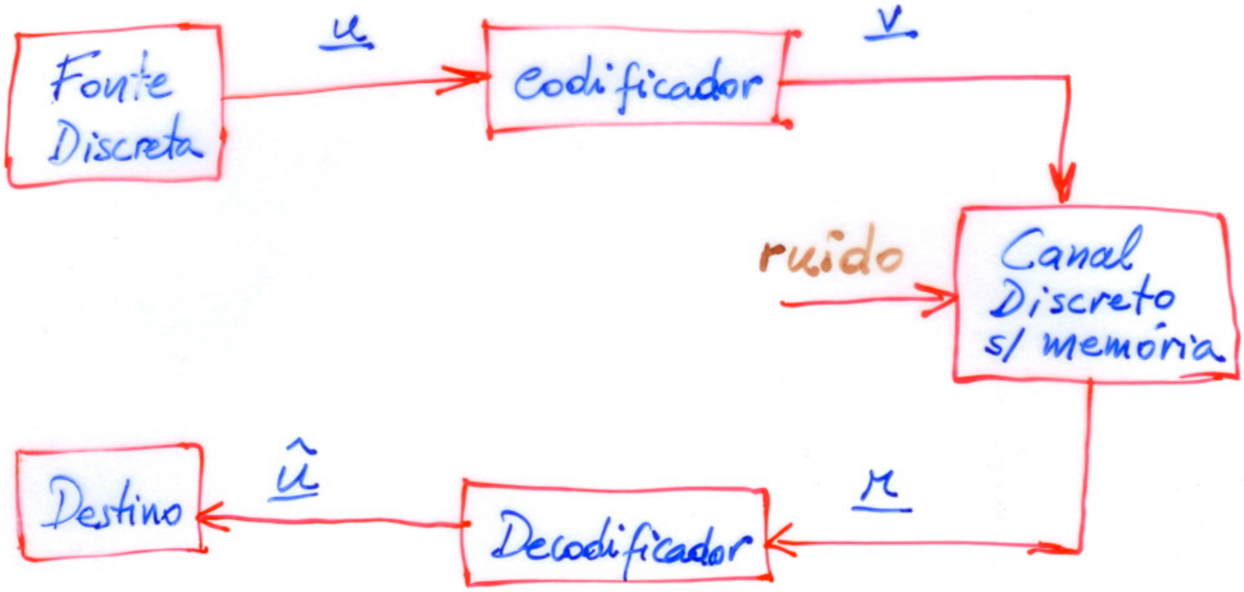


mensagem: $\underline{u} = (u_1, u_2, \ldots, u_k)$

palavra código: $\underline{v} = (v_1, v_2, \ldots, v_n)$

$\boxed{n > k}$

Fonte binária: $u_i \in \{0, 1\}$ ou $u_i \in \{-1, +1\}$

2^k mensagens distintas $\Longrightarrow$ 2^k palavras código distintas

k : # de bits de informação

$n-k$: # de bits de redundância ou de verificação de paridade

n : # de bits transmitidos

TAXA DO CÓDIGO: $\boxed{R \triangleq k/n}$

Código de bloco é o conjunto de palavras código.

Ex: $k=3$, $n=5$ $\Rightarrow R = 3/5$

$\underline{u}$	$\underline{v}$
000	01101
001	10110
010	11000
011	11010
100	00110
101	01100
110	11101
111	01011

Este código é não linear.

Sejam $\underline{v}$ e $\underline{v}'$ $\in$ código (palavras código)

$$\underline{v} \oplus \underline{v}' = (01101) \oplus (10110) = (11011) \notin \text{código.}$$

GF(2): Galois Field

$\oplus$	0	1
0	0	1
1	1	0

$\equiv$ XOR

$\otimes$	0	1
0	0	0
1	0	1

$\equiv$ AND

Não-linearidade $\Rightarrow$ tabela com 2^k linhas

Proibitivo para k grande !!!

