



UNIVERSIDADE FEDERAL DE PELOTAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO
TEC2/TEC4: REDES MULTIMÍDIA (RMM)

Unidade 2

Conceitos Básicos de Codificação Digital

Prof. Guilherme Corrêa
gcorrea@inf.ufpel.edu.br

Sumário

❖ Digitalização

- Amostragem
- Quantização
- Codificação

❖ Codificação de Fonte

❖ Codificação Entrópica

- Codificação de *Huffman*
- *Run-Length Encoding*

Sumário

❖ Digitalização

- Amostragem
- Quantização
- Codificação

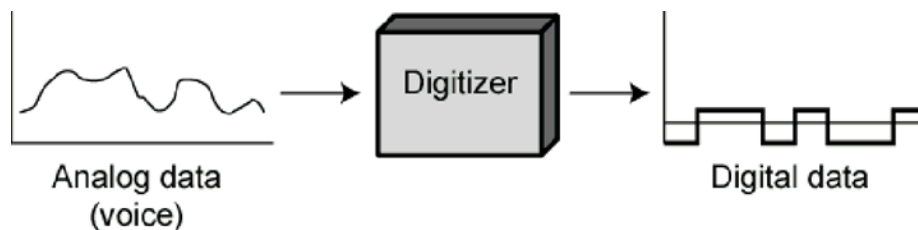
❖ Codificação de Fonte

❖ Codificação Entrópica

- Codificação de *Huffman*
- *Run-Length Encoding*

Digitalização

- ❖ em alguns casos, queremos converter um sinal analógico para digital
 - ex.: voz (analógico) tratada (digitalmente) e transmitida como sinal analógico



- ❖ codec (*coder-decoder*):
 - converte dados analógicos para digitais
 - recupera os dados analógicos a partir dos digitais

Digitalização

Amostragem

- ❖ lida com a dimensão temporal do sinal, tornando-o discreto no tempo

Quantização

- ❖ lida com a amplitude do sinal, tornando-a discreta

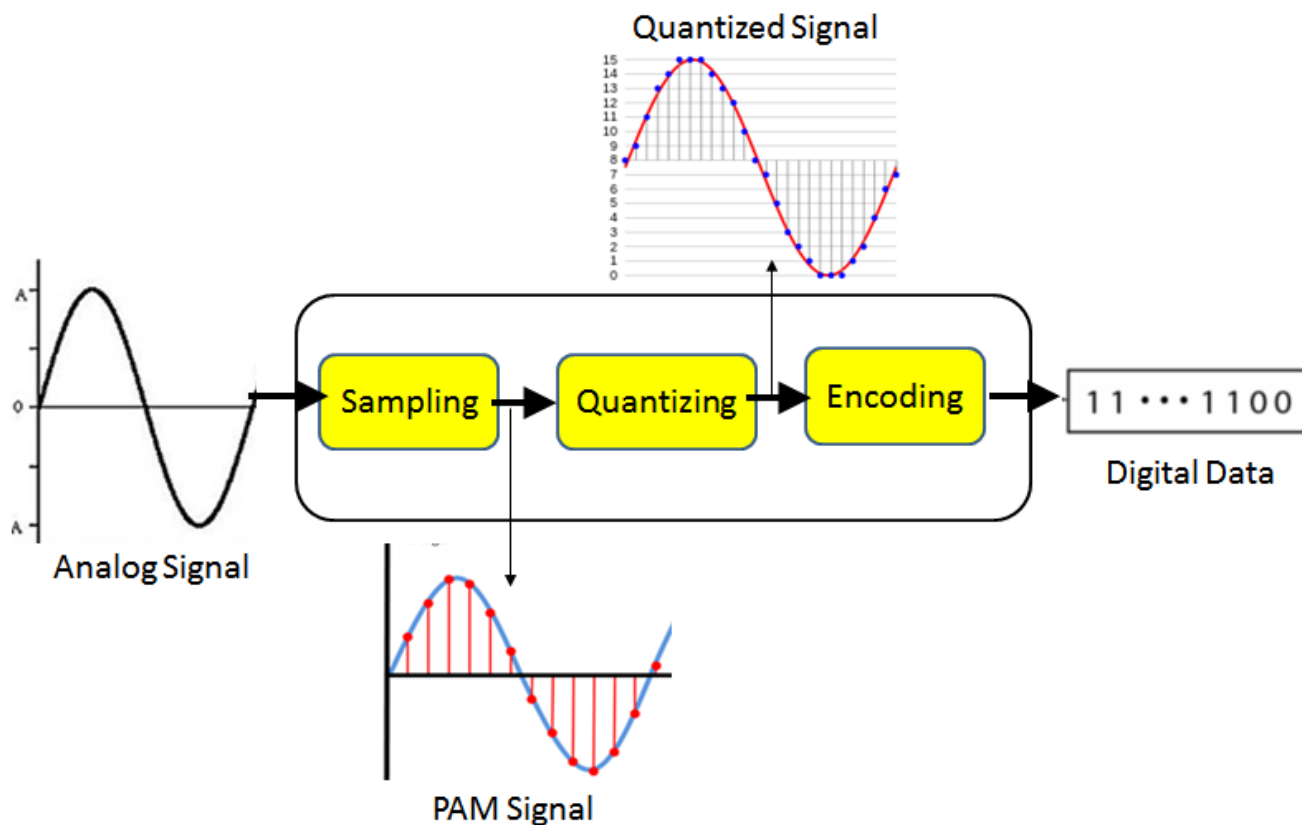
Codificação

- ❖ converte o resultado da quantização em um código binário designado a cada nível

Digitalização

Amostragem

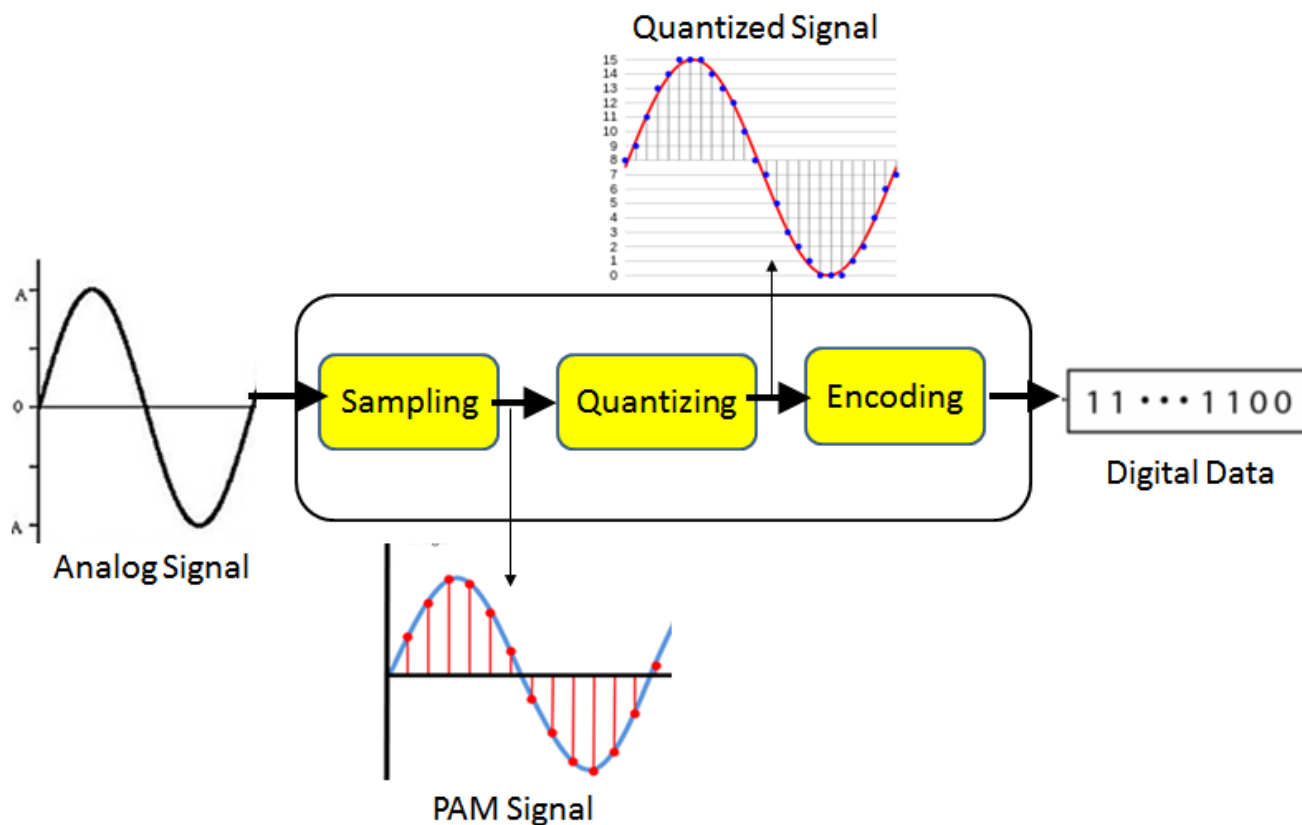
- ❖ *entrada*: sinal analógico (contínuo em tempo, contínuo em amplitude)
- ❖ *saída*: sinal analógico (discreto em tempo, contínuo em amplitude)



