ELE32
Introdução a Comunicações
Códigos convolucionais

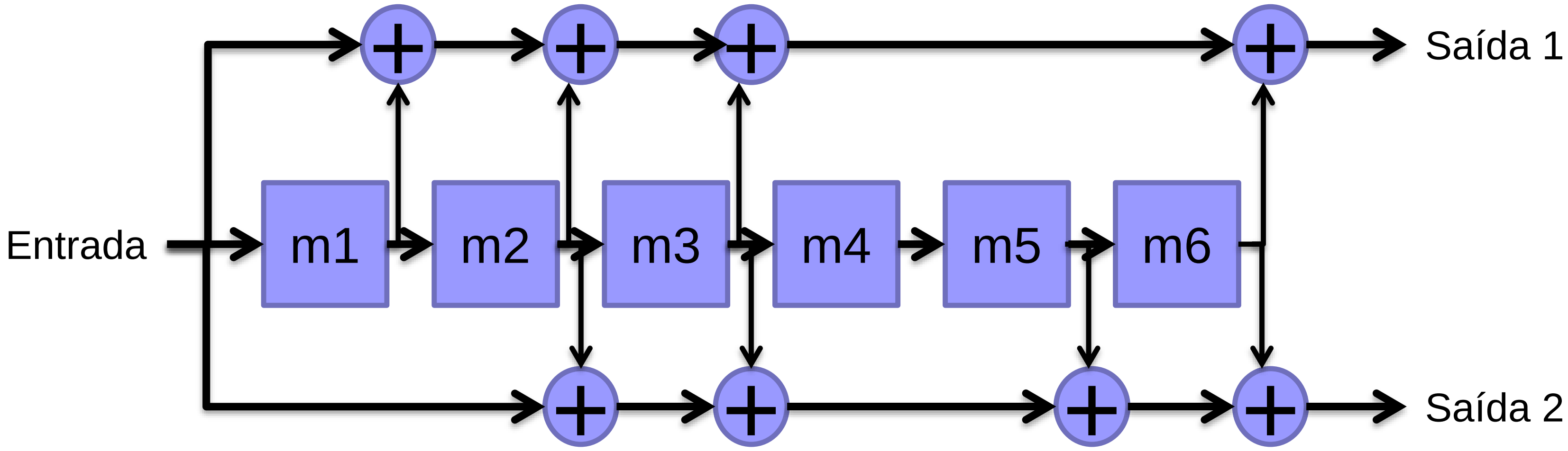
ITA

2º. Semestre de 2019

manish@ita.br

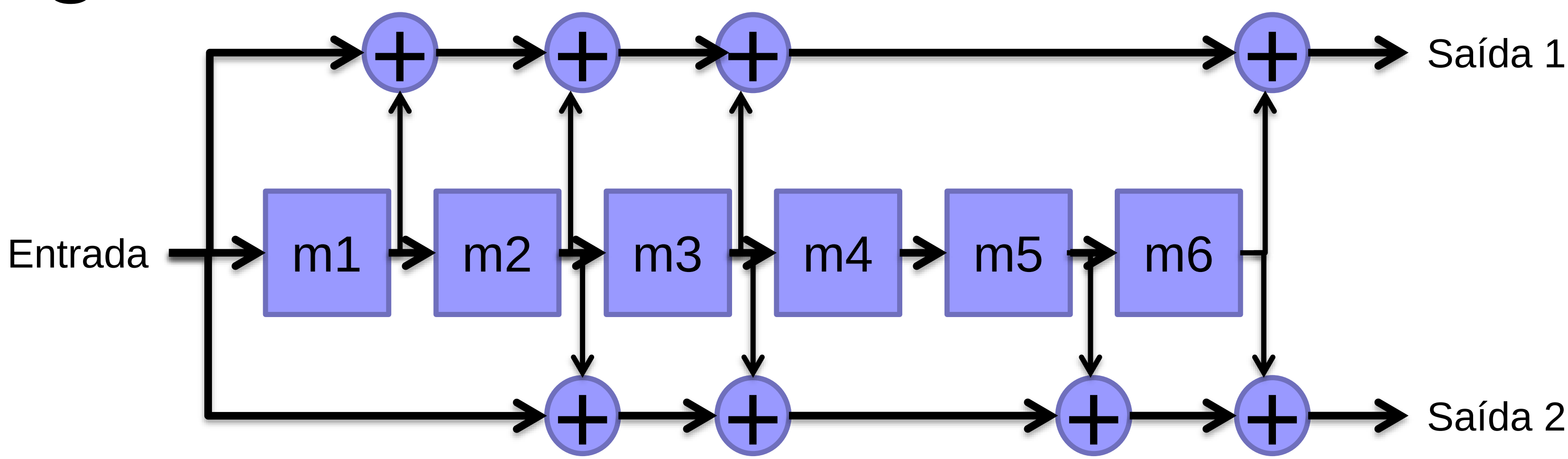
Código convolucional

- Converte uma sequência de bits de informação em uma sequência de bits a serem transmitidos