Definição: Um código de bloco de comprimento n e com 2^k palavras código, denotado por $C(n,k)$, é **linear** se e somente se suas 2^k palavras código formam um subspaço k-dimensional do espaço de todas as n-uplas sobre $GF(2)$.

$$\underline{v} = \underline{u} \cdot G \qquad (G: \text{matriz geradora})$$

$$\dim(G) = k \times n$$

Se $\underline{v}$ e $\underline{v}'$ $\in$ código linear, então

$$\underline{v} \oplus \underline{v}' = \underline{u} \cdot G \oplus \underline{u}' \cdot G$$

$$= (\underline{u} + \underline{u}') \cdot G$$

$$= \underline{u}'' \cdot G \in \text{código } \underline{\text{linear}}.$$

Ex.1: Código de verificação de paridade, $R = 2/3$

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

u	v
00	000
01	011
10	101
11	110

Ex.2: Código de repetição, $R = 1/7$

$$G = [1\,1\,1\,1\,1\,1\,1]$$

u	v
0	0000000
1	1111111

Ex.3: Código de Hamming, $R = 4/7$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Distância de Hamming

$d(\underline{v}, \underline{v}') \triangleq$ # posições em que $\underline{v}$ e $\underline{v}'$ diferem.

Ex.: $d([0\,1\,1], [1\,1\,0]) = 2$.

Distancia Mínima de um Código:

$$d_{min} \triangleq \min \{ d(\underline{v}, \underline{v}') \mid \underline{v}, \underline{v}' \in \text{código}, \ \underline{v} \neq \underline{v}' \}.$$

Ex. 1: Código de verificação de paridade, $R = 2/3$

$\underline{u}$	$\underline{v}$
00	000
01	011
10	101
11	110

$$\begin{aligned} d_{min} &= \min \{ d([000],[011]), d([000],[101]), \\ &\quad d([000],[110]), d([011],[101]), \\ &\quad d([011],[110]), d([101],[110]) \} \\ &= \min \{2, 2, 2, 2, 2, 2\} \\ &= \underline{\underline{2}} \end{aligned}$$

Ex. 2: Código de repetição, $R = 1/n$

$d_{min} = \underline{\underline{n}}$

Peso de Hamming

$w(\underline{v}) \triangleq \#$ posições $\neq 0$.

Ex.: $w([011]) = 2$

PROPRIEDADE: $d(\underline{v}, \underline{v}') = w(\underline{v} \oplus \underline{v}')$

Ex.: $d([011],[110]) = w([011] \oplus [110])$
$= w([101]) = 2$

Para códigos lineares:

$d_{min} \triangleq \min \{ d(\underline{v}, \underline{v}') \mid \underline{v}, \underline{v}' \in \text{código}, \underline{v} \neq \underline{v}' \}$

$= \min \{ w(\underline{v} \oplus \underline{v}') \mid \underline{v}, \underline{v}' \in \text{código}, \underline{v} \neq \underline{v}' \}$

$= \min \{ w(\underline{v}'') \mid \underline{v}'' \in \text{código}, \underline{v}'' \neq 0 \}$

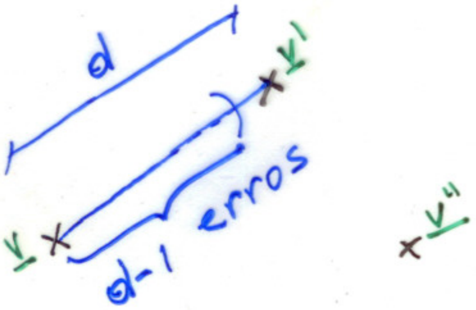
$\Longrightarrow \boxed{d_{min} = w_{min}}$

Ex.: Código de verificação de paridade, $R = 2/3$

$\underline{u}$	$\underline{v}$
00	000
01	011
10	101
11	110

$d_{min} = w_{min} = 2$.