Digitalização

Quantização

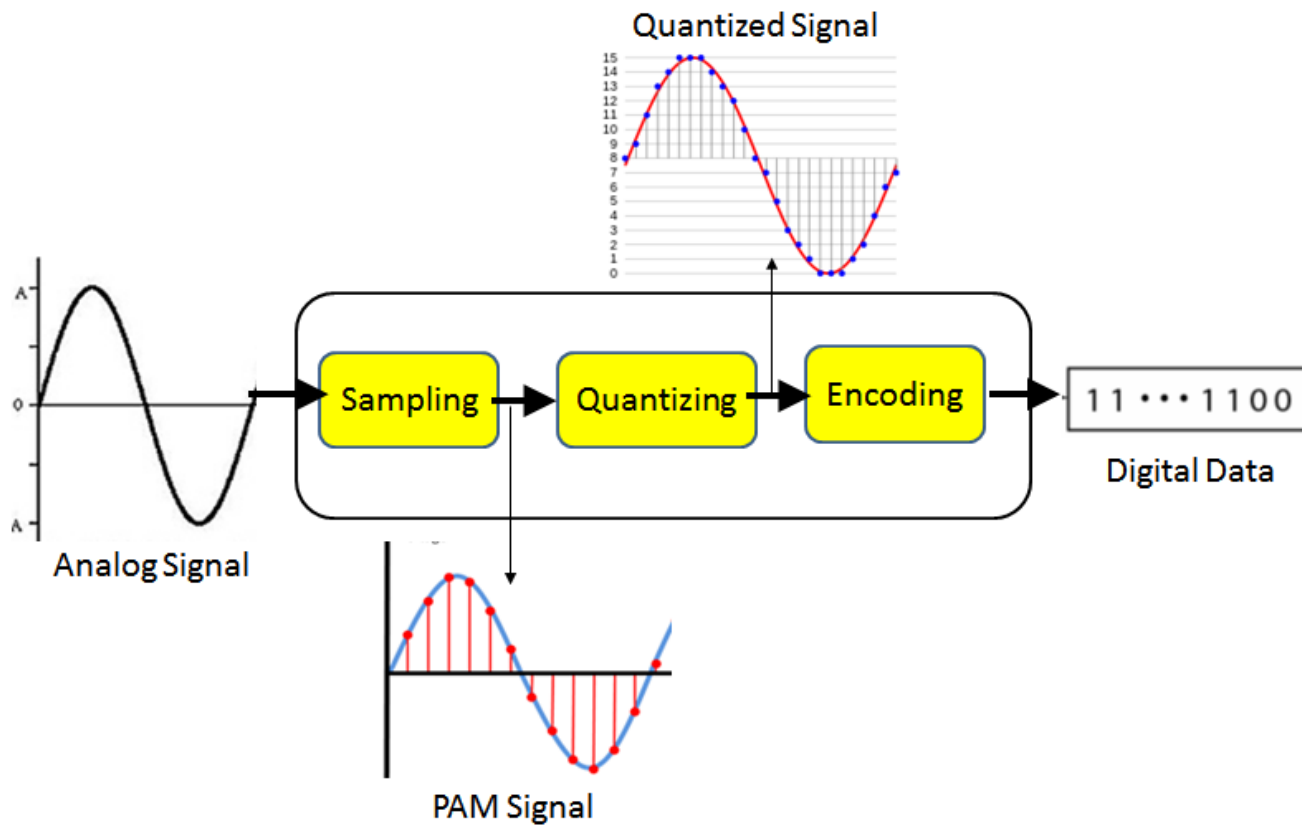
- ❖ *entrada*: sinal analógico (discreto em tempo, contínuo em amplitude)
- ❖ *saída*: sinal digital (discreto em tempo, discreto em amplitude)



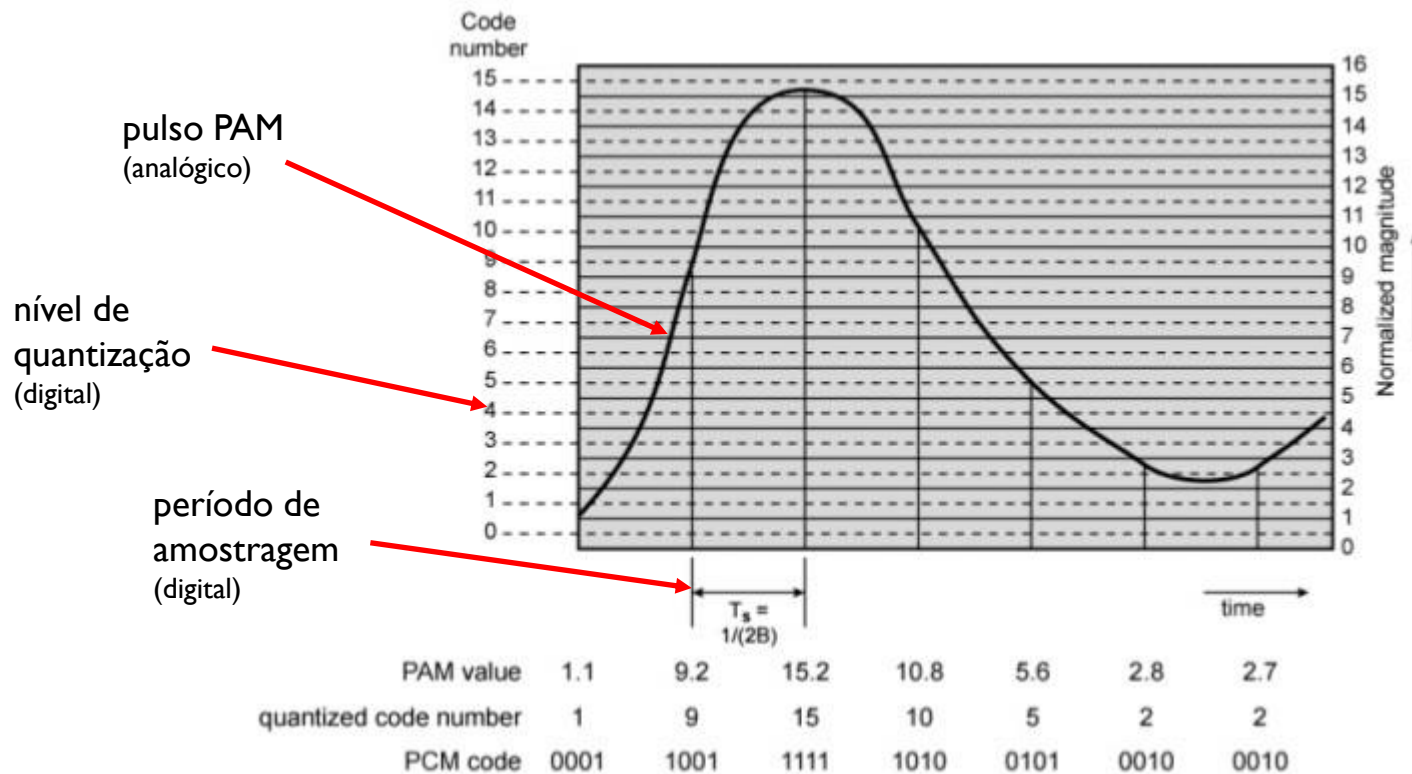
Digitalização

Codificação

- ❖ *entrada*: sinal digital (discreto em tempo, discreto em amplitude)
- ❖ *saída*: código binário