- Neste exemplo a taxa do codificador é ½. As saídas são multiplexadas para formar uma única sequência

Representações: Matriz geradora

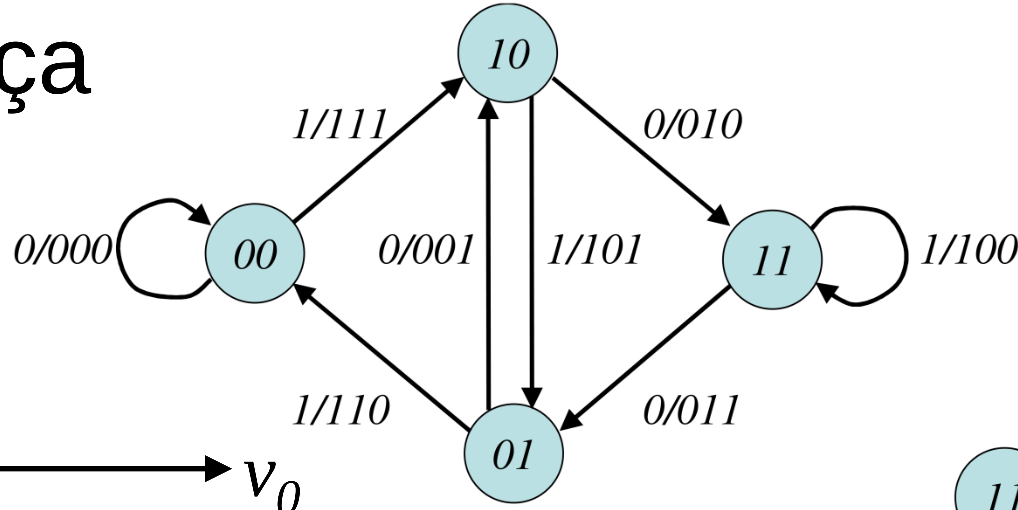


$$\mathbf{G}(D) = [1 + D + D^2 + D^3 + D^6 \qquad 1 + D^2 + D^3 + D^5 + D^6]$$

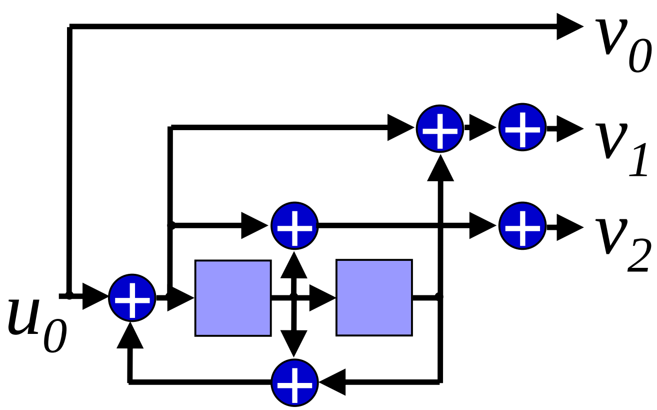
$$\mathbf{v}(D) = u(D)\mathbf{G}(D)$$

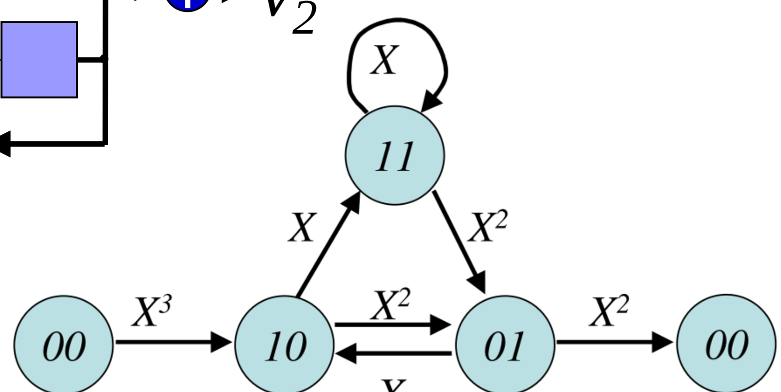
$$v(D) = \sum_{i=1}^{n} v_i(D^n)D^{i-1}$$

