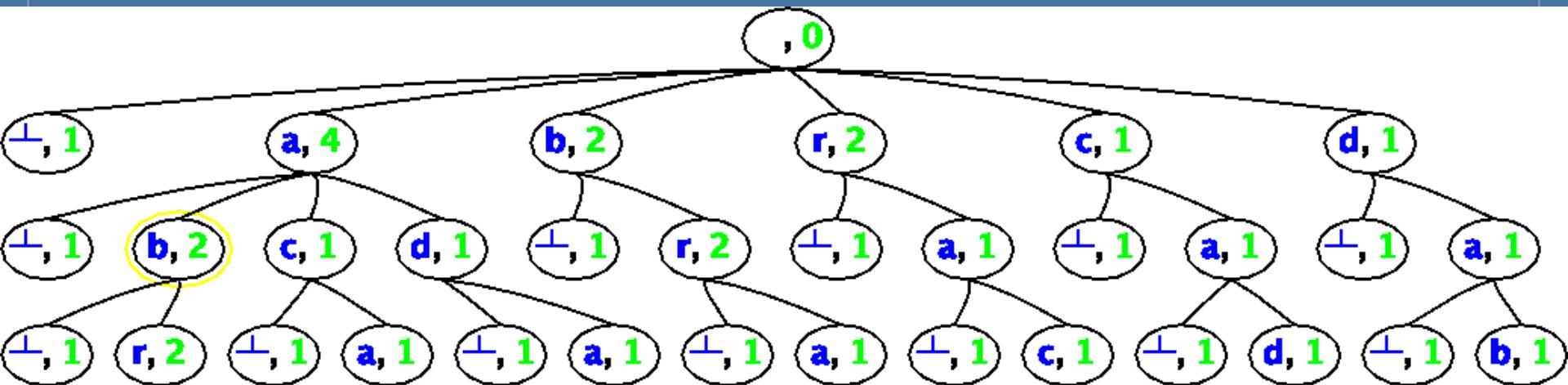
- abracadabra: ‘r’ no contexto ‘ab’ em $K = 2$, codificado com prob. $\frac{1}{2}$.
- Modelo atualizado:

Exemplo: PPM

● abracadabra: Modelo atualizado:

Exemplo: PPM

- abracadabra: Modelo atualizado:



Exemplo: PPM

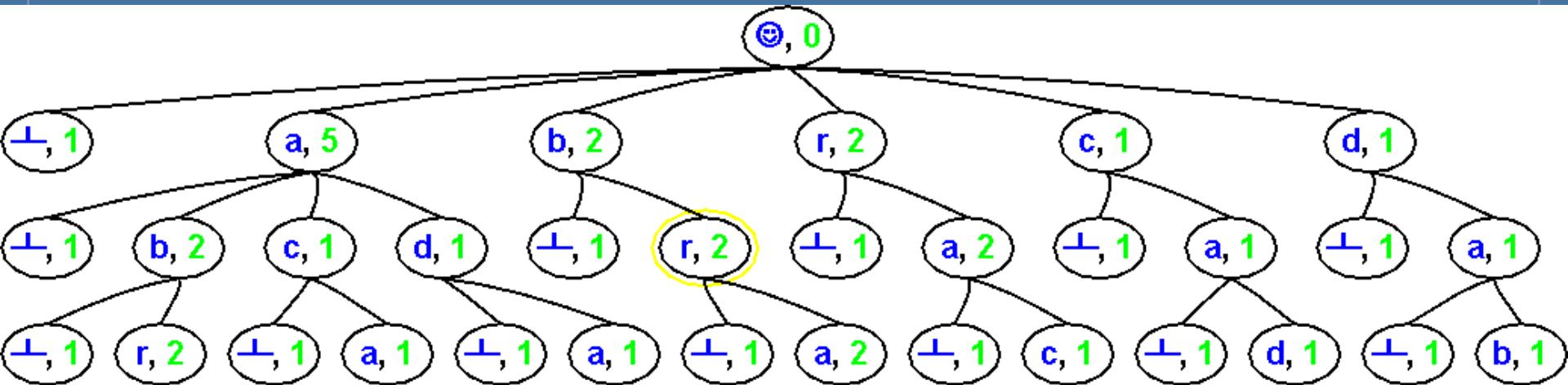
- abracadabra: ‘a’ codificado no contexto ‘br’ em $K = 2$, com prob. $1/2$.
- Modelo atualizado:

Exemplo: PPM

● abracadabra: Modelo atualizado:

Exemplo: PPM

○ Árvore Final:



Exemplo: PPM

○ Saída:

A	B	R	A	C	A	D
ESC	A	ESC	B	ESC	R	A
1 [0,1]	1/5 [0,0.2]	1/2 [0.5,1]	1/4 [0,0.25]	2/4 [0.5,1]	1/3 [0.66,1]	1/6 [0,1/6]
						1 [0,1]

A	B	R	A
A	B	R	A
3/7 [0,3.7]	1/6 [0,1/6]	1/2 [0,0.5]	1/2 [0,0.5]

Codificador Aritmético

- Não utiliza palavras código de tamanho inteiro;
- Toda a mensagem codificada é um único número real;
- Considerado ótimo por atingir a entropia.

Codificador Aritmético

- Exemplo de proporcionalidade de bits:

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$n_1 = 120 \text{ (7 bits)}$$

$$n_2 = 5 \text{ (3 bits)}$$

$$n_1+n_2 = 125 \text{ (7 bits)}$$

$$n_1+n_2+n_2 = 130 \text{ (8 bits)}$$

Codificador Aritmético

Exemplo de proporcionalidade de bits:

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

Equação 7 bits:
 $(120+5+5)*(7/128)$

$$n1 = 120 \text{ (7 bits)}$$

$$n2 = 5 \text{ (3 bits)}$$

$$n1+n2 = 125 \text{ (7 bits)}$$

$$n1+n2+n2 = 130 \text{ (8 bits)}$$

Codificador Aritmético

- Símbolos representados por intervalos de probabilidade
- Ordena os símbolos em ordem decrescente das probabilidades
- A cada novo símbolo codificado o intervalo de probabilidades é ajustado de acordo com o *range* anterior.

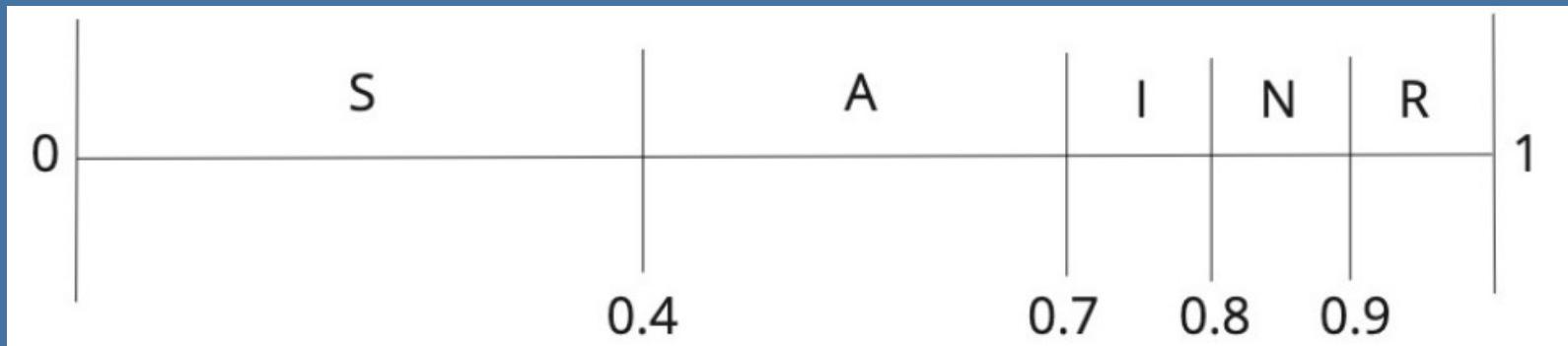
Codificador Aritmético

- Exemplo:
- Mensagem = assassinar
- $A = \{a, i, n, r, s\}$
- Não contextual, semi-adaptativo.

Símbolo	Probabilidade	Faixa
s	4/10	[0, 0.4)
a	3/10	[0.4, 0.7)
i	1/10	[0.7, 0.8)
n	1/10	[0.8, 0.9)
r	1/10	[0.9, 1.0)

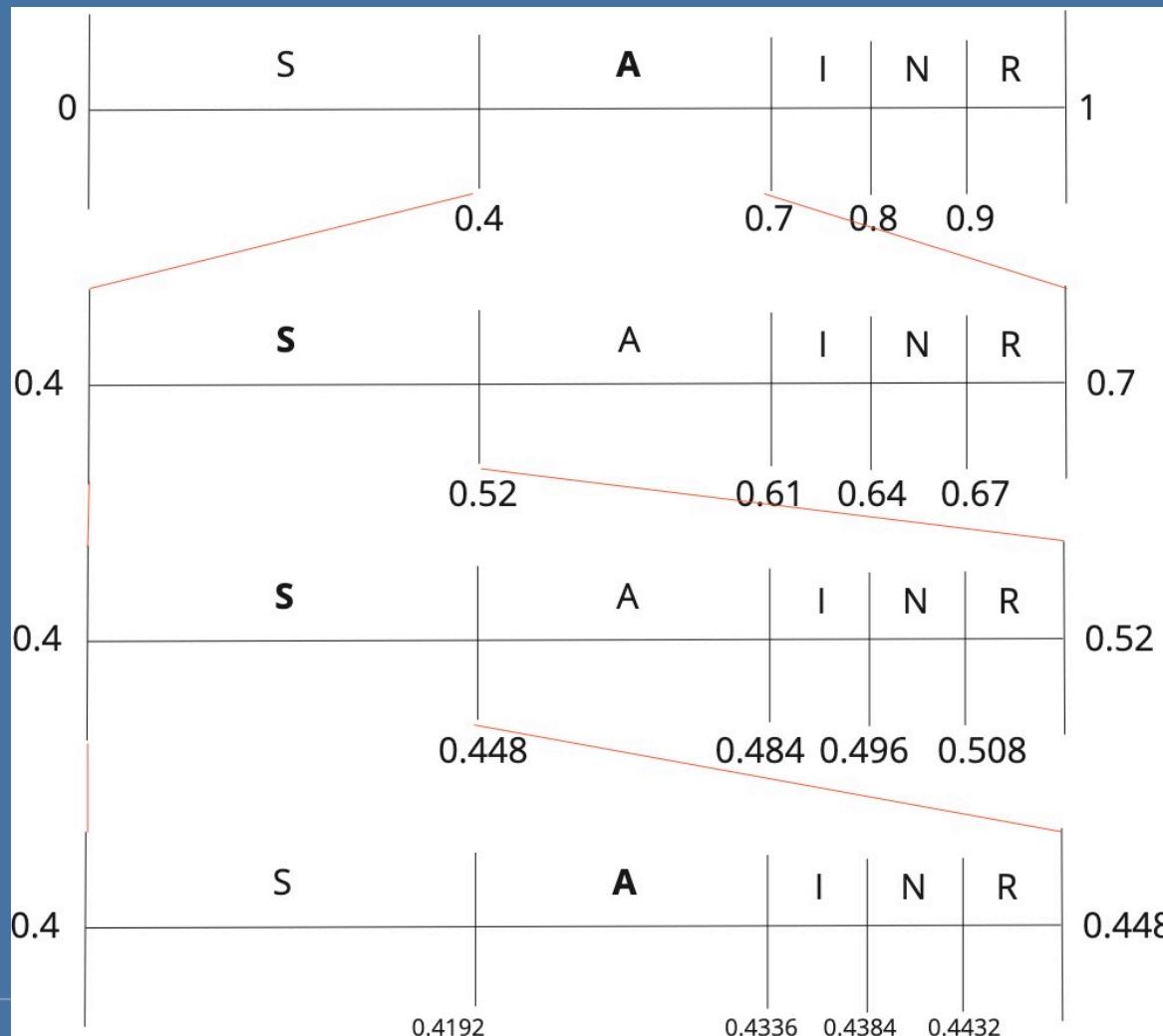
Codificador Aritmético

- ◎ Mensagem = assassinar



Codificador Aritmético

- $M = \text{ASSASSINAR}$. Não contextual, semi-adaptativo.