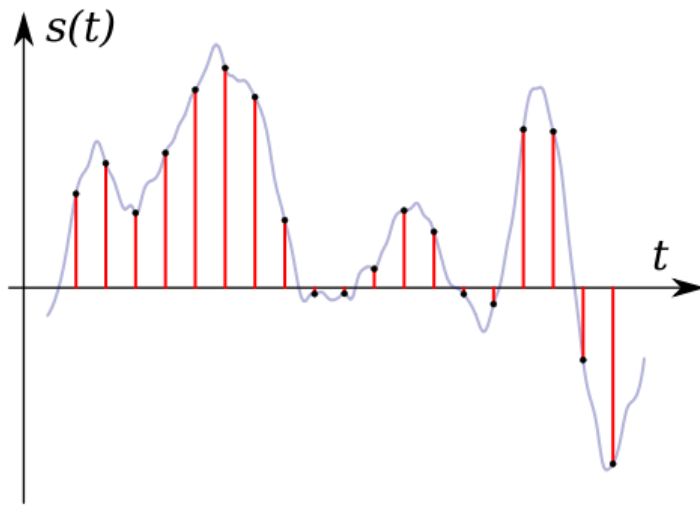


Digitalização

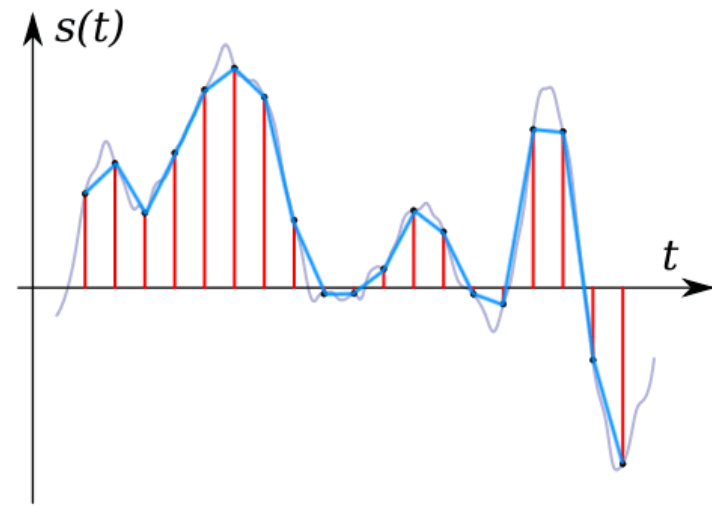


Digitalização

Erro de Subamostragem



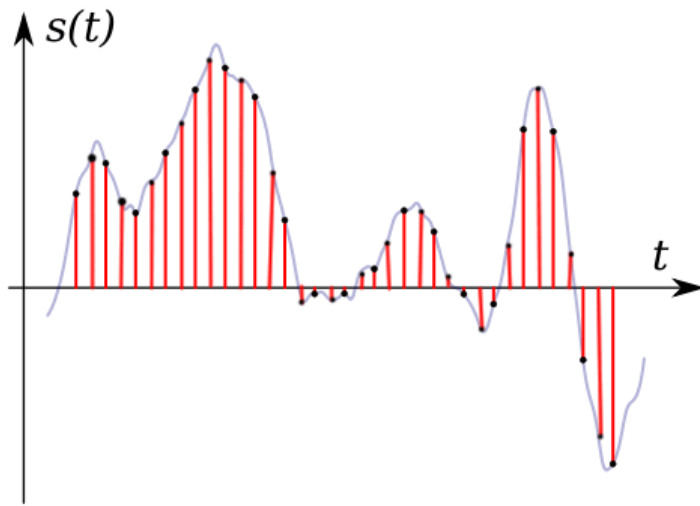
N amostras/segundo



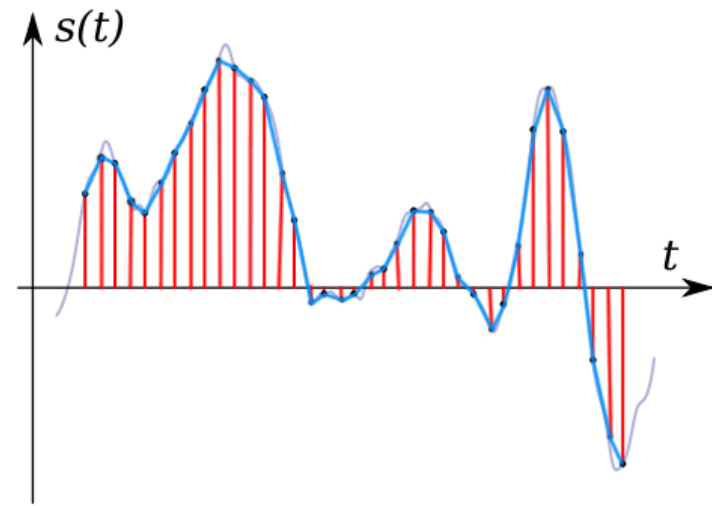
N amostras/segundo

Digitalização

Erro de Subamostragem



$2N$ amostras/segundo



$2N$ amostras/segundo

Digitalização

Teorema de Nyquist

“Um sinal analógico pode ser completamente reconstruído a partir de uma sequência periódica de amostras, desde que a frequência de amostragem (f_a) seja maior ou igual ao dobro da frequência máxima presente no sinal a amostrar (f_{max})”

$$f_a \geq 2 * f_{max}$$

Digitalização

Série de Fourier

Sinal
original



Fundamental
frequency



+ 0.5 \times
2 \times fundamental



=



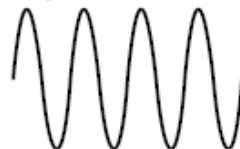
+ 0.33 \times
3 \times fundamental



=



+ 0.25 \times
4 \times fundamental



=



+ 0.5 \times
5 \times fundamental

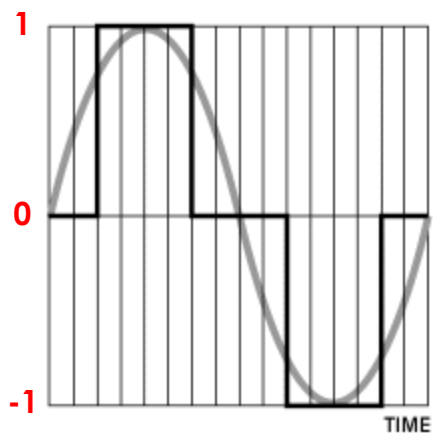


=

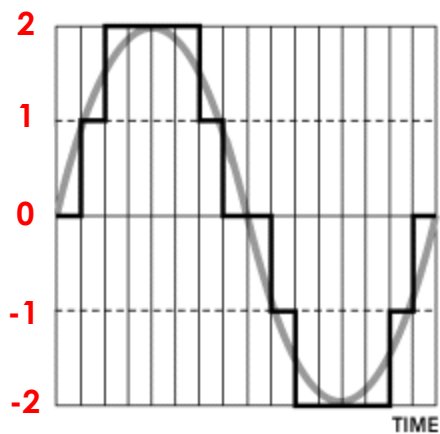


Digitalização

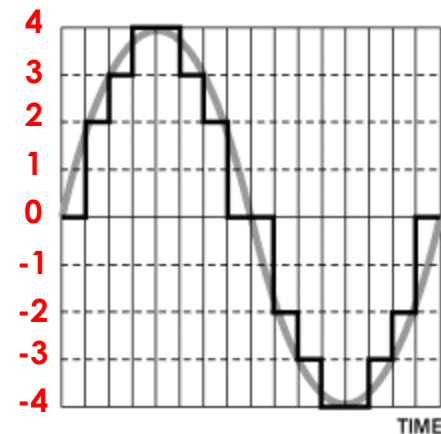
Erro de Quantização



3 níveis: 2 bits/amostra



5 níveis: 3 bits/amostra



9 níveis: 5 bits/amostra

Sumário

❖ Digitalização

- Amostragem
- Quantização
- Codificação

❖ **Codificação de Fonte**

❖ Codificação Entrópica

- Codificação de *Huffman*
- *Run-Length Encoding*

Codificação de Fonte

- ❖ PCM é a forma mais simples de codificação
 - cada amostra é representada independentemente das restantes
- ❖ mesmo número de bits por amostra
 - representação pouco eficiente
- ❖ **Precisamos de técnicas de compressão!**
 - reduzir espaço de armazenamento
 - reduzir requisitos de comunicação
- ❖ **Codificação de Fonte**
 - informação digital representada no formato PCM é codificada numa forma mais eficiente
- ❖ **Decodificação**
 - recuperação do sinal original (ou estimativa) a partir do sinal codificado

Codificação de Fonte

Classificação (I)

❖ **Sem perdas**

- sem perda de informação
- dados descomprimidos são idênticos aos dados originais
- essencial em arquivos de texto, arquivos binários executáveis, etc.