Representações: diagrama de estados e treliça

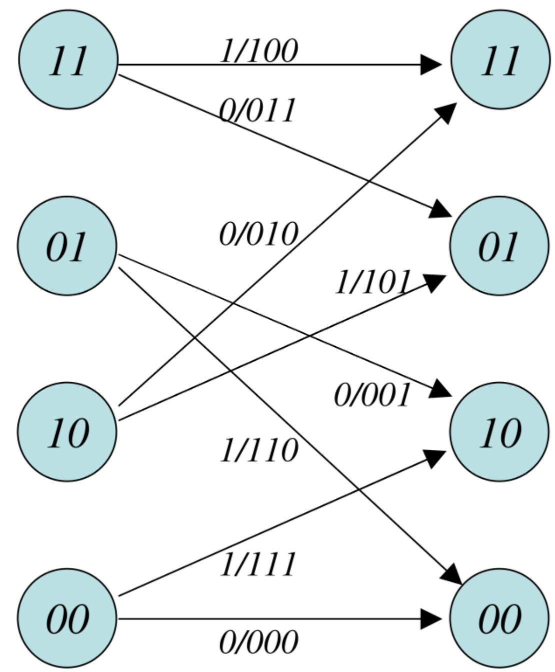


(a)





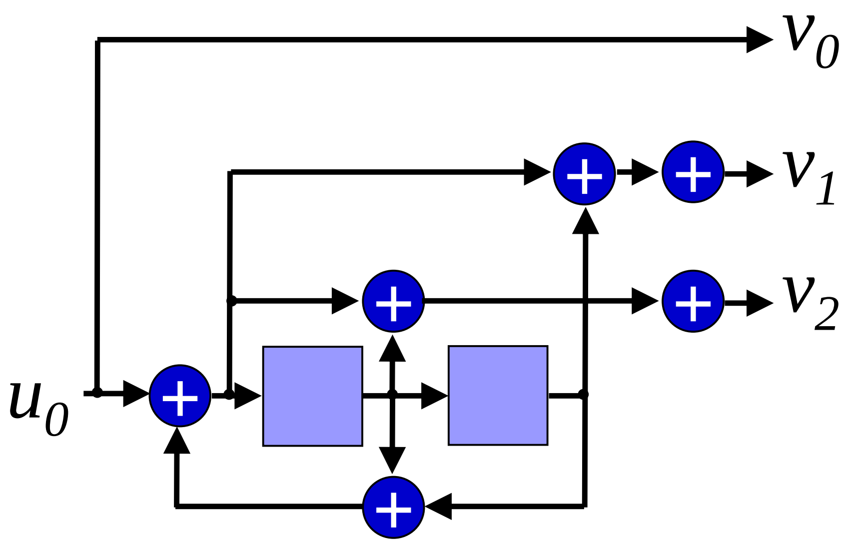
(b)



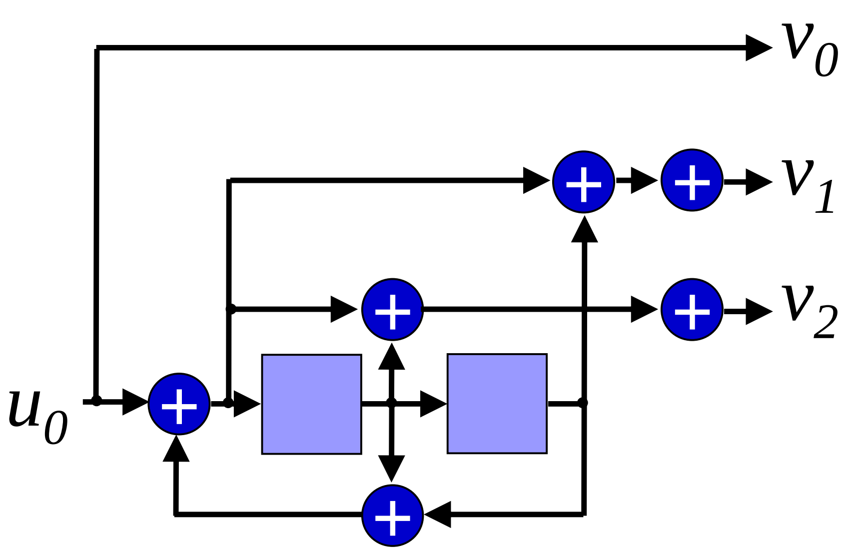
(c)