Codificador Aritmético

- Converte uma sequência de símbolos em um número real.
- Utiliza intervalos de probabilidades

$$l_i = l_{i-1} + r_{i-1} \cdot ls_i$$

$$h_i = l_{i-1} + r_{i-1} \cdot hs_i$$

ls_i: limite inferior do intervalo associado ao símbolo na posição *i*

hs_i: limite superior do intervalo associado ao símbolo na posição *i*

$$r_i = h_i - l_i$$

Codificador Aritmético

Símbolo	l_i	h_i	r_i
	0.0	1.0	1.0
a	$0.0 + 1.0 * 0.4 = 0.4$	$0.0 + 1.0 * 0.7 = 0.7$	0.3
s	$0.4 + 0.3 * 0.0 = 0.4$	$0.4 + 0.3 * 0.4 = 0.52$	0.12
s	$0.4 + 0.12 * 0.0 = 0.4$	$0.4 + 0.12 * 0.4 = 0.448$	0.048
a	$0.4 + 0.048 * 0.4 = 0.4192$	$0.4 + 0.048 * 0.7 = 0.4336$	0.0144
s	0.4192	0.42496	0.00576
s	0.4192	0.421504	0.002304
i	0.4208128	0.4210432	0.0002304
n	0.42099712	0.42102016	0.00002304
a	0.421006336	0.421013248	0.000006912
r	0.4210125568	0.421013248	

○ Saída:

- 0.421013

Codificador Aritmético

- Saída:

- 0.421013

- Binário:

- 0.0110101110001111

Codificador Aritmético

○ Decodificação:

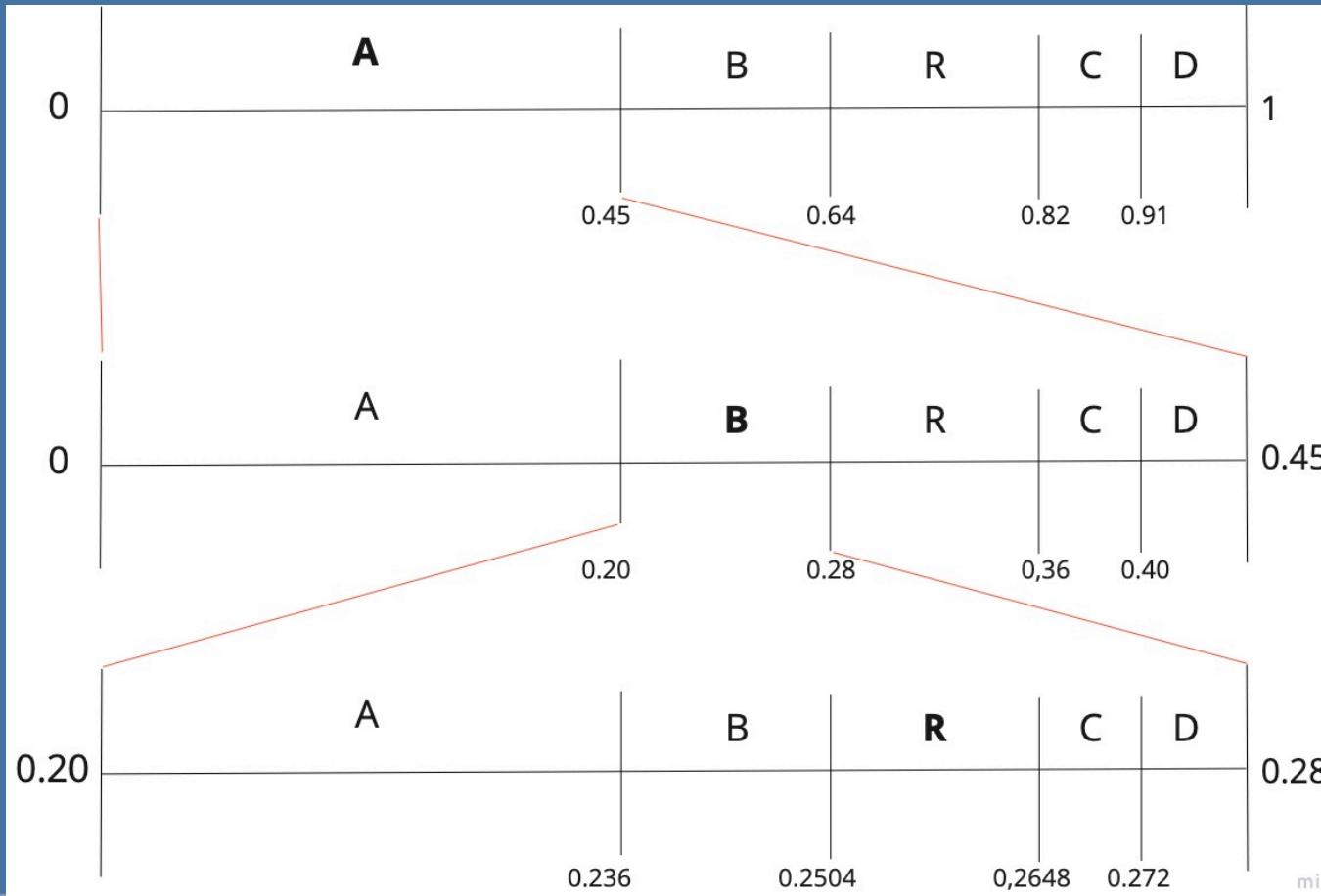
- retirada do efeito do i -ésimo símbolo, s_i , recém-decodificado:
- $v_i = (v_{i-1} - ls_{i-1}) / rs_{i-1}$

Codificador Aritmético

- Exemplo:
- Mensagem = ABRACADABRA
- $A = \{A, B, C, D, R\}$
- Não contextual, semi-adaptativo.

Codificador Aritmético

- $M = \text{ABRACADABRA}$. Não contextual, semi-adaptativo.

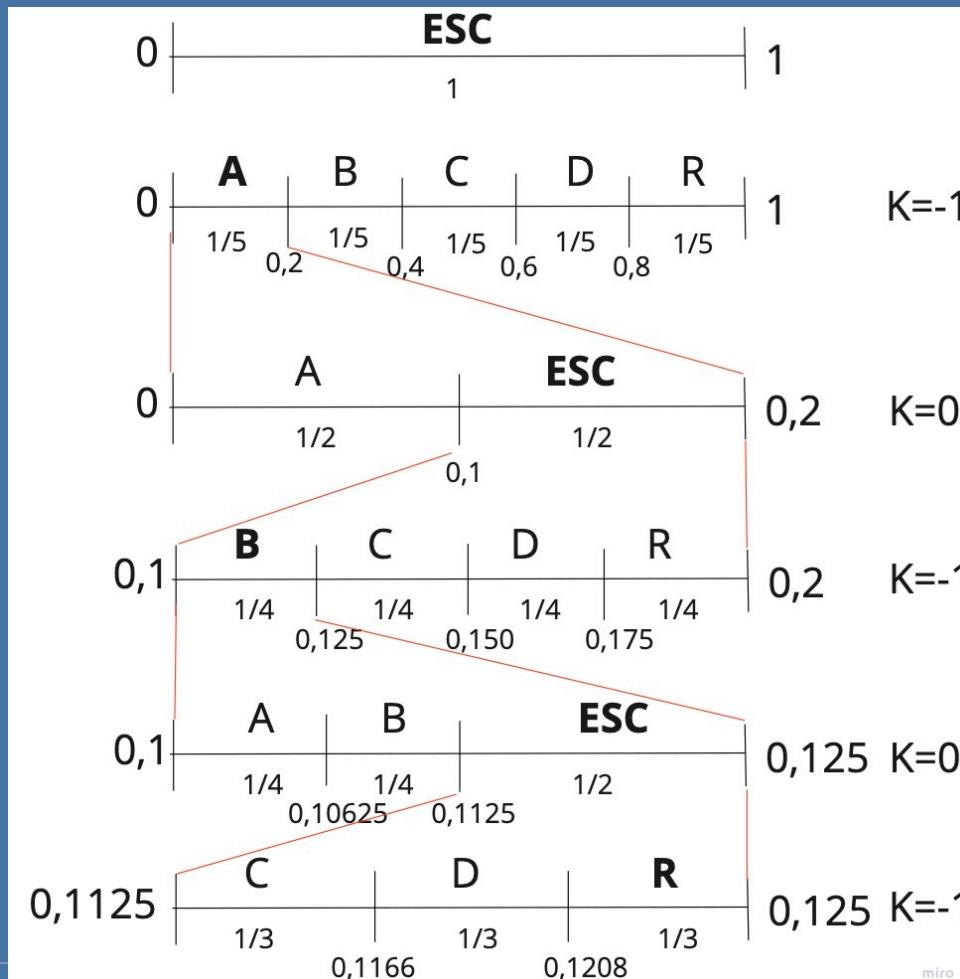


Codificador Aritmético

- Exemplo:
- Mensagem = ABRACADABRA
- $A = \{A, B, C, D, R\}$
- Contextual, adaptativo.
 - PPM

Codificador Aritmético

- M = ABRACADABRA. Contextual, adaptativo.



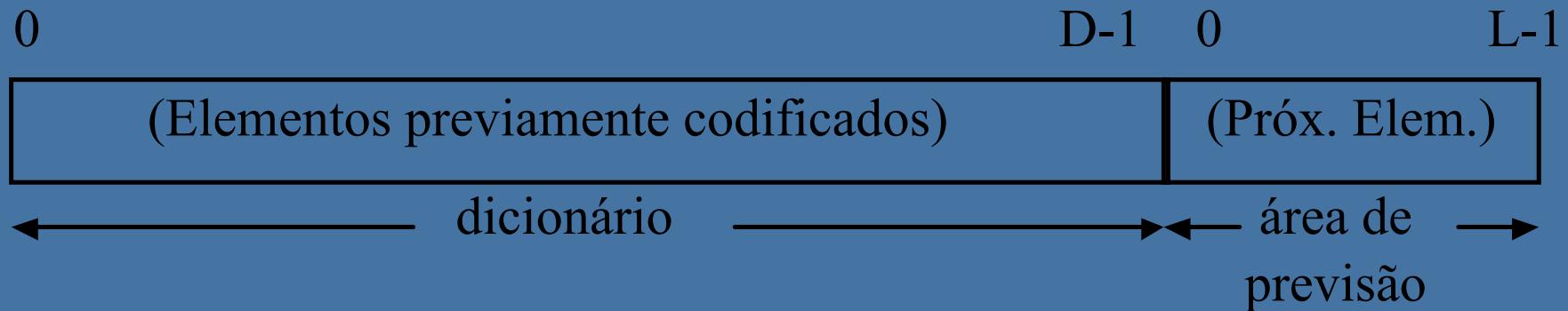
Codificação Baseada em Dicionário

- Ziv e Lempel, 1976: Complexidade de sistemas dinâmicos
- Ziv e Lempel, 1977: LZ77
- Ziv e Lempel, 1978: LZ78
- Welch, 1984: LZW
- LZA, LZM, LZFA, ...

Codificação Baseada em Dicionário

- LZ77:

- Janela deslizante.



Codificação Baseada em Dicionário

• LZ77 - Exemplo:

- W = Segmentos são os componentes do programa. Finalmente, programas são compostos
- A={0,1,2, ..., 254, 255}
- Saída:
 - Tupla (qtd. coincidência (zero), prox. Símbolo)
 - Terno (qtd. coincidência, índice início, prox. símbolo)

Codificação Baseada em Dicionário

○ LZ77:

0	8	16	24	32	40	
						Segmentos são os
						Terno de saída: 0, 'S'
0	8	16	24	32	40	
						Segmentos são os
						Terno de saída: 0, 'e'
0	8	16	24	32	40	
						Segmentos são os c
						Terno de saída: 0, 'g'
0	8	16	24	32	40	
						Segmentos são os co
						Temo de saída: 0, 'm'
0	8	16	24	32	40	
						Segmentos são os com
						Temo de saída: 1, 45, 'n'

Codificação Baseada em Dicionário

○ LZ77:

0	8	16	24	32	40
---	---	----	----	----	----

Segmentos são os componentes do programa. Finalmente, programas

Coincidências: ($n = 1, i = 1$); ($n = 3, i = 4$) e ($n = 4, i = 23$)

Terno de saída: 4, 23, ‘,’

0	8	16	24	32	40
---	---	----	----	----	----

ntos são os componentes do programa. Finalmente, programas são c

Terno de saída: 9, 26, ‘s’

0	8	16	24	32	40
---	---	----	----	----	----

s componentes do programa. Finalmente, programas são compostos t

Temo de saída: 1, 1, ‘s’

Codificação Baseada em Dicionário

- LZ77:
 - Terno (qtd. coincidência, índice início, prox. símbolo)
 - Tamanho da áreas:
 - 4KB de dicionário, e 15bytes de área previsão
 - Ex. armazenamento:
 - (4bits, 12 bits, 8bits)

Codificação Baseada em Dicionário

- LZ78:

- Sem janela deslizante.
- Índices de palavras (frases).
- Saída: Tupla (índice, prox. símbolo)
- Dicionário iniciado com símbolo vazio.
- Adiciona no dicionário índice utilizado + próximo símbolo.

Codificação Baseada em Dicionário

- LZ78 - Exemplo:

- W = Segmentos são os componentes do programa. Finalmente, programas são compostos
- A={0,1,2, ..., 254, 255}
- Tupla (índice, prox. símbolo)

Codificação Baseada em Dicionário

○ LZ78:

Dicionário:

0	"	10	'sã'	20	'o p'	30	'l'	40	'ã'
1	'S'	11	'o '	21	'r'	31	'me'	41	'o c'
2	'e'	12	'os'	22	'og'	32	'nt'	42	'omp'
3	'g'	13	' c'	23	'ra'	33	'e,'	43	'ost'
4	'm'	14	'om'	24	'ma'	34	' p'	44	'os '
5	'en'	15	'p'	25	'.'	35	'ro'	45	'ta'
6	't'	16	'on'	26	' F'	36	'g'		
7	'o'	17	'ent'	27	'i'	37	'ram'		
8	's'	18	'es'	28	'n'	38	'as'		
9	' '	19	' d'	29	'a'	39	' s'		

Saída codificada:

0, 'S', 0, 'e', 0, 'g', 0, 'm', 2, 'n', 0, 't', 0, 'o', 0, 's', 0, ' ', 8, 'ã', 7, ' ', 7, 's', 9, 'c', 7, 'm', 0, 'p', 7, 'n', 5, 't', 2, 's', 9, 'd', 11, 'p', 0, 'r', 7, 'g', 21, 'a', 4, 'a', 0, '.', 9, 'F', 0, 'i', 0, 'n', 0, 'a', 0, 'l', 4, 'e', 28, 't' ...