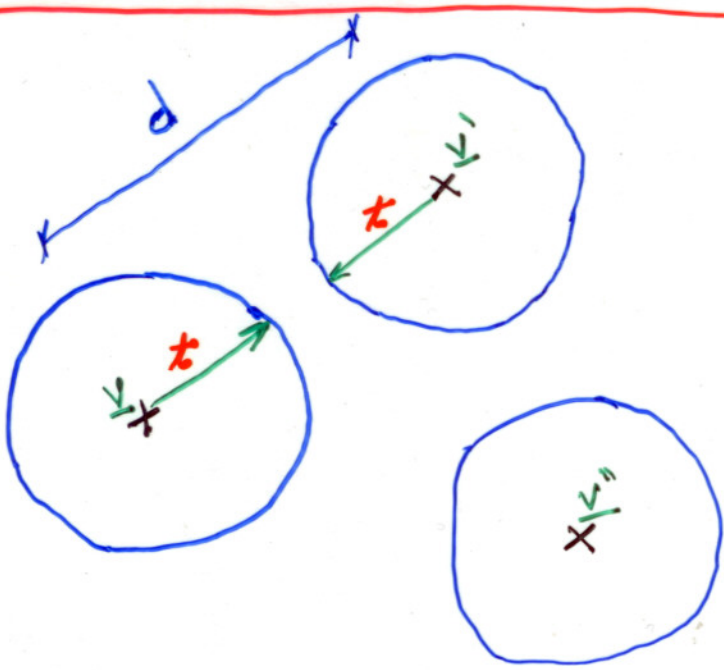
Capacidade de DETECÇÃO de um código com distância mínima $d_{min} = d$.



detecta até $d-1$ erros

× palavras código

Capacidade de CORREÇÃO de um código com distância mínima $d_{min} = d = 2t + 1$



corrige até t erros.

$d = 2t + 1$ ou $t = \left\lfloor \frac{d-1}{2} \right\rfloor$

Ex.: Código de Hamming, $R = 4/7$, $d_{min} = 3 = 2t + 1$
corrige até 1 erro. $\boxed{t = 1}$

Parâmetro a ser optimizado: $\boxed{d_{min}}$

Códigos Convolucionais

$$\underline{v}_j = \underline{u}_j \cdot G_0 + \underline{u}_{j-1} \cdot G_1 + \cdots + \underline{u}_{j-m} \cdot G_m$$

$$= \sum_{i=0}^{m} \underline{u}_{j-i} \cdot G_i \quad \text{(convolução discreta)}$$

$\dim(G_i) = k \times n$

m = memória

Matriz geradora vista como a de um código de bloco:

$$G = \begin{bmatrix} G_0 & G_1 & \cdots & G_m & & & \\ & G_0 & G_1 & \cdots & G_m & & \\ & & G_0 & G_1 & \cdots & G_m & \\ & & & & \ddots & & \end{bmatrix}$$

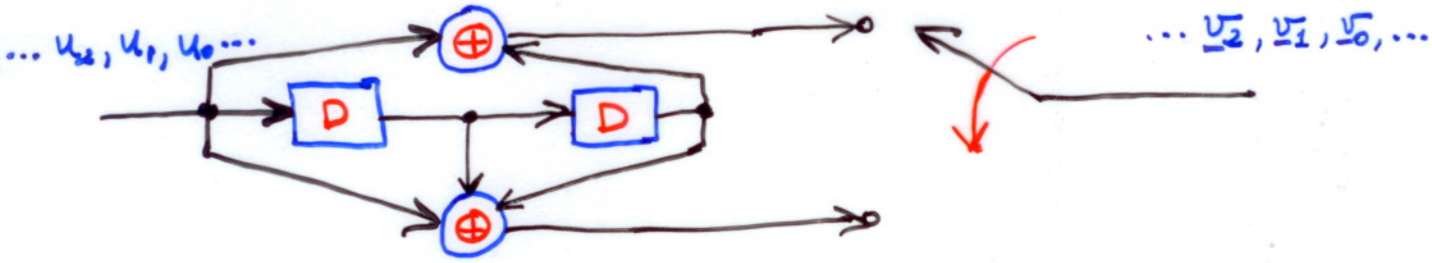
Códigos convolucionais são sempre **lineares**!