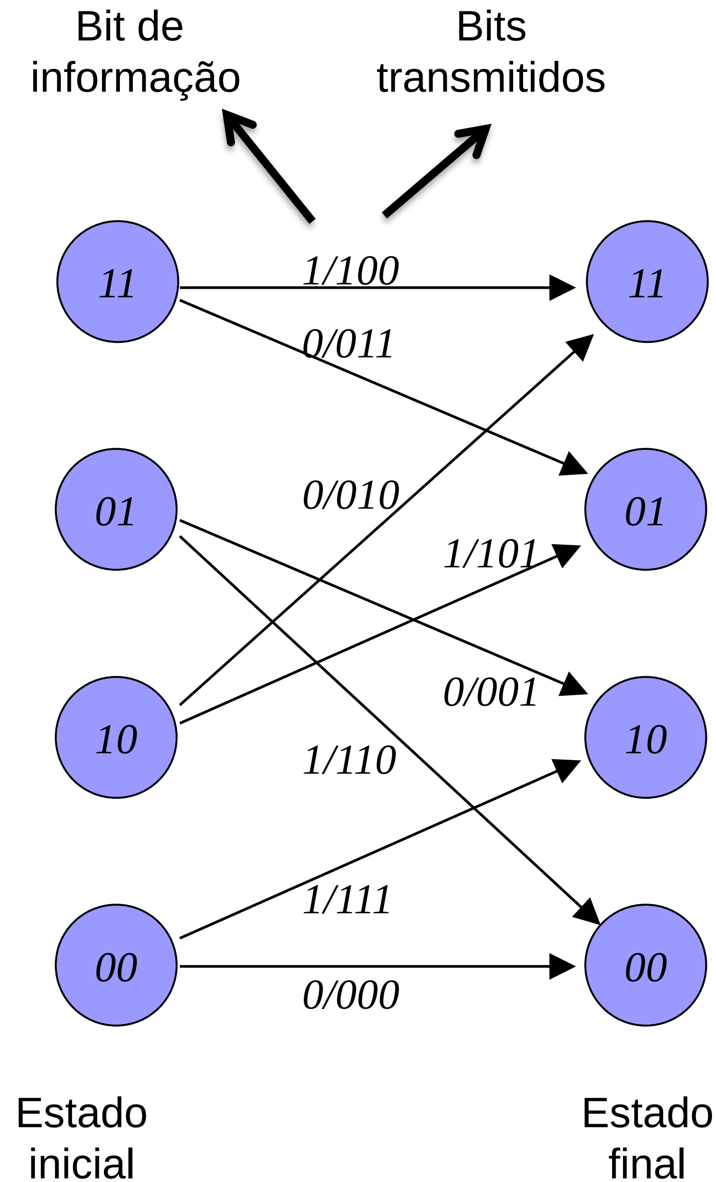
Algoritmo de Viterbi

- É mais fácil demonstrado do que explicado
- Um codificador convolucional possui uma seção de treliça associada