Codificação Baseada em Dicionário

- LZ78:

- Tupla (índice, prox. símbolo)
- Tamanho da áreas:
 - Dicionário com 1024 símbolos, e próximo símbolo com 1 byte.
- Ex. armazenamento:
 - (10bits, 8bits)
 - (16bits, 8bits)

Codificação Baseada em Dicionário

- LZW:

- Baseado no LZ78
- Dicionário pré-inicializado com alfabeto.
- Saída: apenas o índice.
- Adiciona no dicionário índice utilizado + próximo símbolo.

Exemplo LZW

- Exemplo: $W = \text{abracadabra}$; alfabeto = $\{a, b, c, d, r\}$
- Codificação
- Dicionário inicial:

0 a

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

- Frase 'a' está no dicionário
- Frase 'ab' não está no dicionário
- Mensagem codificada = 0 (última frase encontrada no dicionário)
- Adiciona o 'a' seguido do próximo símbolo ('b')

Dicionário:

0 a 5 ab

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

- ‘b’ está no dicionário; ‘br’ não está no dicionário
- Mensagem codificada = 01

Dicionário:

0 a 5 ab

1 b 6 br

2 c

3 d

4 r

Exemplo LZW (Cont.)

- ‘r’ está no dicionário; ‘ra’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’

Dicionário:

0 a	5 ab
1 b	6 br
2 c	7 ra
3 d	
4 r	

Exemplo LZW (Cont.)

- ‘a’ está no dicionário; ‘ac’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’ ‘0’

Dicionário:

0 a	5 ab
1 b	6 br
2 c	7 ra
3 d	8 ac
4 r	

Exemplo LZW (Cont.)

- ‘c’ está no dicionário; ‘ca’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’ ‘0’ ‘2’

Dicionário:

0 a	5 ab
1 b	6 br
2 c	7 ra
3 d	8 ac
4 r	9 ca

Exemplo LZW (Cont.)

- ‘a’ está no dicionário; ‘ad’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’ ‘0’ ‘2’ ‘0’

Dicionário:

0 a	5 ab	10 ad
1 b	6 br	
2 c	7 ra	
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- ‘d’ está no dicionário; ‘da’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’ ‘0’ ‘2’ ‘0’ ‘3’

Dicionário:

0 a	5 ab	10 ad
1 b	6 br	11 da
2 c	7 ra	
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- ‘a’ está no dicionário; ‘ab’ está no dicionário; ‘abr’ não está no dicionário
- Mensagem codificada = ‘0’ ‘1’ ‘4’ ‘0’ ‘2’ ‘0’ ‘3’ ‘5’

Dicionário:

0 a	5 ab	10 ad
1 b	6 br	11 da
2 c	7 ra	12 abr
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- 'r' está no dicionário; 'ra' está no dicionário; EOF
- Mensagem codificada = '0' '1' '4' '0' '2' '0'
 '3' '5' '7'

Dicionário:

0 a	5 ab	10 ad
1 b	6 br	11 da
2 c	7 ra	12 abr
3 d	8 ac	13 raEOF
4 r	9 ca	

Exemplo LZW (Cont.)

- Decodificação
- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- Dicionário inicial:

0 a

1 b

2 c

3 d

4 r

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a
- Dicionário :

0 a	5 a?
1 b	
2 c	
3 d	
4 r	

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b

- Dicionário :

0 a	5 ab
-----	------

1 b	6 b?
-----	------

2 c

3 d

4 r

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r

- Dicionário :

0 a	5 ab
-----	------

1 b	6 br
-----	------

2 c	7 r?
-----	------

3 d	
-----	--

4 r	
-----	--

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a
- Dicionário :

0 a	5 ab
1 b	6 br
2 c	7 ra
3 d	8 a?
4 r	

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a c

- Dicionário :

0 a	5 ab
-----	------

1 b	6 br
-----	------

2 c	7 ra
-----	------

3 d	8 ac
-----	------

4 r	9 c?
-----	------

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a c a

Dicionário :

0 a	5 ab	10 a?
1 b	6 br	
2 c	7 ra	
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a c a d

- Dicionário :

0 a	5 ab	10 ad
1 b	6 br	11 d?
2 c	7 ra	
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a c a d ab

Dicionário :

0 a	5 ab	10 ad
1 b	6 br	11 da
2 c	7 ra	12 ab?
3 d	8 ac	
4 r	9 ca	

Exemplo LZW (Cont.)

- W codificada: '0' '1' '4' '0' '2' '0' '3' '5' '7'
- W: a b r a c a d ab ra

- Dicionário :

0 a	5 ab	10 ad
1 b	6 br	11 da
2 c	7 ra	12 abr
3 d	8 ac	13 raEOF
4 r	9 ca	

LZW - Exemplo 2

- Exemplo: $W = aabababaaa$; alfabeto = {a, b}
- Codificação
- Dicionário inicial:

0 a

1 b

LZW - Exemplo 2 (Cont.)

- $W = \text{aabababaaa}$
- ‘a’ está no dicionário; ‘aa’ não está no dicionário
- Mensagem codificada: 0
- Dicionário:

0 a

1 b

2 aa

LZW - Exemplo 2 (Cont.)

- $W = aabababaaa$
- ‘a’ está no dicionário; ‘ab’ não está no dicionário
- Mensagem codificada: ‘0’ ‘0’
- Dicionário:

0 a

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

- $W = \text{aabababaaa}$
- ‘b’ está no dicionário; ‘ba’ não está no dicionário
- Mensagem codificada: ‘0’ ‘0’ ‘1’
- Dicionário:

0 a 4 ba

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

- $W = aabababaaa$
- ‘a’ está no dicionário; ‘ab’ está no dicionário; ‘aba’ não está no dicionário
- Mensagem codificada: ‘0’ ‘0’ ‘1’ ‘3’
- Dicionário:

0 a 4 ba

1 b 5 aba

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

- $W = aabababaaa$
- ‘a’ está no dic.; ‘ab’ está no dic.; ‘aba’ está no dic.; ‘abaa’ não está no dic.
- Mensagem codificada: ‘0’ ‘0’ ‘1’ ‘3’ ‘5’
- Dicionário:

0 a	4 ba
1 b	5 aba
2 aa	6 abaa
3 ab	

LZW - Exemplo 2 (Cont.)

- W = aabababaaa
- 'a' está no dic.; 'aa' está no dic.; aaEOF não está no dic.
- Mensagem codificada: '0' '0' '1' '3' '5' '2'
- Dicionário:

0 a	4 ba
1 b	5 aba
2 aa	6 abaa
3 ab	7 aaEOF

LZW - Exemplo 2 (Cont.)

- Decodificação

- W codificada: '0' '0' '1' '3' '5' '2'

- Dicionário inicial:

0 a

1 b

LZW - Exemplo 2 (Cont.)

- W codificada: '0' '0' '1' '3' '5' '2'

- W: a

- Dicionário:

0 a

1 b

2 a?

LZW - Exemplo 2 (Cont.)

- W codificada: '0' '0' '1' '3' '5' '2'

- W: a a

- Dicionário:

0 a

1 b

2 aa

3 a?

LZW - Exemplo 2 (Cont.)

- W codificada: '0' '0' '1' '3' '5' '2'

- W:
a a b

- Dicionário:

0 a 4 b?

1 b

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

- W codificada: '0' '0' '1' '3' '5' '2'

- W: a a b ab

- Dicionário:

0 a	4 ba
-----	------

1 b	5 ab?
-----	-------

2 aa

3 ab

LZW - Exemplo 2 (Cont.)

- Mas lo símbolo da frase em 5 é o último símbolo da última frase (5):
- W codificada: '0' '0' '1' '3' '5' '2'
- W: a a b ab aba
- Dicionário:

0 a	4 ba
1 b	5 aba
2 aa	6 aba?
3 ab	

LZW - Exemplo 2 (Cont.)

- W codificada: '0' '0' '1' '3' '5' '2'
- W: a a b ab aba aa

- Dicionário:

0 a	4 ba
1 b	5 aba
2 aa	6 abaa
3 ab	7 aaEOF

LZW - Observações

- Dicionário de tamanho limitado a P frases
- Número de bits, b, dos índices das frases pode crescer conforme necessário, a medida em que novas frases são inseridas
- No LZW padrão, índices de tamanho fixo $b = \log_2 P$

LZW - Observações

- O que fazer quando o dicionário enche?
 - Eliminar frases LRU (Least **R**ecently Used)
 - Eliminar frases menos utilizadas
 - Manter dicionário estático
 - Reiniciar dicionário
 - Monitorar compressão e, caso se detecte redução na RC, reiniciar dicionário.

Burrows-Wheeler Transform (BWT)

- Transformada reversível criada por Wheeler in 1983, mas não publicada
- Aplicação para compressão: Michael Burrows and David Wheeler, 1994
- BWT não comprime, mas torna a seqüência mais apropriada à compressão por métodos simples

Burrows-Wheeler Transform (BWT)

- Mensagem com N símbolos
- Passos:
 1. Listar as N mensagens diferentes obtidas por rotações (deslocamentos cílicos) da mensagem original
 2. Ordenar lexicograficamente as N mensagens obtidas com as rotações e indexar a lista ordenada
 3. Reter o índice da mensagem original e o último símbolo das mensagens da lista ordenada, preservando a ordem

BWT

- Exemplo:
- $W = \text{quequalquasequando}$ ($N = 18$);
Alfabeto = {a, d, e, l, n, o, q, s, u}

- Rotações:

quequalquasequando

oquequalquasequand

doquequalquasequan

...

BWT – Rotações

quequalquasequando
o quequalquasequand
do queequalquasequau
ndo queequalquasequa
ando queequalquasequ
uando queequalquaseq
quando queequalquase
equando queequalquas
sequando queequalqua
asequando queequalqu
uasequando queequalq
quasequando queequal
lquasequando queequa
alquasequando queequ
ualquasequando queeq
qualquasequando que
equalquasequando qu
uequalquasequando q

BWT – Ordenação e Indexação

0 alquasequandoquequ
1 andoqueuequalquasequ
2 asequandoqueequalqu
3 doqueuequalquasequan
4 equalquasequandoqu
5 equandoqueequalquas
6 lquasequandoquequa
7 ndoqueuequalquasequa
8 oqueuequalquasequand
9 qualquasequandoque
10 quandoqueequalquase
11 quasequandoqueequal
12 queequalquasequando
13 sequandoqueequalqua
14 ualquasequandoqueq
15 uandoqueequalquaseq
16 uasequandoqueequalq
17 uequalquasequandoq

BWT – Passo 3

- ◎ Mensagem transformada:

12 uuunusaadeeloaqqqqq

BWT – Passo 3

- A BWT informa explicitamente o último símbolo de cada rotação na lista ordenada
- Lista ordenada -> sabe-se o símbolo que se segue a cada um destes símbolos finais, na mensagem original

BWT – Passo 3

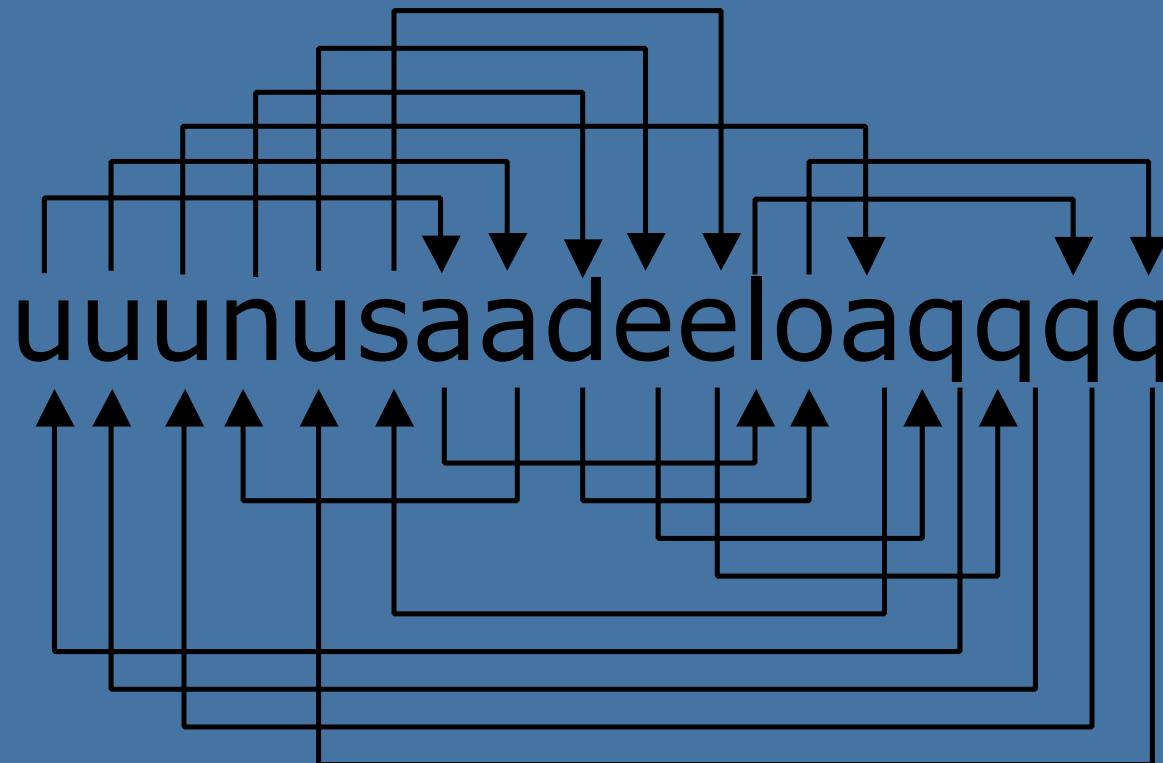
- O 1º símbolo da 1ª rotação ('a') encontra-se na última posição da rotação de índice 6, e é o símbolo que se segue ao último símbolo da 1ª rotação ('u')
- O 1º símbolo da 2ª rotação ('a') encontra-se na última posição da rotação de índice 7, e é o símbolo que se segue ao último símbolo da 2ª rotação ('u')

BWT – Passo 3

- O 1º símbolo da 3ª rotação ('a') encontra-se na última posição da rotação de índice 13, e é o símbolo que se segue ao último símbolo da 3ª rotação ('u')
- O 1º símbolo da 4ª rotação ('d') encontra-se na última posição da rotação de índice 8, e é o símbolo que se segue ao último símbolo da 4ª rotação ('n')
- ...