❖ **Com perdas**

- aproximação dos dados originais
- taxa de compressão melhor que na codificação sem perdas
- compromisso entre taxa de compressão e fidelidade

Codificação de Fonte

Classificação (2)

❖ Codificação Entrópica

- sem perdas
- explora a *redundância estatística* do sinal
- não considera características específicas da mídia
- dados tomados como simples sequências digitais
- atribui palavras de código mais curtas a símbolos que ocorrer mais frequentemente

“o rato roeu a roupa do rei de roma”

o: 6 vezes	u: 2 vezes
r: 5 vezes	d: 2 vezes
a: 4 vezes	t: 1 vez
e: 3 vezes	p: 1 vez
	m: 1 vez

Classificação (2)

❖ Codificação Preditiva

- explora a *redundância temporal* e *redundância temporal* dos sinais
 - isto é, explora a existência de amostras obtidas em instantes/zonas diferentes que são iguais ou parecidas

❖ Codificação por Transformada

- no domínio da transformada, identifica e explora *redundância espectral*

Codificação de Fonte

Redundância Perceptual ou Irrelevância

- ❖ as classes de codificações mencionadas até aqui não causam perdas
- ❖ entretanto, alguns aspectos do sinal podem não ser essenciais para a sua correta percepção
 - exemplo: áudio acima dos 20 kHz
- ❖ a codificação de fonte pode eliminar este tipo de informação
- ❖ o dado decodificado é diferente do dado original



Sumário

❖ Digitalização

- Amostragem
- Quantização
- Codificação

❖ Codificação de Fonte

❖ **Codificação de Entropia**

- Codificação de *Huffman*
- *Run-Length Encoding*

Codificação de Entropia

- ❖ Mede o “grau de desordem” ou
mede o “grau de informação” de um sinal

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

Codificação de Entropia

- ❖ Mede o “grau de desordem” ou
mede o “grau de informação” de um sinal

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

$$H(X) = \sum_{i=1}^M p_i \cdot I(s_i)$$

Codificação de Entropia

- ❖ Mede o “grau de desordem” ou
mede o “grau de informação” de um sinal

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

$$H(X) = \sum_{i=1}^M p_i \cdot I(s_i) = \sum_{i=1}^M p_i \cdot \log_2 \left(\frac{1}{p_i} \right)$$

Codificação de Entropia

- ❖ Mede o “grau de desordem” ou
mede o “grau de informação” de um sinal

$$I(s_k) = \log_2 \left(\frac{1}{p_k} \right)$$

$$H(X) = \sum_{i=1}^M p_i \cdot I(s_i) = \sum_{i=1}^M p_i \cdot \log_2 \left(\frac{1}{p_i} \right) = - \sum_{i=1}^M p_i \cdot \log_2(p_i) \quad \text{bit/símbolo}$$

Codificação de Entropia

- ❖ A compressão máxima teórica é dada por:

$$MaxComp = \frac{\text{\#bits por amostra}}{\text{entropia}}$$

- ❖ Tipicamente o # bits por amostra é oito (uma amostra de áudio, um pixel, um caractere)
- ❖ Logo, quanto maior a entropia, menor a taxa de compressão

Codificação de Entropia

- ❖ A codificação de entropia pode utilizar diversos algoritmos:
 - ❖ **Huffman**
 - ❖ Exp-Golomb
 - ❖ *Run-Length Encoding*
 - ❖ CAVLC (Context-Based Adaptive Variable Length Coding)
 - ❖ CABAC (Context-Based Adaptive Binary Arithmetic Coding)

Codificação de Huffman

- ❖ Utiliza a **probabilidade de ocorrência** dos símbolos para determinar a sua representação
 - ❖ Princípio de Morse

Código Morse internacional			
a	●-	n	-●
b	-●●●	o	---
c	-●-●	p	●--●
d	-●●	q	--●-
e	●	r	●-●
f	●●-●	s	●●●
g	--●	t	-
h	●●●●	u	●●-
i	●●	v	●●●-
j	●---	w	●--
k	-●-	x	-●●-
l	●-●●	y	-●--
m	--	z	--●●
á	●--●-	8	---●●
ä	●-●-	9	----●
é	●●-●●	0	-----
ñ	--●--	,	--●●-- (vírgula)
ö	---●	.	●-●-●-
ü	●●--	?	●●--●●
1	●----	;	-●-●-●
2	●●----	:	---●●●
3	●●●--	'	●-----● (apóstrofo)
4	●●●●-	-	-●●●●- (hífen)
5	●●●●●	/	-●●-●
6	-●●●●	(-●-●-● (parêntese esquerdo)
7	--●●●)	-●-●-● (parêntese direito)

Codificação de Huffman

- ❖ Utiliza a **probabilidade de ocorrência** dos símbolos para determinar a sua representação
 - ❖ Princípio de Morse
- ❖ No nosso contexto, o que é um **símbolo**?
 - ❖ uma letra, um valor de pixel, uma amostra de som
- ❖ Monta uma árvore binária (**Árvore de Huffman**) com a junção de pares de probabilidade, onde as folhas apresentam as menores probabilidades

Codificação de Huffman

- ❖ Garante a geração de **códigos únicos, não ambíguos**
- ❖ É um **método livre** para uso (sua patente já expirou), portanto, pode ser usado para um compressor comercial

Codificação de Huffman

❖ *Codificação de Huffman* – **Algoritmo**

enquanto tamanho(alfabeto) > 1:

 S0 := retira_menor_probabilidade(alfabeto)

 S1 := retira_menor_probabilidade(alfabeto)

 X := novo_nó

 X.filho0 := S0

 X.filho1 := S1

 X.probabilidade := S0.probabilidade + S1.probabilidade

 insere(alfabeto, X)

fim enquanto

X = retira_menor_símbolo(alfabeto)

para cada folha em folhas(X):

 código[folha] := percorre_da_raiz_até_a_folha(folha)

fim para

Codificação de Huffman

Exemplo:

- ❖ Vamos considerar a codificação de uma sequência de 22 caracteres:

AAAAAAAAAABBBBCCCDDEEF

- ❖ Quantos símbolos existem?
- ❖ Quantos bits precisaremos utilizar para representar cada símbolo?