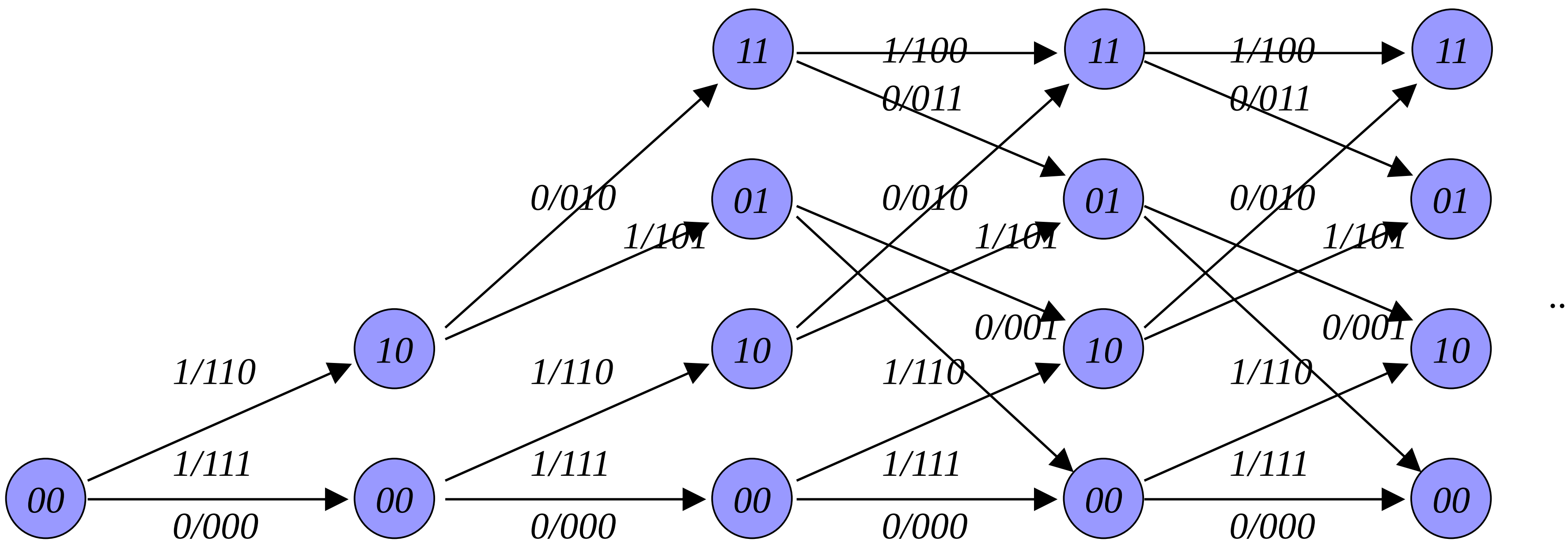


Seção de treliça

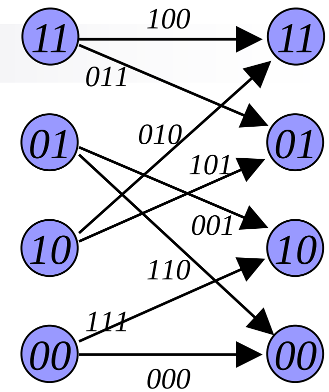




Treliça de um codificador

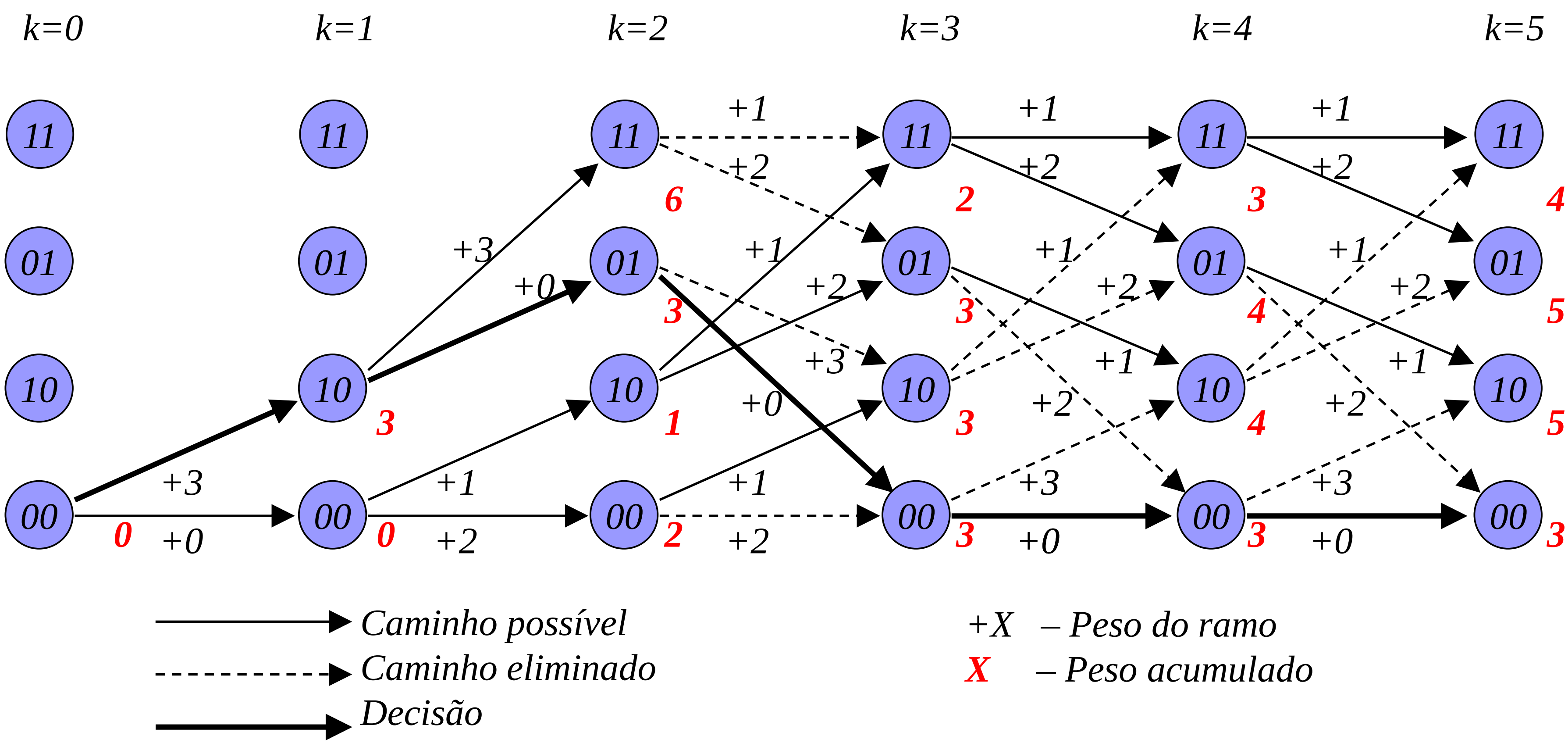


Algoritmo de Viterbi



Vetor recebido

000 *101* *110* *000* *000*



Algoritmo de Viterbi: formalismo matemático (1/2)

- Definições:
  - σ_i^k :i-ésimo estado no instante k, i=0,1,2,...,2^m-1 e k = 0,1,2,...,K
  - $C(\sigma_i^k)$: custo do i-ésimo estado no instante k
  - ρ_{ij}^k : ramo que causa a transição de σ_i^k para σ_j^{k+1}
  - $C(\rho_{ij}^k)$: custo do ramo que causa a transição de σ_i^k para σ_j^{k+1}
  - $s(\rho_{ij}^k)$: símbolo de entrada associado ao ramo ρ_{ij}^k .
  - $t(\rho_{ij}^k)$: símbolo de saída (transmitido) associado ao ramo ρ_{ij}^k .
  - r^k símbolo recebido no k-ésimo instante.
  - $\boldsymbol{s}_i^k = [s_0,s_1,s_2,...,s_{k-1}]$: Sequência de símbolos que formam o caminho sobrevivente que chega ao estado σ_i^k

Algoritmo de Viterbi: formalismo matemático (2/2)

1. Iniciação de variáveis

$$C(\sigma_0^0) = 0$$
$$C\left(\sigma_i^0\right) = \infty$$
$$k = 0, \boldsymbol{s}_0^0 = \{\}$$

2. Cálculo dos custos dos ramos:

$$C\left(\rho_{i,j}^k\right) = d_H(r^k, t(\rho_{i,j}^k))$$