BWT – Passo 3

- Localizam-se os símbolos seguintes observando-se apenas a BWT:



BWT – Passo 3

- Índice do símbolo na BWT->índice do símbolo seguinte na mensagem original:

u	u	u	n	u	s	a	a	d	e	e	l	o	a	q	q	q	q
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	7	13	8	9	10	11	3	12	14	15	16	17	5	0	1	2	4

BWT – Decodificação

- Índice da mensagem original informa a posição, na BWT, do último símbolo da mensagem
- Mapeamento informa o índice do símbolo seguinte, ou seja, do 1º símbolo da mensagem
- Seguindo-se o mapeamento, encontram-se todos os símbolos seguintes

BWT – Decodificação

- 12 -> 17 (q) -> 4 (u) -> 9(e) -> 14(q) ->
0(u) -> 6(a) -> 11(l) -> 16(q) -> 2(u) ->
13(a) -> 5(s) -> 10(e) -> 15(q) -> 1(u) ->
7(a) -> 3(n) -> 8(d) -> 12(o)

u	u	u	n	u	s	a	a	d	e	e	l	o	a	q	q	q	q
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
6	7	13	8	9	10	11	3	12	14	15	16	17	5	0	1	2	4

Compressão com BWT

- Passos:

- BWT
- Move-To-Front
- Codificação de entropia (aritmético)

Move-To-Front (MTF)

- Inicialmente, símbolos são ordenados e indexados (índices 0, 1, ...)
- Na codificação, cada símbolo da mensagem é substituído pelo seu índice, e então passa a ocupar a 1^a posição na lista

MTF - exemplo

- $W = \text{quequalquasequando}$
- $\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	d	e	l	n	o	q	s	u

- Codificação do 1º símbolo: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	u	a	d	e	l	n	o	q	s

- Codificação do 2º símbolo: 0
- Codificação do 3º símbolo: 0
- Codificação do 4º símbolo: 5

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	n	u	a	d	e	l	o	q	s

- Codificação do 5º símbolo: 1

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	u	n	a	d	e	l	o	q	s

- Codificação do 6º símbolo: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	s	u	n	a	d	e	l	o	q

- Codificação do 7º símbolo: 3

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	s	u	n	d	e	l	o	q

- Codificação do 8º símbolo: 0
- Codificação do 9º símbolo: 4

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	d	a	s	u	n	e	l	o	q

- Codificação do 10º símbolo: 5

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	e	d	a	s	u	n	l	o	q

- Codificação do 11º símbolo: 0
- Codificação do 12º símbolo: 6

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	l	e	d	a	s	u	n	o	q

- Codificação do 13º símbolo: 7

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	o	l	e	d	a	s	u	n	q

- Codificação do 14º símbolo: 4

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	a	o	l	e	d	s	u	n	q

- Codificação do 15º símbolo: 8

MTF - exemplo

$\text{BWT}(W) = \text{uuunusaadeeloaqqqq}$

Índice	0	1	2	3	4	5	6	7	8
Símbolo	q	a	o	l	e	d	s	u	n

- Codificação do 16º símbolo: 0
- Codificação do 17º símbolo: 0
- Codificação do 18º símbolo: 0

MTF - exemplo

- $\text{MTF}(\text{BWT}(W)) = 8\ 0\ 0\ 5\ 1\ 8\ 3\ 0\ 4\ 5\ 0\ 6\ 7$
4 8 0 0 0
- No exemplo, Huffman semi-adaptativo não-contextual produz 53 bits sobre W , e 48 bits sobre $\text{MTF}(\text{BWT}(W))$
- O ganho de compressão tende a aumentar com o aumento de W
- Modelos não-contextuais são eficientes com a BWT, que já explora informação contextual

Modelos Determinísticos

- Modelos estatísticos contextuais: número de contextos diferentes cresce exponencialmente com o aumento do tamanho do contexto
- Alternativa: modelos determinísticos de predição
- Para sinais físicos: predição polinomial de ordem K

Predição Polinomial de Ordem K

- Predição do próximo símbolo baseada nos últimos K símbolos
- Mensagem: $W = w_0w_1\dots w_{i-1}$
- Predição para o próximo símbolo: v_i
- Valor real do próximo símbolo: w_i
- Erro de predição: $e_i = w_i - v_i$

Predição Polinomial de Ordem K

K	v_i	e_i
1	w_{i-1}	$w_i - w_{i-1}$
2	$2w_{i-1} - w_{i-2}$	$w_i - 2w_{i-1} + w_{i-2}$
3	$3w_{i-1} - 3w_{i-2} + w_{i-3}$	$w_i - 3w_{i-1} + 3w_{i-2} - w_{i-3}$

Predição Polinomial de Ordem K

- Exemplo (mensagem com 32 símbolos)

$W = \begin{matrix} 7 & 6 & 5 & 4 & 4 & 4 & 3 & 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 1 \\ 1 & 2 & 3 & 5 & 5 & 5 & 6 & 6 & 6 & 7 & 7 & 7 \end{matrix}$

$N_0 = N_1 = N_2 = \dots = N_7 \rightarrow H = 3 \text{ bits/símbolo} =$
comprimento médio do código de Huffman

Predição Polinomial de Ordem K

- Exemplo (mensagem com 32 símbolos)

$K = 1 \rightarrow V = 7 -1 -1 -1 0 0 -1 1 -1 -1 -1$
 $0 0 0 1 1 1 -1 -1 0 1 1 2 0 0 1$
 $0 0 1 0 0$

$N_0 = 12; N_{-1} = 10; N_1 = 8; N_2 = 1; N_7 = 1$

$H = 1,87$ bits/símbolo

Compr. médio código de Huffman = 2 bits/
símbolo

- Neste caso, não codificar 1o símbolo aumenta RC

Predição Polinomial de Ordem K

- Exemplo (mensagem com 32 símbolos)

$K = 2 \rightarrow V = 7 \ 6 \ 0 \ 0 \ 1 \ 0 \ -1 \ 2 \ -2 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0$
 $1 \ 0 \ 0 \ -2 \ 0 \ 1 \ 1 \ 0 \ 1 \ -2 \ 0 \ 1 \ -1 \ 0$
 $1 \ -1 \ 0$

$N_0 = 15; N_1 = 8; N_{-1} = 3; N_{-2} = 3; N_2 = 1; N_6 = 1; N_7 = 1$
 $\rightarrow H = 1,76 \text{ bits/símbolo}$

Compr. médio código de Huffman = 2,16
bits/símbolo

Predição Polinomial de Ordem K

- Exemplo (mensagem com 32 símbolos)

$K = 3 \rightarrow V = 7 \ 6 \ 5 \ 4 \ 3 \ 5 \dots$

$H = ?$

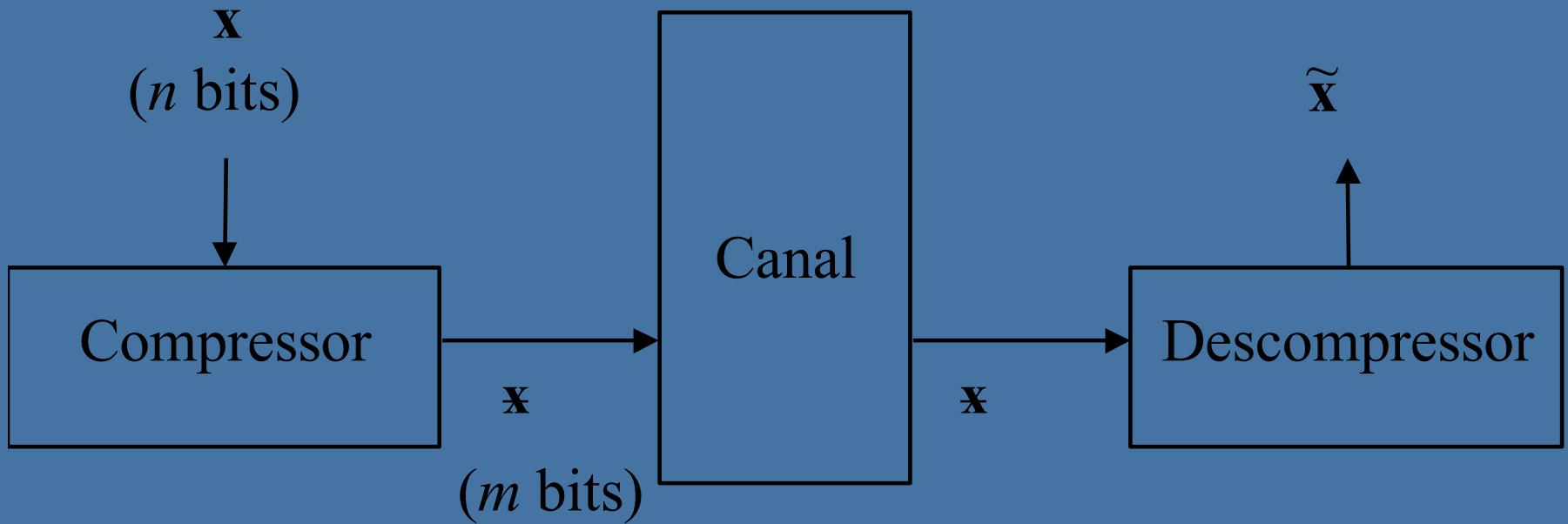
Compr. médio código de Huffman = ?

Compressão com perdas

- Em alguns casos, distorções introduzidas pela compressão são aceitáveis
- Ex: vídeo, imagens, som, outros sinais
- Imagens e sinais críticos (e. g. utilizados no diagnóstico médico): distorções devem ser cuidadosamente controladas
- Eficácia medida pelo compromisso entre RC e distorção
- Sub-amostragem ou quantização (redução de alfabeto)

Compressão com perdas

- Compressor/descompressor genérico



Quantização Escalar

- Opera por redução de alfabeto
- Símbolos tratados individualmente
- Quantização uniforme: divisão seguida por arredondamento para inteiro.