Codificação de Huffman

Exemplo:

- ❖ Vamos considerar a codificação de uma sequência de 22 caracteres:

AAAAAAAAAABBBBCCCDDEEF

- ❖ Códigos para cada símbolo:

A = 000; B = 001; C = 010; D = 011; E = 100; F = 101

- ❖ Código (sem compressão) para a representação da sequência:

000000000000000000000000000000001001001001010010010011011011100100101

Total = 66 bits

Codificação de Huffman

Exemplo:

- ❖ Vamos considerar a codificação de uma sequência de 22 caracteres:

AAAAAAAAAABBBBCCCDDEEF

- ❖ Contagem dos símbolos e cálculo das probabilidades

$A = 9$; $B = 4$; $C = 3$; $D = 3$; $E = 2$; $F = 1$

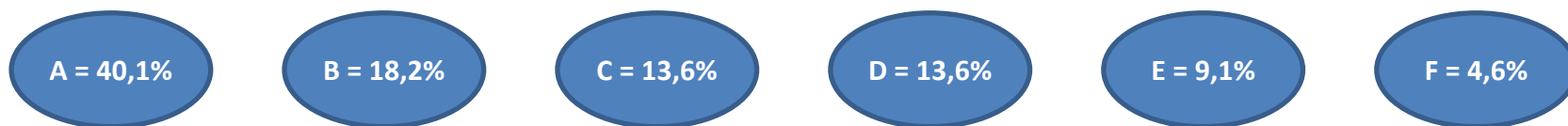
$A = 40,1\%$; $B = 18,2\%$; $C = 13,6\%$; $D = 13,6\%$; $E = 9,1\%$; $F = 4,6\%$

Codificação de Huffman

Exemplo:

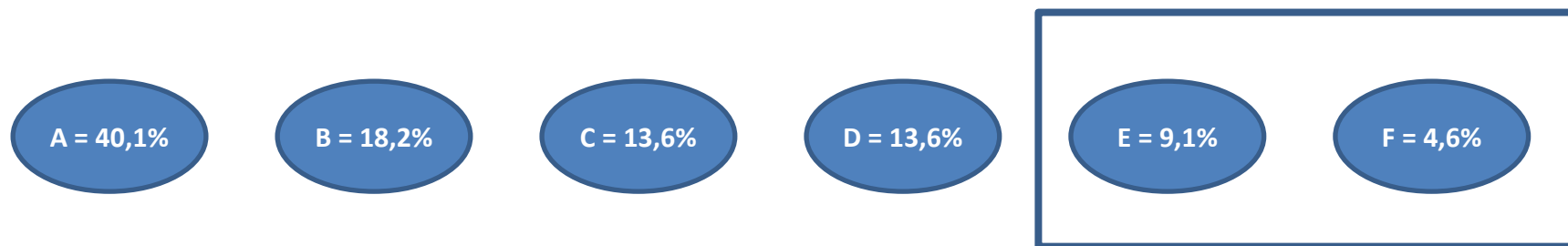
- ❖ A partir das probabilidades, vamos agrupar pares para formar a árvore, começando pelas menores ocorrências

A = 40,1%; B = 18,2%; C = 13,6%; D = 13,6%; E = 9,1%; F = 4,6%

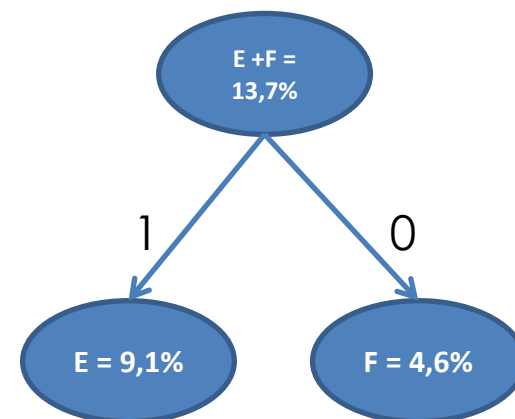


Codificação de Huffman

Exemplo:

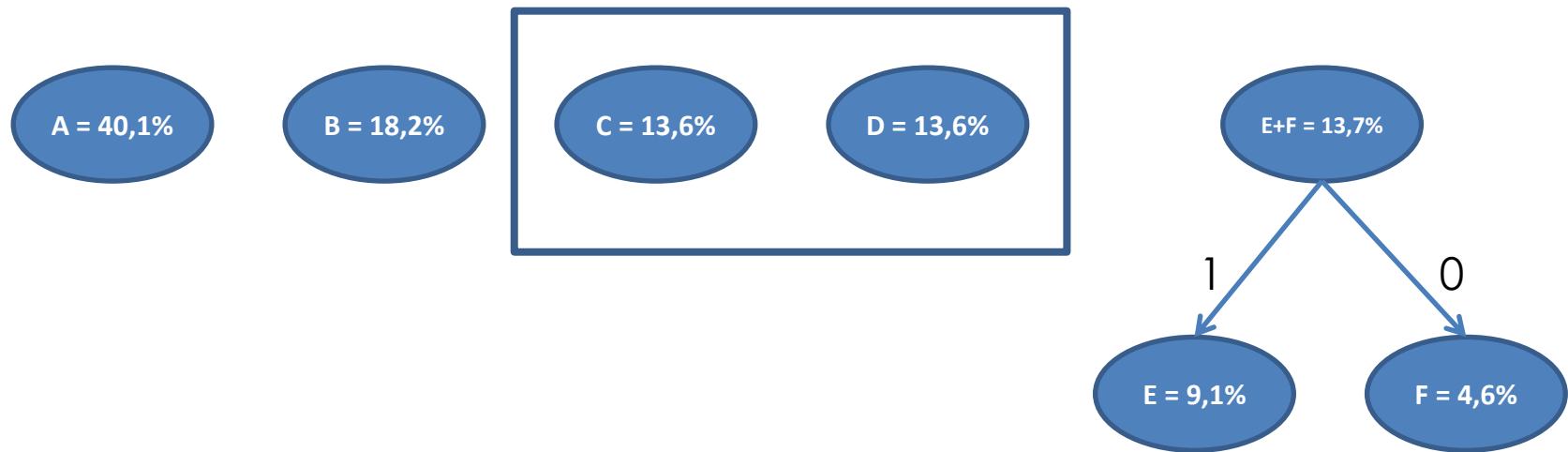


- ❖ As arestas da esquerda recebem '1', e as arestas da direita '0'



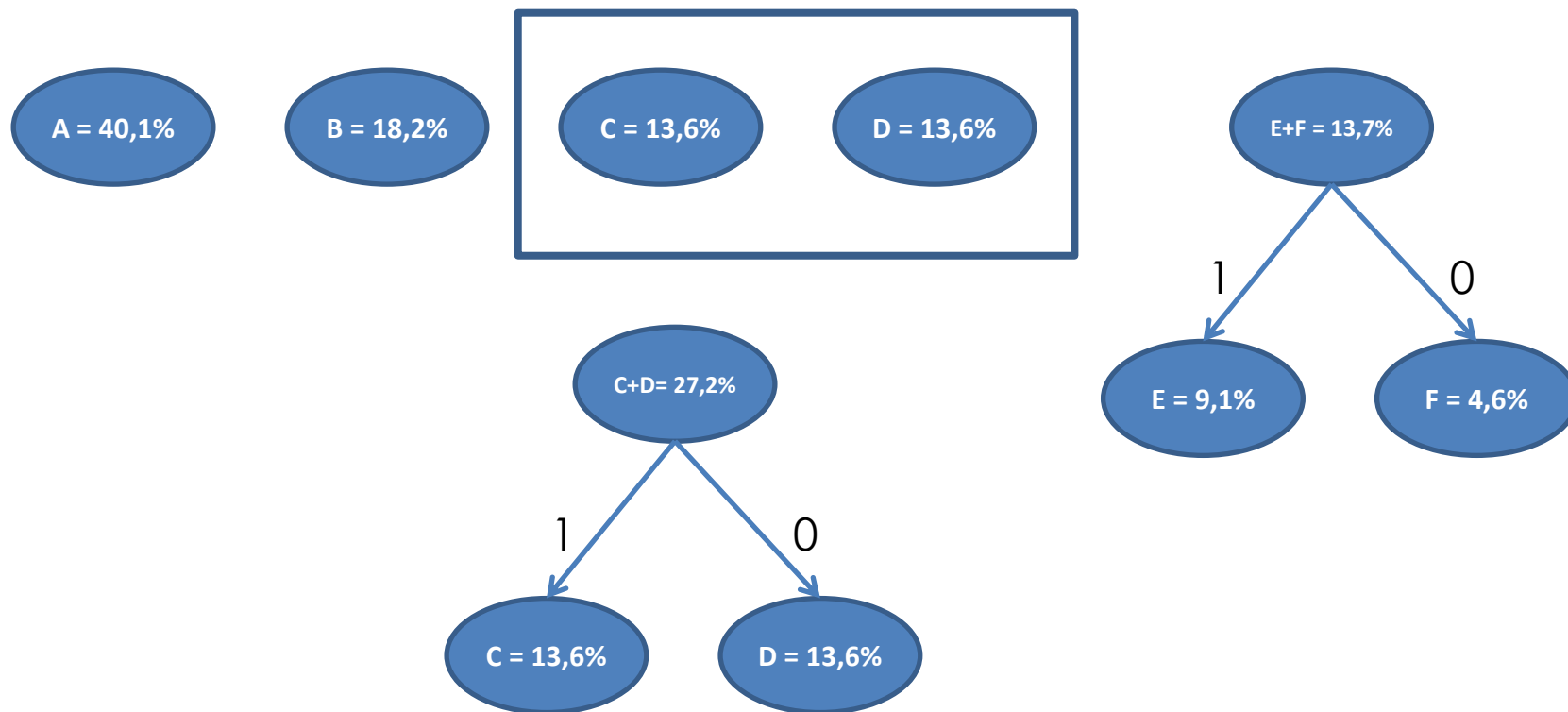
Codificação de Huffman

Exemplo:



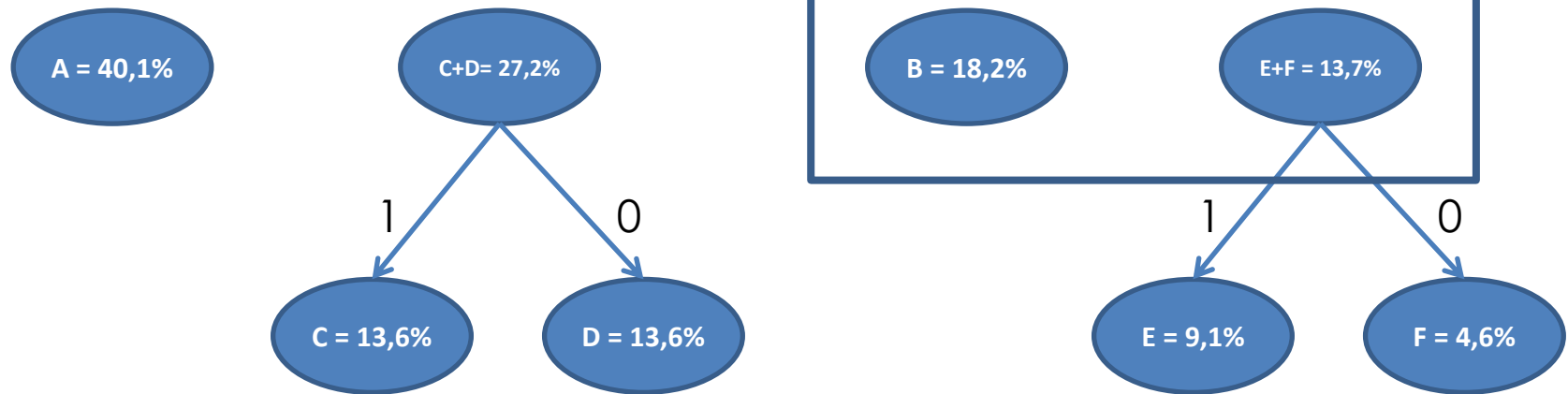
Codificação de Huffman

Exemplo:



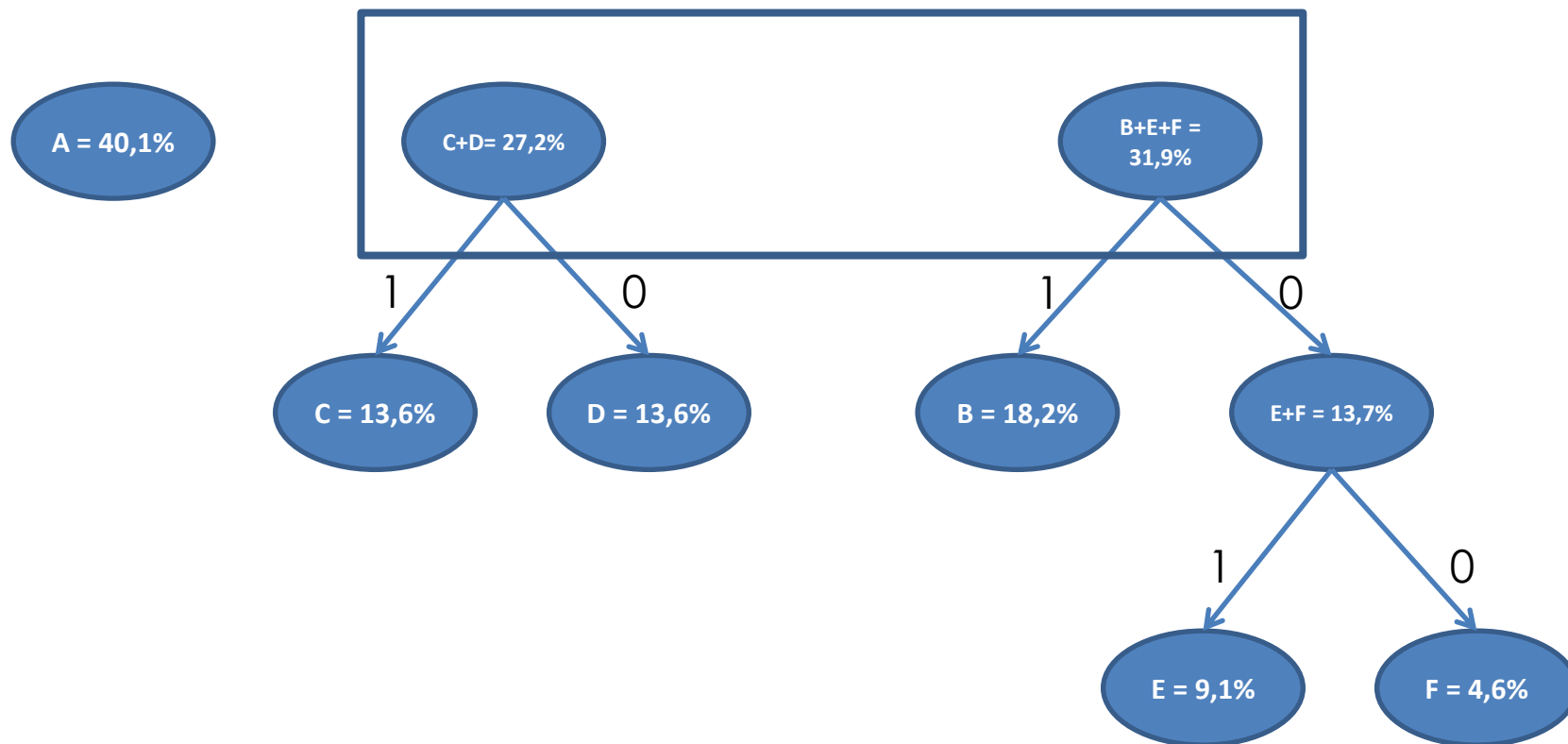
Codificação de Huffman

Exemplo:



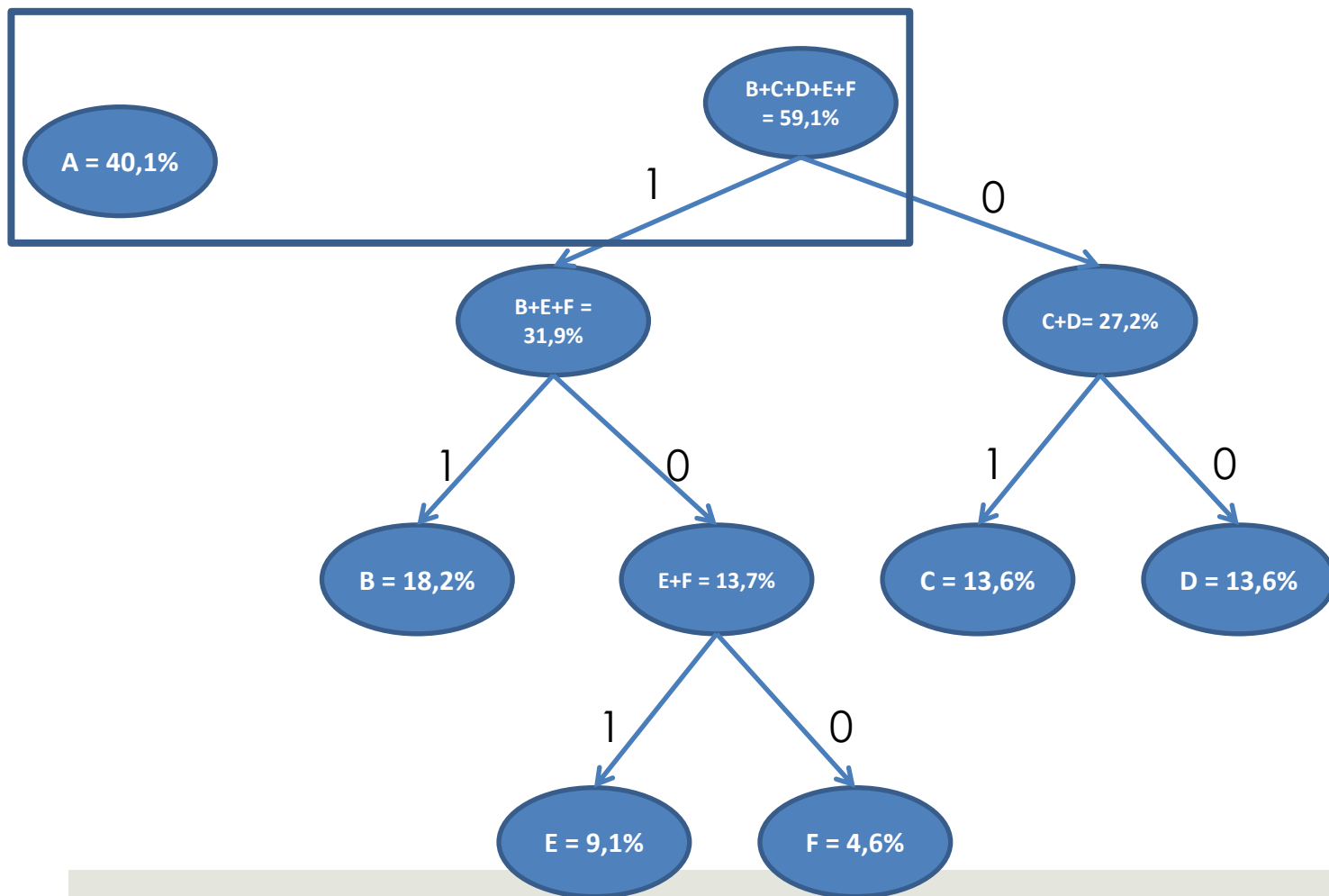
Codificação de Huffman

Exemplo:



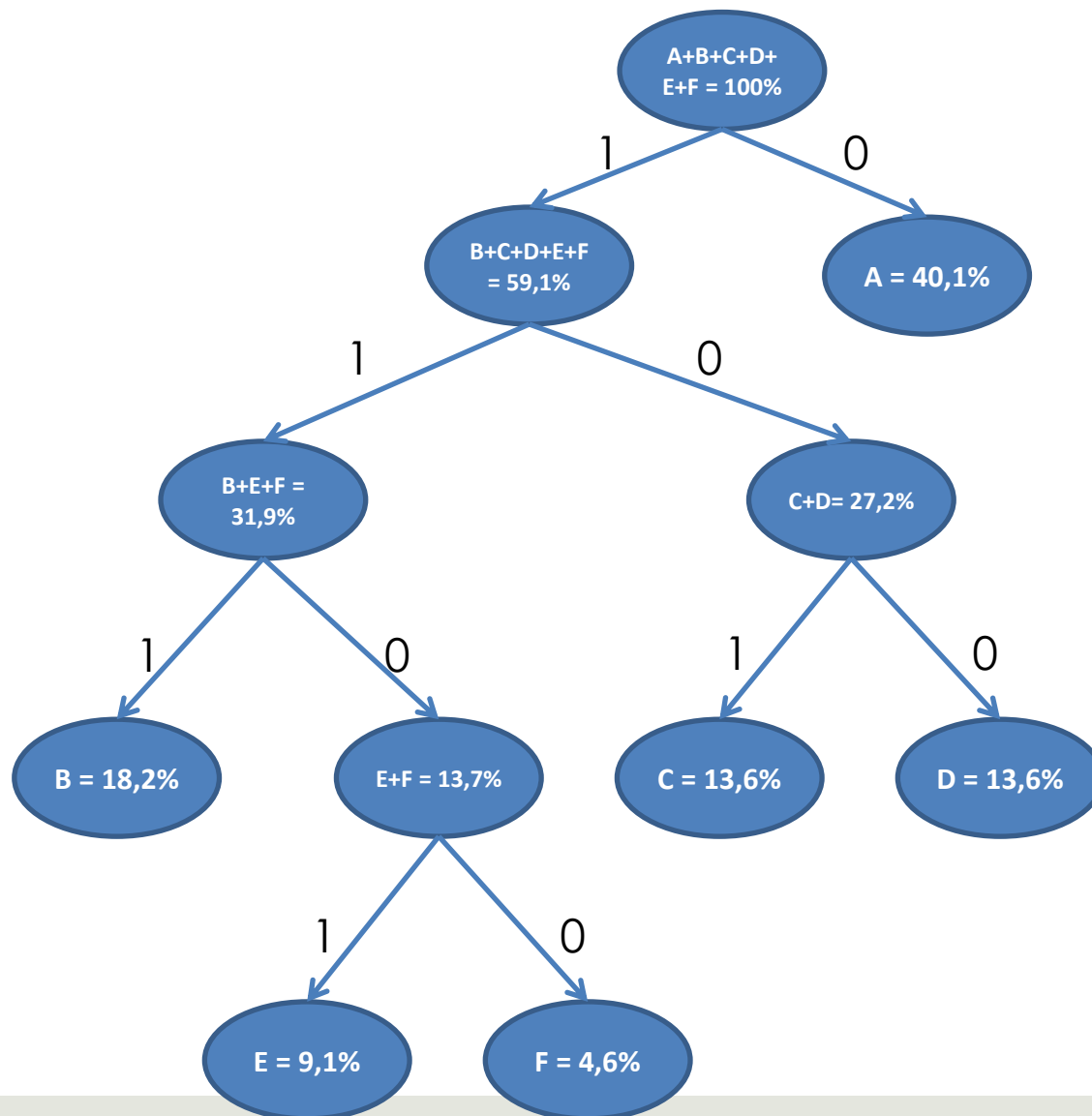
Codificação de Huffman

Exemplo:



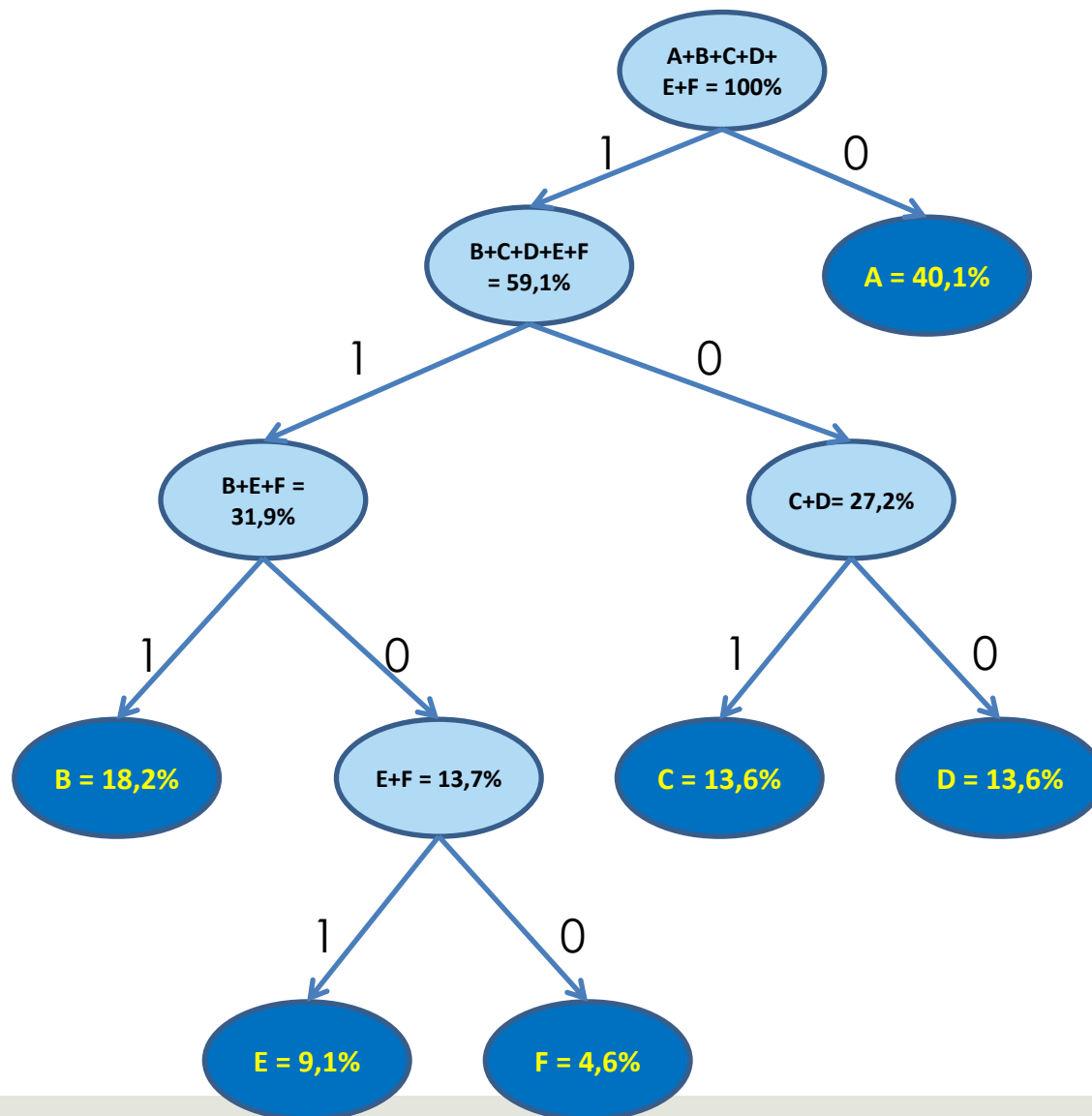
Codificação de Huffman

Exemplo:



Codificação de Huffman

Exemplo:



Codificação de Huffman

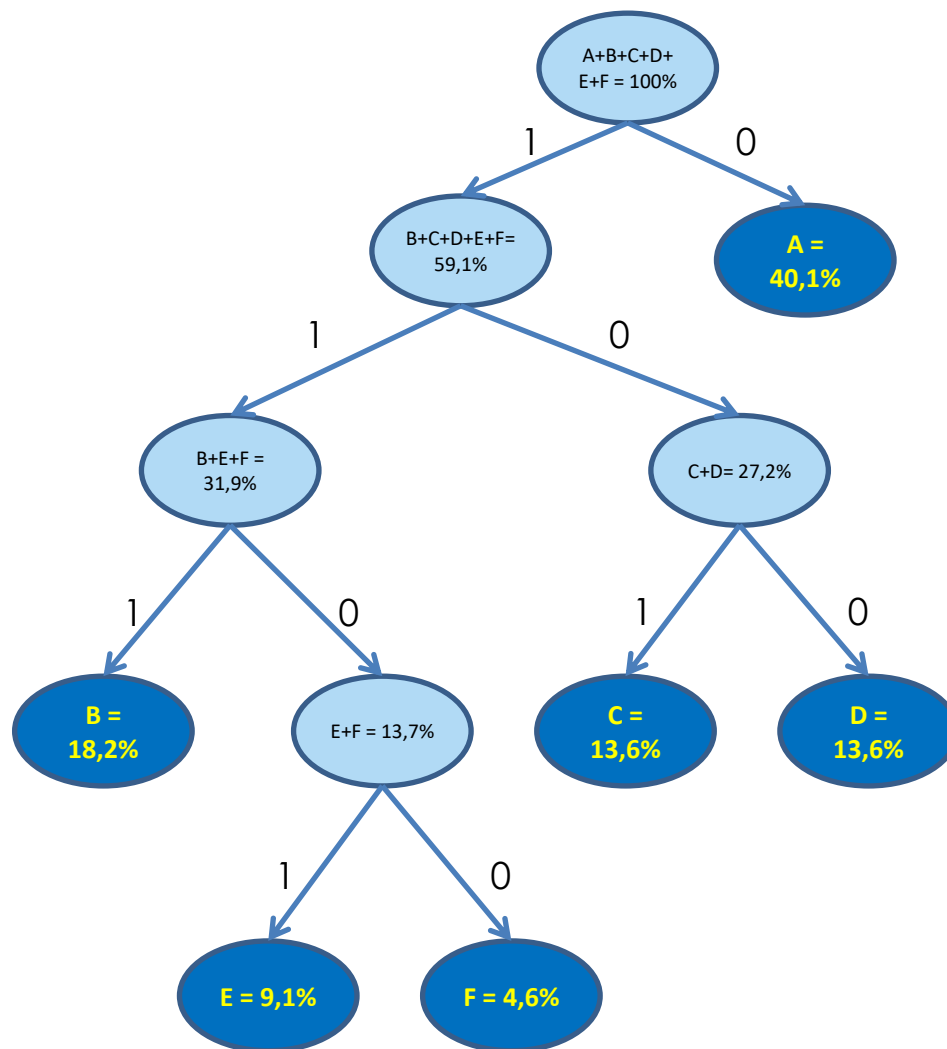
Exemplo:

❖ Códigos:

- ❖ A = 0
- ❖ B = 111
- ❖ C = 101
- ❖ D = 100
- ❖ E = 1101
- ❖ F = 1100

❖ Geramos alguns códigos de 4 bits!

❖ Isso é pior do que o que tínhamos?



Codificação de Huffman

Exemplo:

A = 9; B = 4; C = 3; D = 3; E = 2; F = 1

❖ Códigos:

A = 0 => 9 bits

B = 111 => 12 bits

C = 101 => 9 bits

D = 100 => 9 bits

E = 1101 => 8 bits

F = 1100 => 4 bits

❖ *Código de Huffman:*

AAAAAAAAAABBBBCCCDDEEF

0000000001111111111101101101100100100110111011100

❖ **Total = 51 bits!** **Redução de 22,7%!**

Codificação de Huffman

Exercício:

- ❖ Defina o código Huffman para o seguinte conjunto de símbolos, considerando o percentual de ocorrência informado:
 - 0 = 45%
 - 1 = 20%
 - 1 = 18%
 - 2 = 10%
 - 2 = 7%
- ❖ Considerando que são usados 8 bits para cada amostra, defina o percentual de redução de bits atingido com a codificação Huffman

Codificação de Huffman

- ❖ Várias tabelas podem ser utilizadas, uma para cada característica de amostras de entrada
 - ❖ Por exemplo, duas tabelas
 - ❖ Uma para amostras com baixa amplitude
 - ❖ Outra para amostras com alta amplitude
 - ❖ Custo adicional para informar a tabela utilizada
- ❖ A codificação Huffman pode gerar códigos muito grandes para elementos com baixas probabilidades
 - ❖ Podemos otimizar a árvore com o uso de códigos de exceção
 - ❖ A árvore é construída apenas para os elementos mais prováveis, os demais concatenam o valor original a um código de exceção

Run-Length Encoding (RLE)

- ❖ Uma repetição de símbolos iguais pode ser representada pelo símbolo, e o número referente as repetições
- ❖ Exemplo:
 - ❖ A sequência AAAAAAAAAA, pode ser representada como (10,A)
- ❖ Necessita de um caractere especial, que não faça parte do dicionário de símbolos
- ❖ Exemplo 2:
 - ❖ A codificação do texto “all is too well” seria “a2l is t2o we2l”
 - ❖ Como identificar se o número 2 estava realmente presente no texto original ou foi introduzido pela codificação?
 - ❖ Usando "@" como caractere especial, teremos:
 - ❖ a@2l is t@2o we@2l

Run-Length Encoding (RLE)

- ❖ Não é muito aplicável para a compressão de texto
 - ❖ Qual a taxa de repetição de letras na língua inglesa?
 - ❖ E no português?
- ❖ Maior aplicação em codificação de imagem e vídeo
 - ❖ Característica dos píxeis em background



UNIVERSIDADE FEDERAL DE PELOTAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO
TEC2/TEC4: REDES MULTIMÍDIA (RMM)

Unidade 2

Conceitos Básicos de Codificação Digital

Prof. Guilherme Corrêa
gcorrea@inf.ufpel.edu.br