3. Calculo dos custos dos estados futuros e seleção de caminho sobrevivente.

$$i(j)* = \min_i\{C\left(\sigma_i^k\right) + C\left(\rho_{i,j}^k\right)\}$$
$$C\left(\sigma_j^{k+1}\right) = C\left(\sigma_{i(j)*}^k\right) + C\left(\rho_{i(j)*,j}^k\right)$$
$$\boldsymbol{s}_j^{k+1} = [\boldsymbol{s}_{i(j)*}^{k+1} \quad s(\rho_{i(j)*,j}^k)]$$

4. Se k < K(tamanho da sequência), k = k + 1. Retorne ao passo 2.
5. Caminho vitorioso é aquele associado ao estado final de menor custo

Atividade

1. Implemente um codificador convolucional para os códigos abaixo (notação octal, bit mais significativo a esquerda de cada octeto)

m	$g_1(D)$	$g_2(D)$	$g_3(D)$
3	13	15	17
4	25	33	37
6	117	127	155

Notação octal:
13 = 001011
que equivale a
$1+0D+1D^2+1D^3$

2. Simule a passagem de uma sequência muito grande de bits de informação por um canal BSC com parametro p de forma análoga a feita nos laboratórios anteriores (dica: o código linear)
3. Implemente o algoritmo de Viterbi para decodificar a sequência recebida
4. Estime a probabilidade de erro para os bits de informação para cada código e cada valor de p
5. Desafio muito difícil: Implemente um código Turbo e compare com desempenho dos códigos convolucionais “normais”