- Valor quantizado:

$$\hat{x} = x // q$$

q: tamanho do passo
de quantização

- Valor dequantizado:

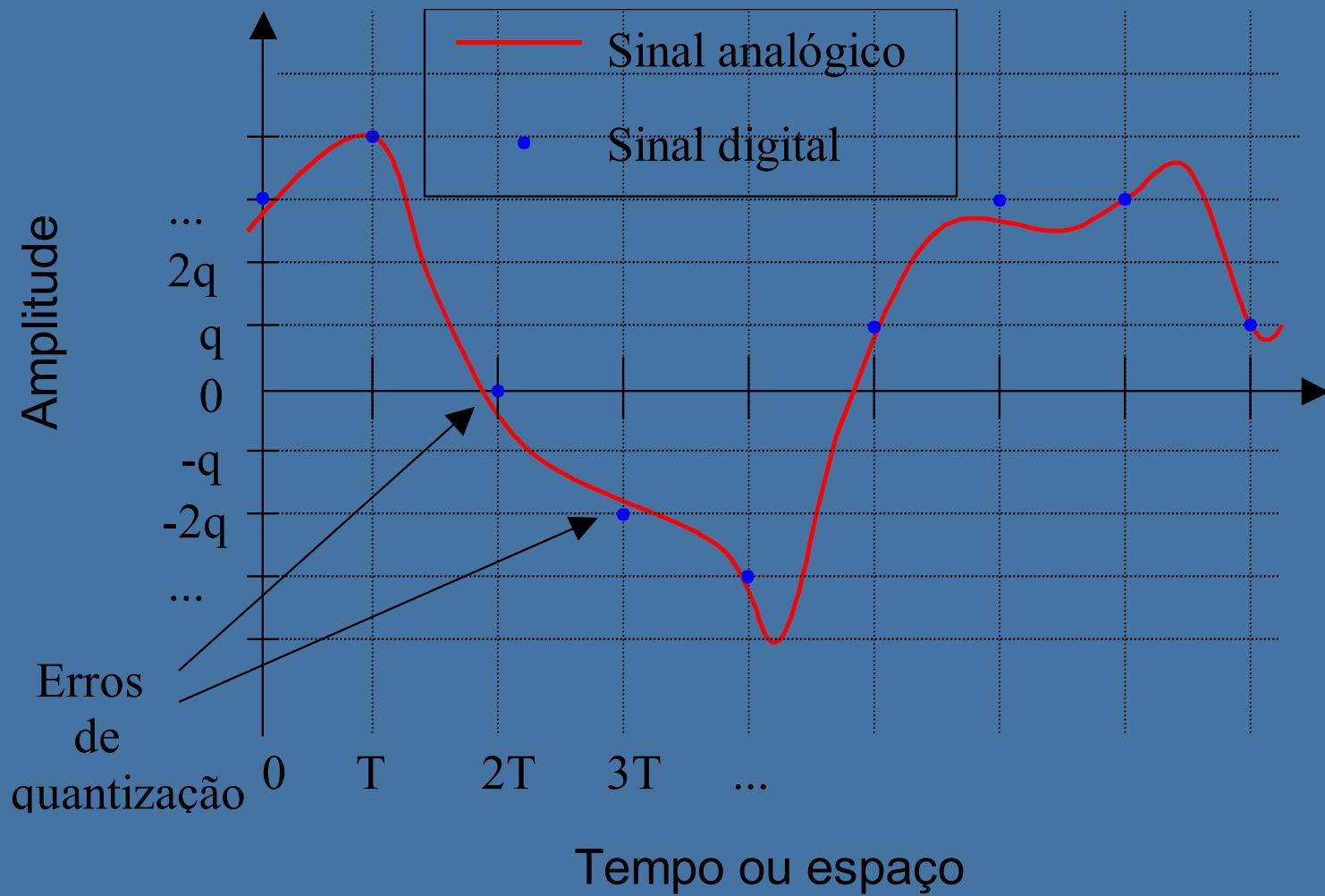
$$\tilde{x} = q\hat{x}$$

Quantização Escalar

- Ex: Alfabeto $A = \{0, 1, \dots, 255\}$
- $q = 2 \quad \rightarrow Q_2 = \{0, 1, \dots, 127\}$
 $\rightarrow D_2 = \{0, 2, 4, \dots, 254\}$
- $q = 3 \quad \rightarrow Q_3 = \{0, 1, \dots, 85\}$
 $\rightarrow D_3 = \{0, 3, 6, \dots, 255\}$
- $q = 4 \quad \rightarrow Q_4 = \{0, 1, \dots, 63\}$
 $\rightarrow D_4 = \{0, 4, 8, \dots, 252\}$

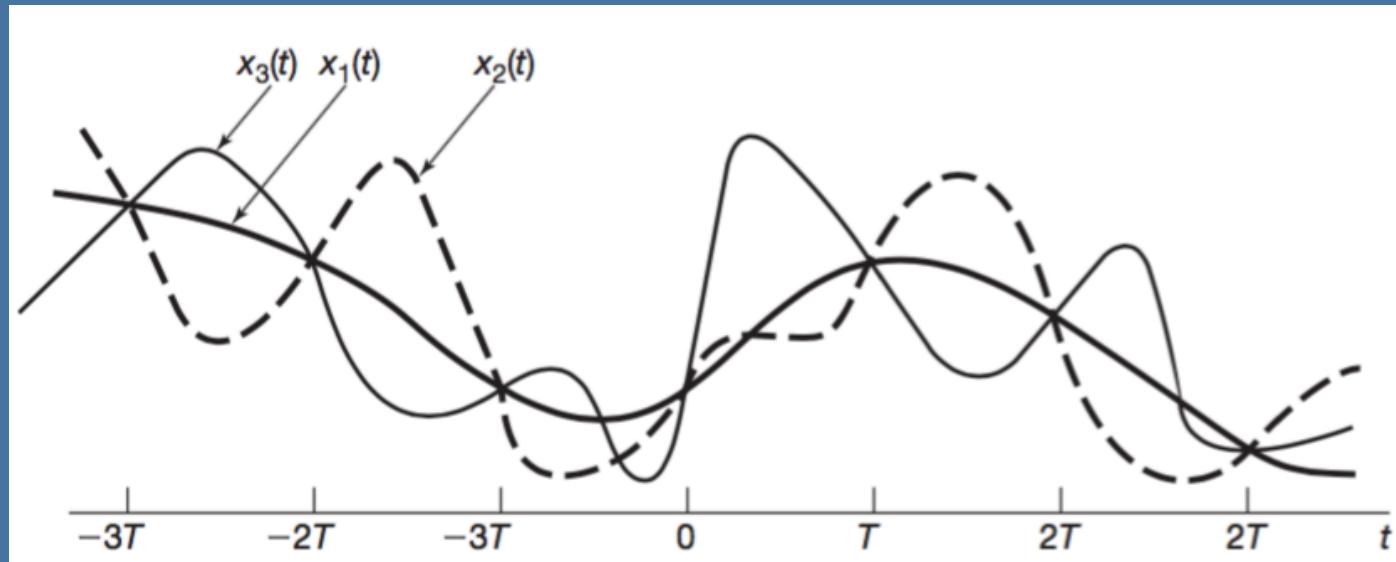
$$E_{\max} = \lfloor q / 2 \rfloor \text{ (Erro Máximo)}$$

Amostragem e Quantização



Amostragem

- Infinitos sinais podem gerar as mesmas amostras.



Amostragem

○ Teorema da amostragem:

- Se o sinal for limitado em banda, isto é,
 $X(jw)=0$, para $|w| > w_M$
- $x(t)$ é representado unicamente por suas amostras se:

$$w_s > 2w_M$$

$w_s \rightarrow$ Frequência de amostragem
(sampling)

$w_M \rightarrow$ Maior frequência sinal original

- Caso a condição não seja respeitada acontecerá *aliasing* em partes do sinal.

Amostragem

○ Reconstrução sinal – Interpolação:

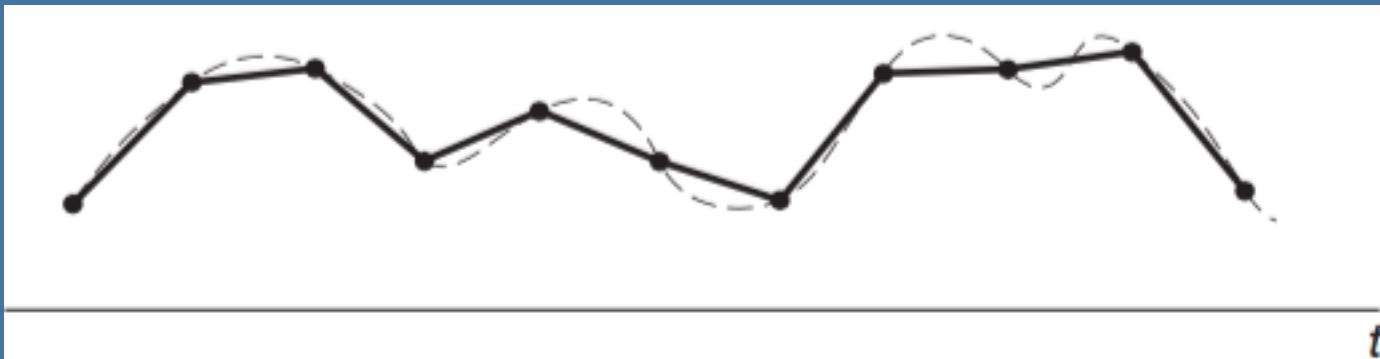
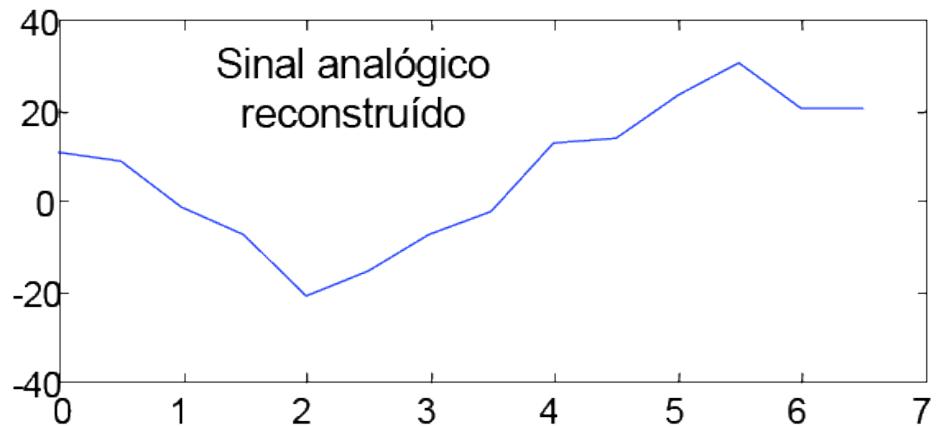
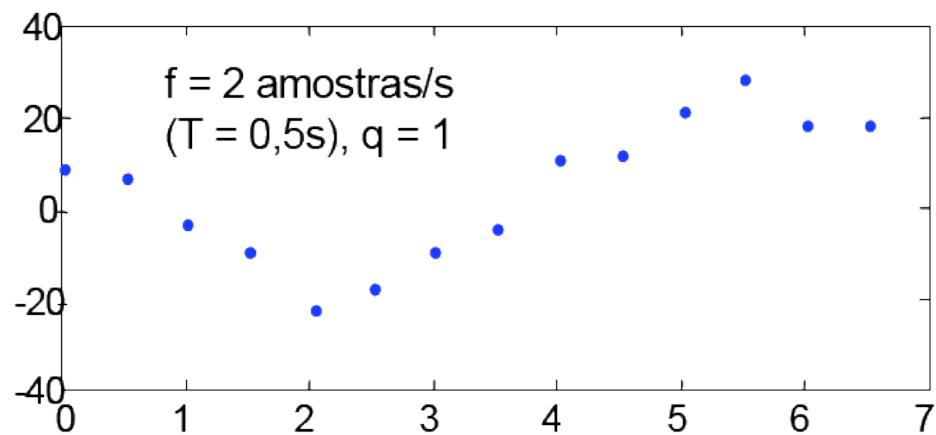
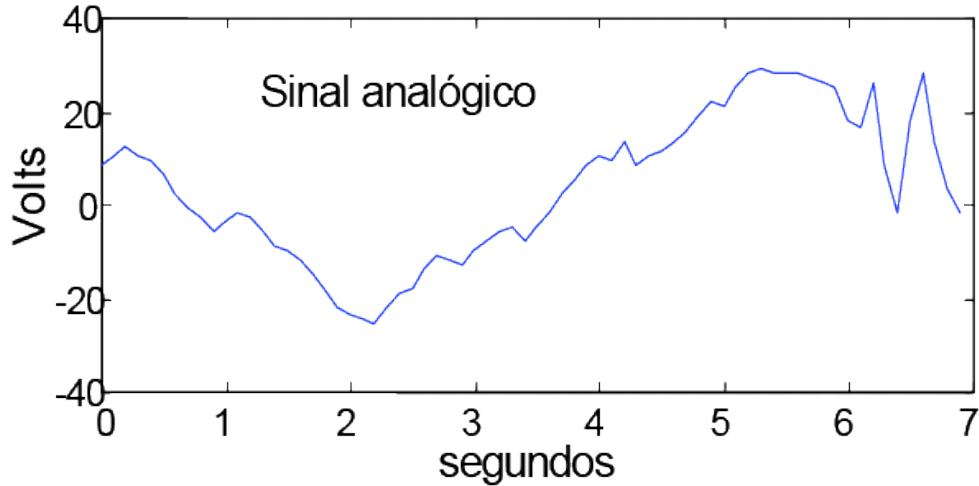


Figura 7.9 Interpolação linear entre os pontos das amostras. A curva tracejada representa o sinal original, e a curva sólida, a interpolação linear.