Ex.: Código convolucional, $R = 1/2$, $m = 2$

$G_0 = [1\ 1]$ $G_1 = [0\ 1]$ $G_2 = [1\ 1]$

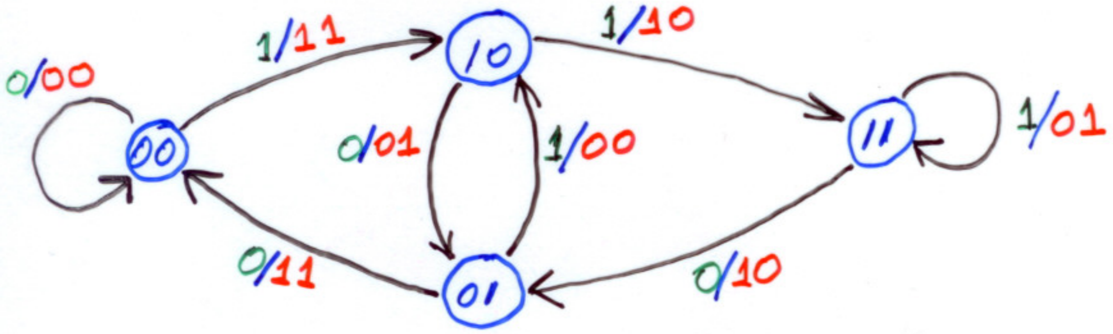
Representações do codificador convolucional

1) Circuito sequencial linear

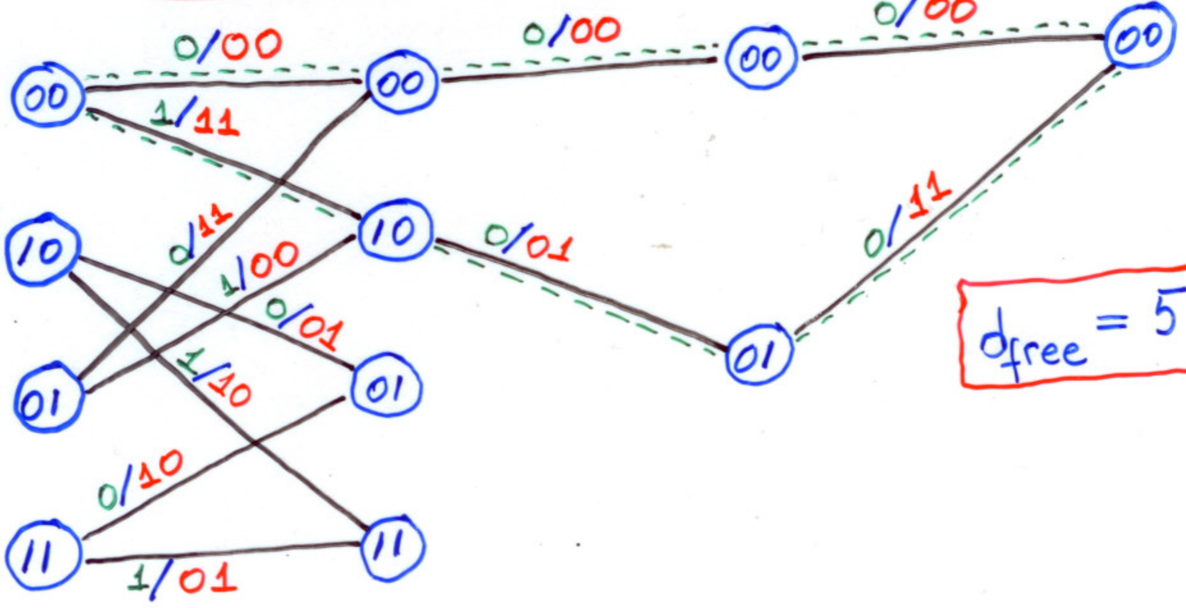


2) Diagrama de estados

$\underline{u} / \underline{v}$



3) Diagrama de treliça



$\boxed{d_{free} = 5}$