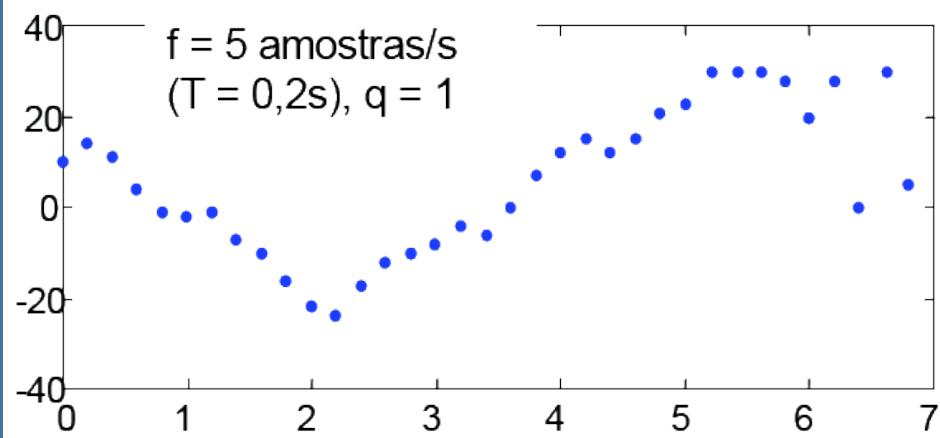
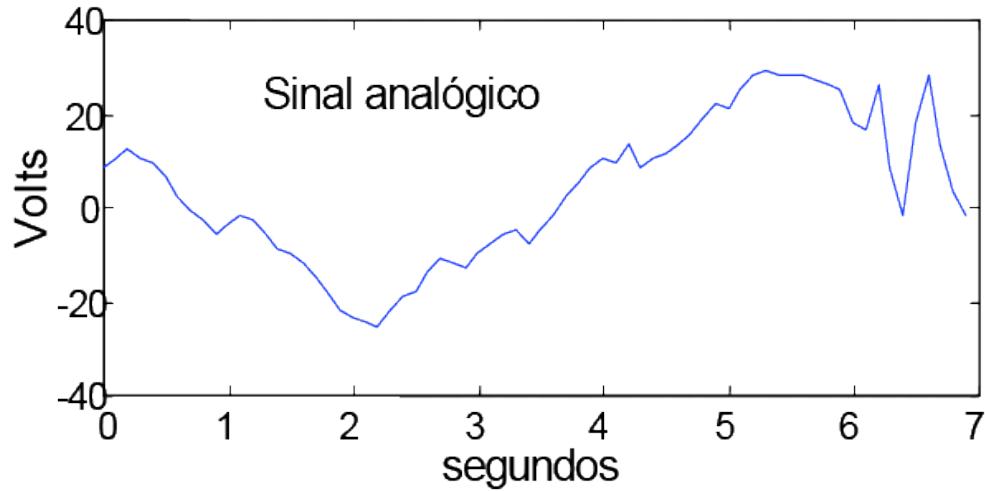
Amostragem

○ Reconstrução sinal – Interpolação:



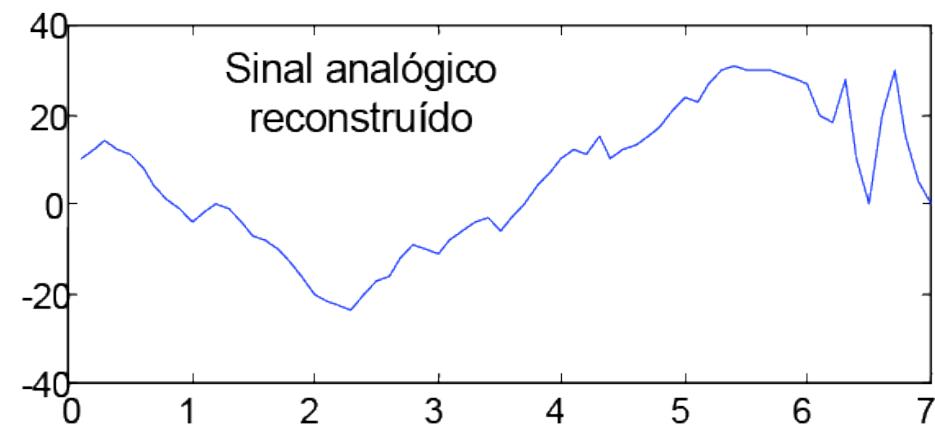
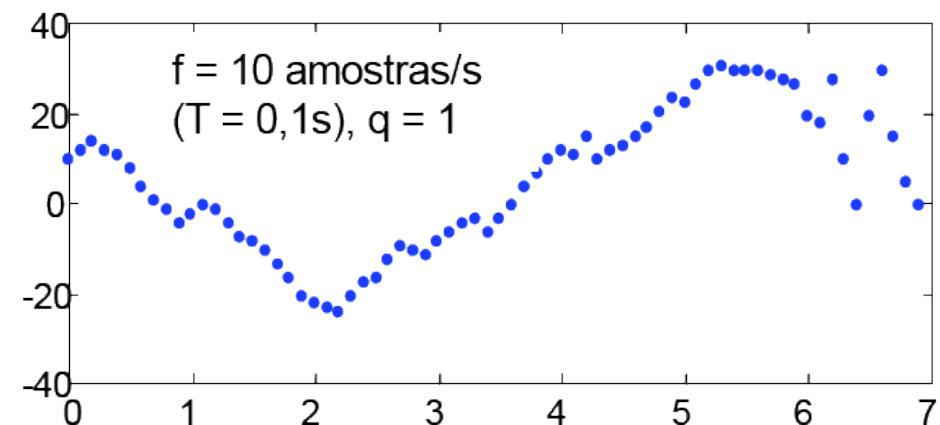
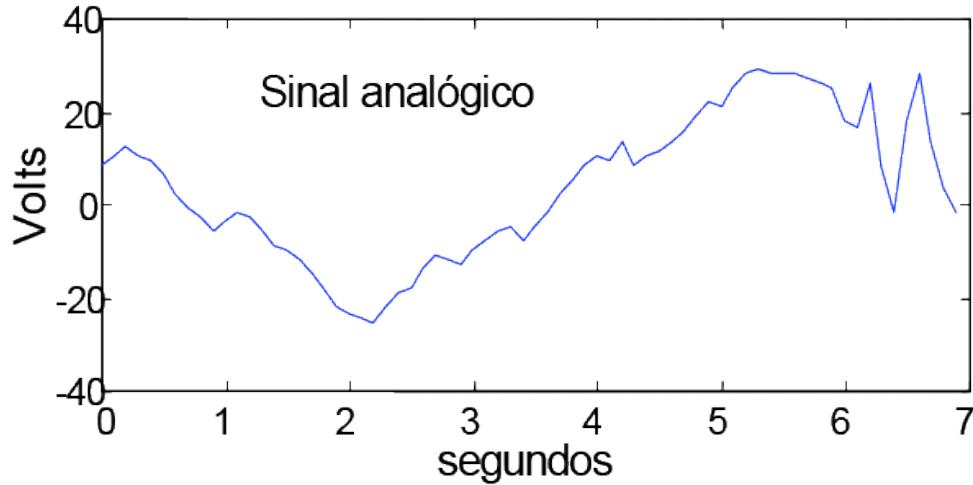
Amostragem

○ Reconstrução sinal – Interpolação:



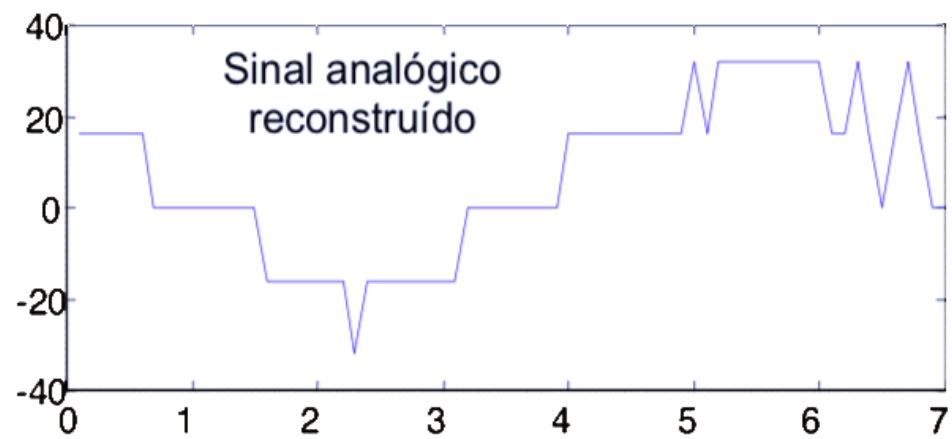
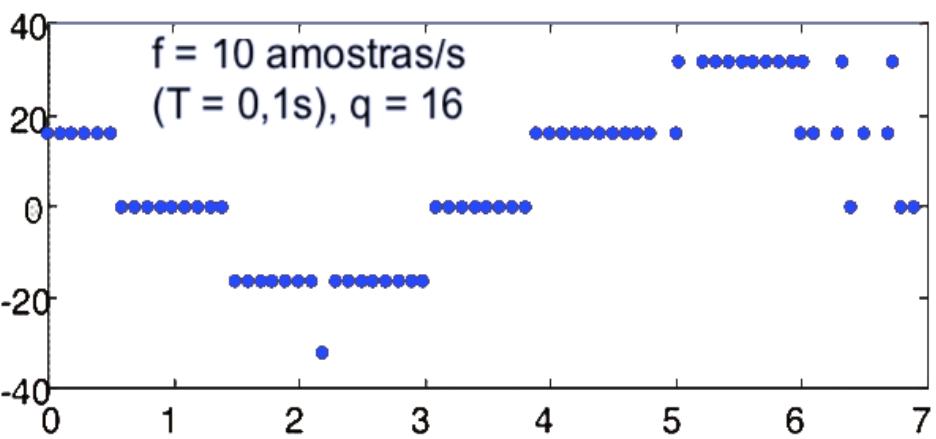
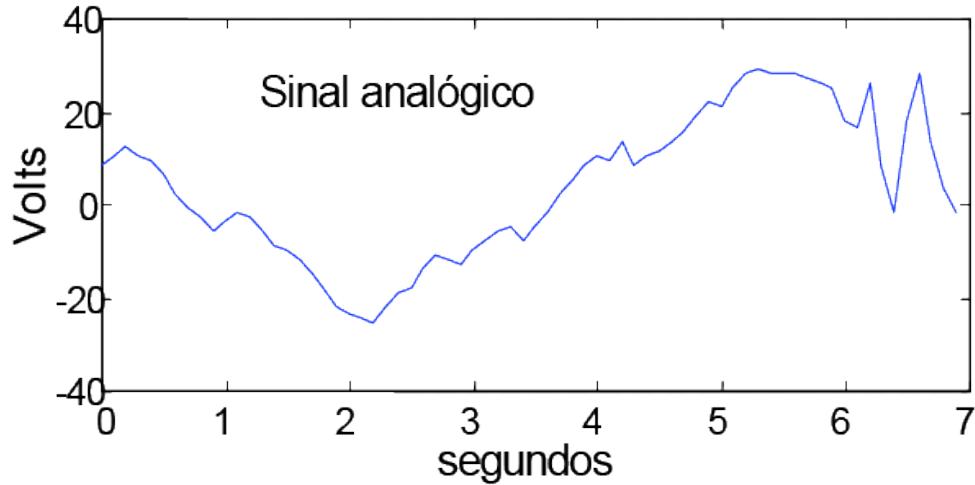
Amostragem

○ Reconstrução sinal – Interpolação:



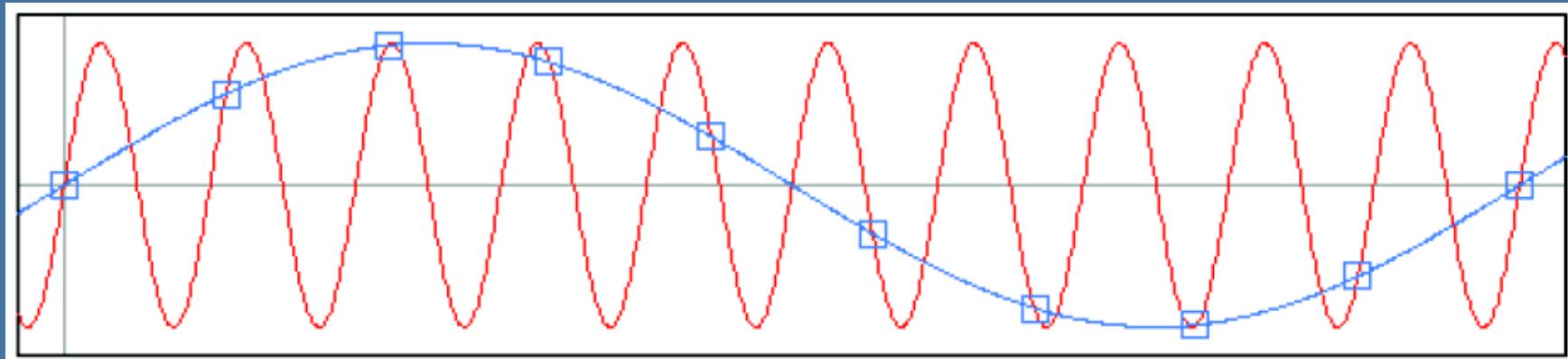
Amostragem

○ Reconstrução sinal – Interpolação:



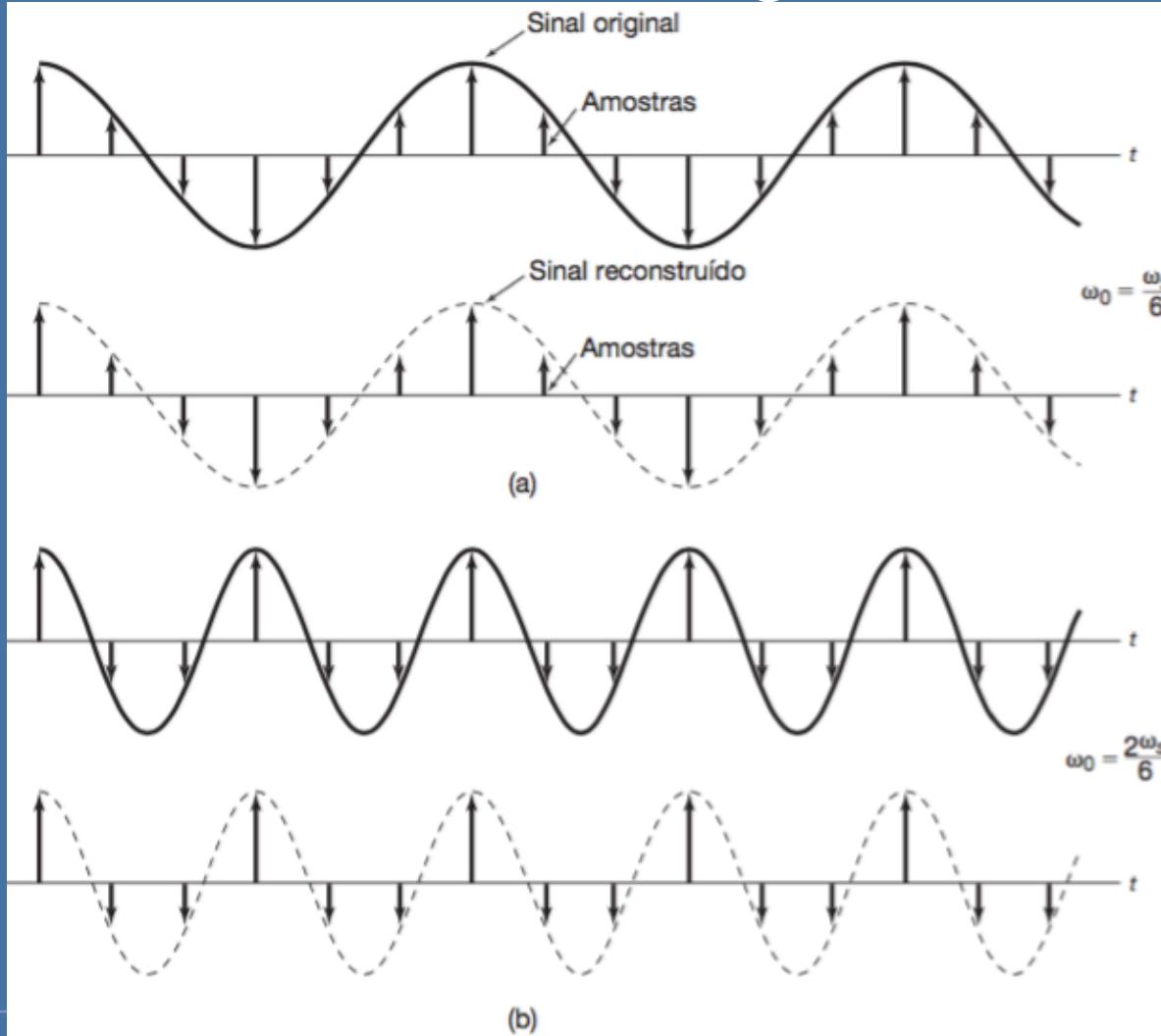
Amostragem

○ O efeito da subamostragem: aliasing



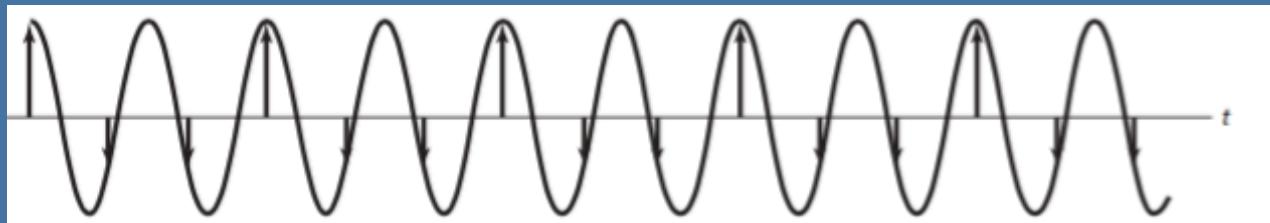
Amostragem

- O efeito da subamostragem: SEM aliasing.



Amostragem

○ O efeito da subamostragem: COM aliasing



(c)

$$\omega_0 = \frac{4\omega_s}{6}$$

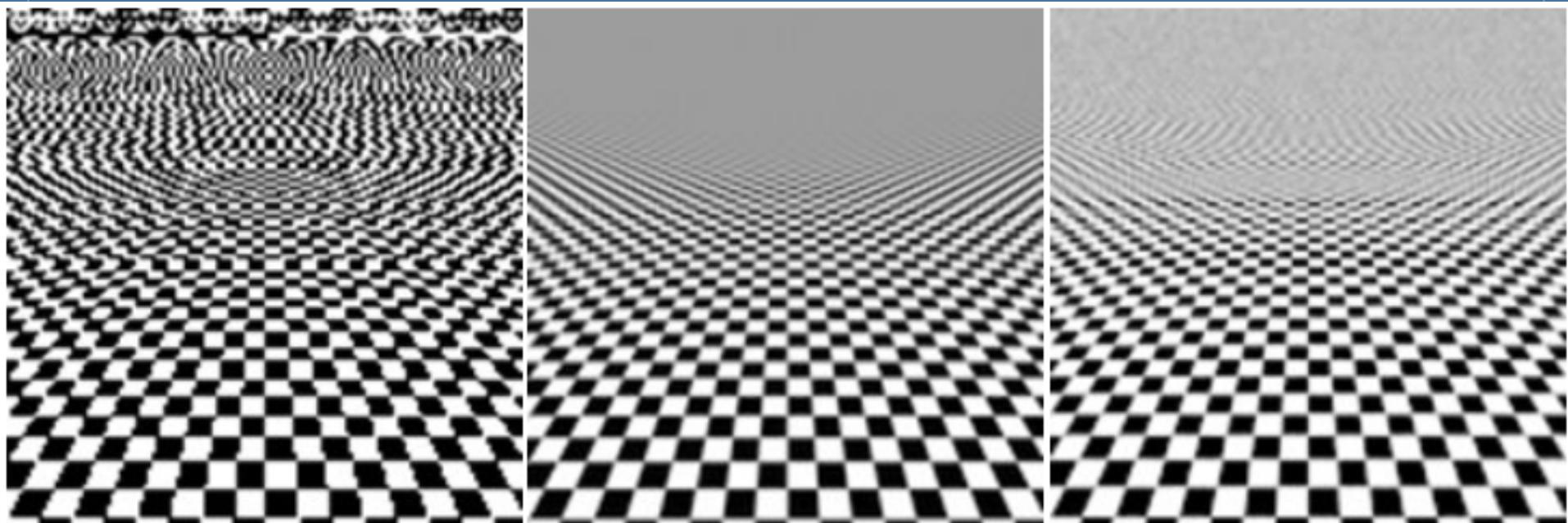


(d)

$$\omega_0 = \frac{5\omega_s}{6}$$

Amostragem

- Aliasing em imagens (filtros anti-aliasing):



Quantização em imagens



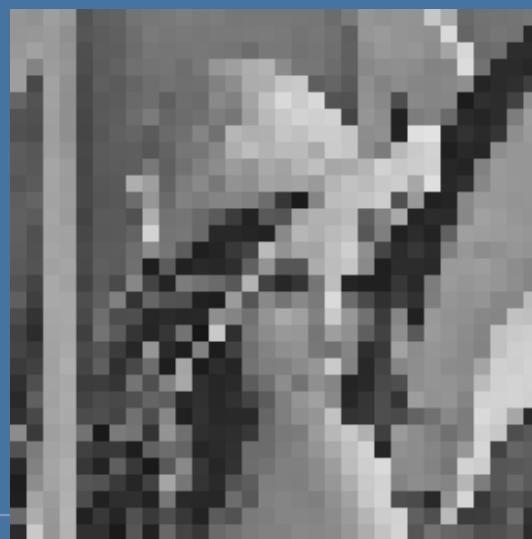
256x256 / 256 níveis



256x256 / 16 níveis



256x256 / 2 níveis



32x32 / 256 níveis

Quantização Vetorial (QV)

- Complexidade computacional extremamente elevada para compressão
- Complexidade computacional extremamente reduzida para descompressão
- Shannon, 1948: codificar elementos agrupados em vetores é mais eficiente (do ponto de vista de compressão) que codificá-los isoladamente.

Quantização Vetorial (QV)

- Mensagens são particionadas em blocos ou vetores com L elementos consecutivos
- K vetores (*vetores-código*) são selecionados, dentre aqueles que podem ser gerados pela fonte, e armazenados em uma tabela (*codebook*) disponível para o compressor e para o descompressor
- Espera-se que os vetores-código representem adequadamente a fonte

Quantização Vetorial (QV)

- Ex: fonte octal, $L = 3$, mensagem W

$$W = \left| \begin{array}{c|c|c|c|c|c|c|c|c} 7 & 6 & 5 & 4 & 4 & 4 & 3 & 4 & 3 \\ \hline \mathbf{v}_0 & \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 & \dots \end{array} \right| \dots$$

Quantização Vetorial (QV)

- Alfabeto original com cardinalidade M :
 $V = M^L$ vetores diferentes
- $M = 8$ e $L = 3 \Rightarrow V = 8^3 = 512$
- $M = 16$ e $L = 4 \Rightarrow V = 64K$
- $M = 256$ e $L = 4 \Rightarrow V = 4G$

Quantização Vetorial (QV)

- \mathbf{c}_k , $k = 0, 1, \dots, K-1$: k -ésimo vetor-código do *codebook*
- c_{kj} , $j = 0, 1, \dots, L-1$: j -ésimo elemento de \mathbf{c}_k
- \mathbf{v}_i , $i = 0, 1, \dots$: i -ésimo vetor na mensagem produzida pela fonte
- v_{ij} , $j = 0, 1, \dots, L-1$: j -ésimo elemento de \mathbf{v}_i .

Quantização Vetorial (QV)

- Um codebook exemplo

k	c_{k0}	c_{k1}	...	$c_{k,L-1}$
0	100	110	...	99
1	45	225	...	230
...
$K-1$	2	0	...	3

Quantização Vetorial (QV)

- Ex: fonte octal, L=3, mensagem W, K=16

- Codebook exemplo:

k	Vetor \mathbf{c}_k		
0	0	1	0
1	7	6	6
2	3	0	5
...
14	3	3	3
15	1	1	2

Quantização Vetorial (QV)

- Para codificar \mathbf{v}_i , o QV calcula a distorção $D(\mathbf{v}_i, \mathbf{c}_k)$, $k = 0, 1, \dots, K-1$.
- Erro médio quadrático:

$$D(\mathbf{v}_i, \mathbf{c}_k) = \frac{1}{L} \sum_{j=0}^{L-1} (v_{ij} - c_{kj})^2$$

- Se $D(\mathbf{v}_i, \mathbf{c}_k)$ é mínima para $k = k'$, índice k' é anexado à mensagem comprimida.
- Na descompressão, k' é substituído pelo vetor-código $\mathbf{c}_{k'}$

Quantização Vetorial (QV)

- Ex: fonte octal, $L = 3$, mensagem W

$W = 7 \ 6 \ 5 \ 4 \ 4 \ 4 \ 3 \ 4 \ 3 \ 2 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 2 \ 1 \ 0 \ \dots$

k Vetor \mathbf{c}_k

0	0	1	0
1	7	6	6
2	3	0	5
...
14	3	3	3
15	1	1	2

$$D(\mathbf{v}_0, \mathbf{c}_0) = (7-0)^2 + (6-1)^2 + (5-0)^2 = 99$$

$$D(\mathbf{v}_0, \mathbf{c}_1) = (7-7)^2 + (6-6)^2 + (5-6)^2 = 1$$

...

$$D(\mathbf{v}_0, \mathbf{c}_{15}) = (7-1)^2 + (6-1)^2 + (5-2)^2 = 70$$

Se $D(\mathbf{v}_0, \mathbf{c}_1)$ é o menor valor para \mathbf{v}_0 , este será codificado como $k = 1$, com $\log_2 16 = 4$ bits -> $RC = 9/4$.

Quantização Vetorial (QV)

- Ex: fonte octal, L = 3, mensagem W

$$W_o = \begin{array}{c|c|c|c|c|c|c|c|c} \mathbf{v}_0 & \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 & \dots \\ \hline 7 & 6 & 5 & 4 & 4 & 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 2 & 1 & 0 & \dots \end{array}$$
$$W_c = \begin{array}{c|c|c|c|c|c|c|c|c} & 1 & 14 & 14 & 0 & 0 & 15 & 0 \\ \hline & 1 & 14 & 14 & 0 & 0 & 15 & 0 \end{array}$$
$$W_d = \begin{array}{c|c|c|c|c|c|c|c|c} & 7 & 6 & 6 & 3 & 3 & 3 & 3 & 3 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 0 & 1 & 0 & \dots \\ \hline & 7 & 6 & 6 & 3 & 3 & 3 & 3 & 3 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 0 & 1 & 0 & \dots \end{array}$$

Quantização Vetorial (QV)

- Se os índices são codificados com $\log_2(K)$ bits, a taxa resultante é de $\log_2(K)/L$ bits/elemento.
- Alfabeto original com cardinalidade M:
- $RC = \log_2(M^L)/\log_2(K)$
- Ex: $M = 256$, $L = 4$, $K = 128$
- $RC = \log_2(256^4)/\log_2(128) = 4,57$

Quantização Vetorial (QV)

- Teoricamente, quanto maior L , mais eficiente a quantização.
- Dimensão dos vetores é limitada na prática.
- Desempenho depende ainda do tamanho do *codebook*, da escolha dos vetores-código e da medida de distorção.
- Como selecionar vetores-código?

Quantização Vetorial (QV)

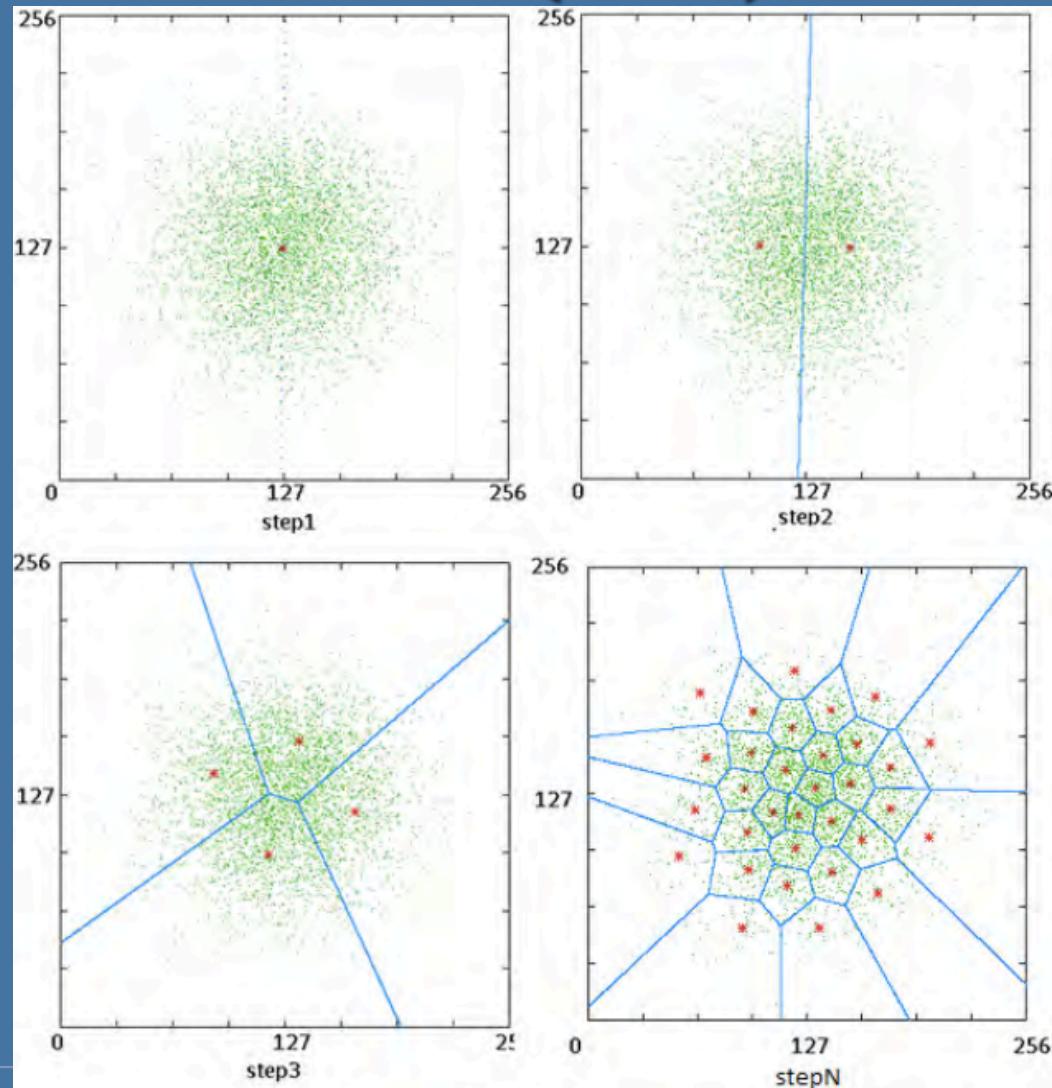
- O *codebook* é gerado por *treinamento*.
- No treinamento, uma quantidade elevada de mensagens é examinada.
- Vetores mais representativos são selecionados como vetores-código.
- Algoritmo LBG: método de treinamento popular
- Uma grande variedade de técnicas tem sido apresentada, incluindo redes neurais
- *Codebooks* estáticos x dinâmicos

Quantização Vetorial (QV)

- Algoritmo LBG.
 - Espaço dos possíveis vetores que aconteceram nas mensagens de treino;
 - Ordena em relação ao centro;
 - Divide espaço em 2;
 - Calcula novos centros.

Quantização Vetorial (QV)

- Algoritmo LBG.
 - Vetor $L=2$ (2D)
 - Cada ponto é uma msg de treino.
 - Linhas azuis Ilustrativas
 - [link](#)



Paradigma Transformada, Quantização, Codificação (TQC)

- Transformadas: Cosseno Discreta (DCT) e wavelets, principalmente
- Quantização: normalmente, escalar
- Comparado com QV, TQC apresenta baixo custo computacional e compromisso taxa-distorção semelhante
- Utilizado nos padrões JPEG, JPEG2000, MPEG2, MPEG4, MPEG7, MP3...

JPEG - Transformada

- $\text{Banda}_X = \text{Banda}_X - 128$
- Aplicação da DCT bidimensional a blocos (8 x 8 pixels) adjacentes da Banda_X.

JPEG - Quantização

- Matriz de quantização Q , 8×8
- Q_{ij} é o tamanho do passo de quantização BlocoDCT_{ij}
- Quantização:
 $\text{round}(\text{BlocoDCT}_{ij} / Q_{ij})$

JPEG - Quantização

- O comitê JPEG deixou livre a escolha de Q, mas divulgou matrizes de quantização, classificadas por taxa de compressão e qualidade, obtidas por intermédio de experimentos psicofisiológicos.

JPEG - Quantização

- ◎ Matriz de quantização sugerida:

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	162
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG – Exemplo Quantização/ Dequantização

◎ Bloco da Banda X

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

JPEG – Exemplo Quantização/ Dequantização

○ DCT de (BandaX-128)

235,6	-1,0	-12,1	-5,2	2,1	-1,7	-2,7	1,3
-22,6	-17,5	-6,2	-3,2	-2,9	-0,1	0,4	-1,2
-10,9	-9,3	-1,6	1,5	0,2	-0,9	-0,6	-0,1
-7,1	-1,9	0,2	1,5	0,9	-0,1	0,0	0,3
-0,6	-0,8	1,5	1,6	-0,1	-0,7	0,6	1,3
1,8	-0,2	1,6	-0,3	-0,8	1,5	1,0	-1,0
-1,3	-0,4	-0,3	-1,5	-0,5	1,7	1,1	-0,8
-2,6	1,6	-3,8	-1,8	1,9	1,2	-0,6	-0,4

JPEG – Exemplo Quantização/ Dequantização

◦ Q

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	162
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG – Exemplo Quantização/ Dequantização

- ## ● Bloco DCT quantizado por Q

JPEG – Exemplo Quantização/ Dequantização

● Bloco DCT dequantizado

JPEG – Exemplo Quantização/ Dequantização

- ◎ Aproximação para bloco original

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

JPEG – Exemplo Quantização/ Dequantização

○ Comparação

Original

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

Aproximação

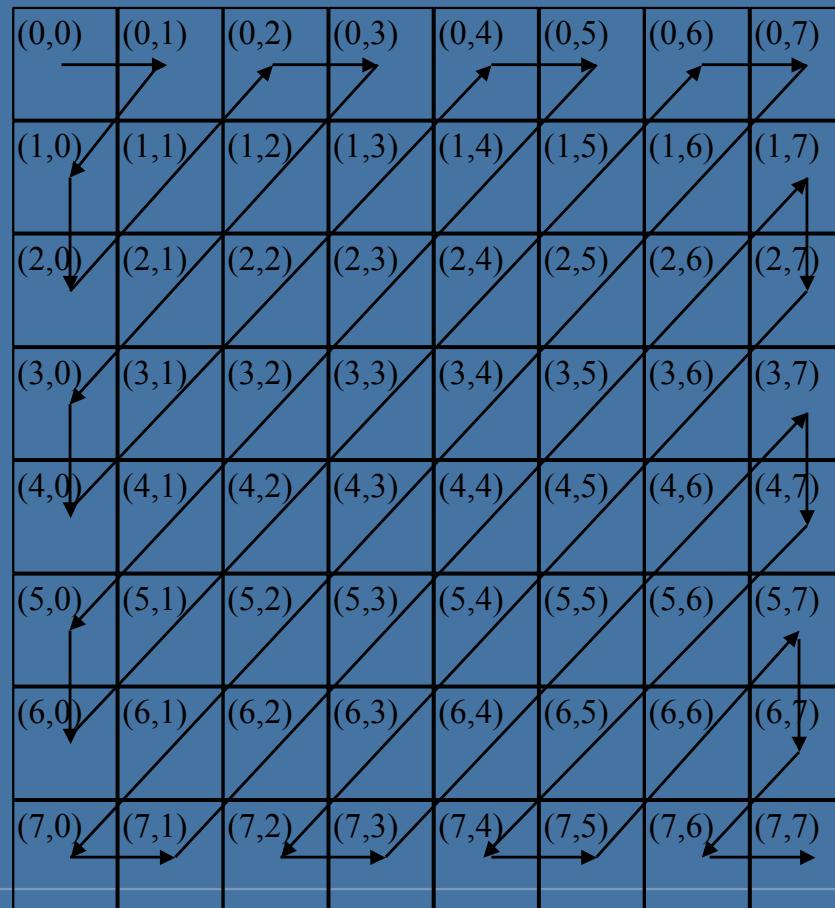
144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

JPEG – Codificação de Entropia

- Substituição do coeficiente DC de cada bloco pela diferença entre o próprio coeficiente e o coeficiente DC do bloco antecedente.
- RLEZ dos coeficientes de cada bloco empregando percurso em zig-zag.
- Codificação de Huffman.

JPEG – Codificação de Entropia

○ Percurso em zig-zag



JPEG – Codificação de Entropia

- Coeficiente AC não nulo de valor a é representado em combinação com o comprimento c da seqüência de coeficientes AC nulos que o precedeu no percurso zig-zag.
- Cada par (c, a) é usualmente representado por um par de símbolos $s_1 = (c, b)$ e $s_2 = a$
- b : nº de bits usados para codificar a .

JPEG – Codificação de Entropia

- $s_1 = (15, 0)$: seqüência de 16 zeros, obrigatoriamente seguido por outro símbolo s_1 .
- $s_1 = (0,0)$: seqüência de zeros de qualquer tamanho que se estende até o fim do bloco.
- Termo DC: $s_1 = b$ e $s_2 = a$ (valor do coef DC diferencial).

JPEG – Codificação de Entropia

- Relação entre b e a , para coefs AC:

b	a
1	-1, 1
2	-3, -2, 2, 3
3	-7, ..., -4, 4, ..., 7
4	-15, ..., -8, 8, ... 15
5	-31, ..., -16, 16, ..., 31
6	-63, ..., -32, 32, ..., 63
7	-127, ..., -64, 64, ..., 127
8	-255, ..., -128, 128, ..., 255
9	-511, ..., -256, 256, ..., 511
10	-1023, ..., -512, 512, ..., 1023
11	-2047, ..., -1024, 1024, 2047

JPEG – Codificação de Entropia

- s_1 codificado por Huffman ou codificação aritmética, e s_2 por código fixo:

a	Código
-1	0
1	1
-3	00
-2	01
2	10
3	11
...	...

JPEG – Exemplo de Codificação de Entropia

- ## ● Bloco DCT quantizado:

JPEG – Exemplo de Codificação de Entropia

- ◎ Dados:

- Comp. DC do bloco anterior: 12
- Cód. Huffman de s_1 (DC) de valor 2: 011
- Cód. Huffman para símbolos s_1 (AC):

s_1	Código
(0,0)	1010
(0,1)	00
(1,2)	11011
(2,1)	11100

JPEG – Exemplo de Codificação de Entropia

- Coeficiente DC do bloco atual será substituído por $15-12 = 3$.
- $a = 3$ será representado por $b = 2$ bits.
- $(s_1, s_2) = (2, 3)$

JPEG – Exemplo de Codificação de Entropia

- No percurso zig-zag, o coef. AC quantizado de valor -2 é precedido por um zero: $(s_1, s_2) = [(1, 2), -2]$ Três coef. de valor -1 adjacentes: três pares $(s_1, s_2) = [(0, 1), -1]$
- Dois zeros seguido por outro coeficiente de valor -1: $(s_1, s_2) = [(2, 1), -1]$.
- Zeros até o fim do bloco: $s_1 = (0,0)$

JPEG – Exemplo de Codificação de Entropia

- Das tabelas com o código fixo para s_2 e Huffman para s_1 , o bloco DCT codificado será:
- 011 11 11011 01 00 0 00 0 00 0 11100 0 1010
- Se as imagem original usava 8 bits por pixel, RC resultante $64*8/31 = 16,5$.