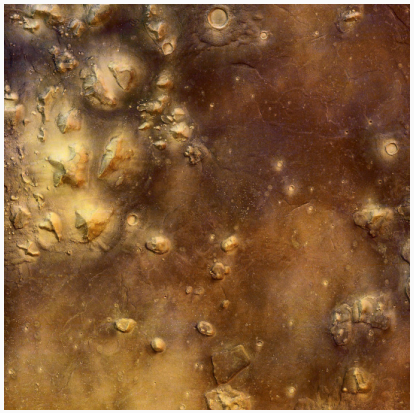
Teoria da Informação e Codificação

Códigos convolucionais



Prof. Roberto Wanderley da Nóbrega

Instituto Federal de Santa Catarina

Introdução

Códigos convolucionais

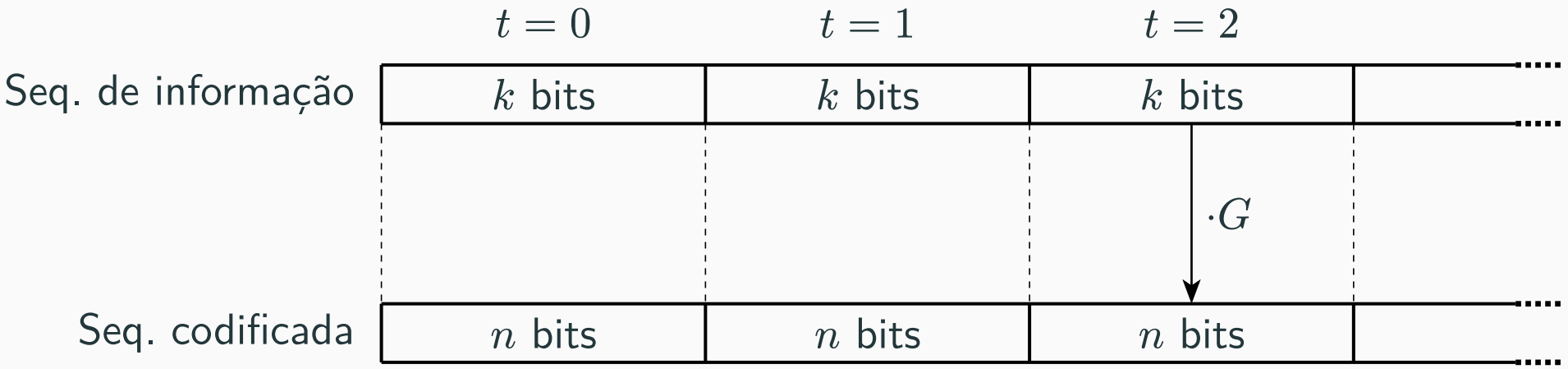
- Propostos por Peter Elias em 1955 como alternativa aos códigos de bloco.
- Generalizam códigos de bloco lineares.



Peter Elias (1923–2001).
Fonte: MIT Museum.

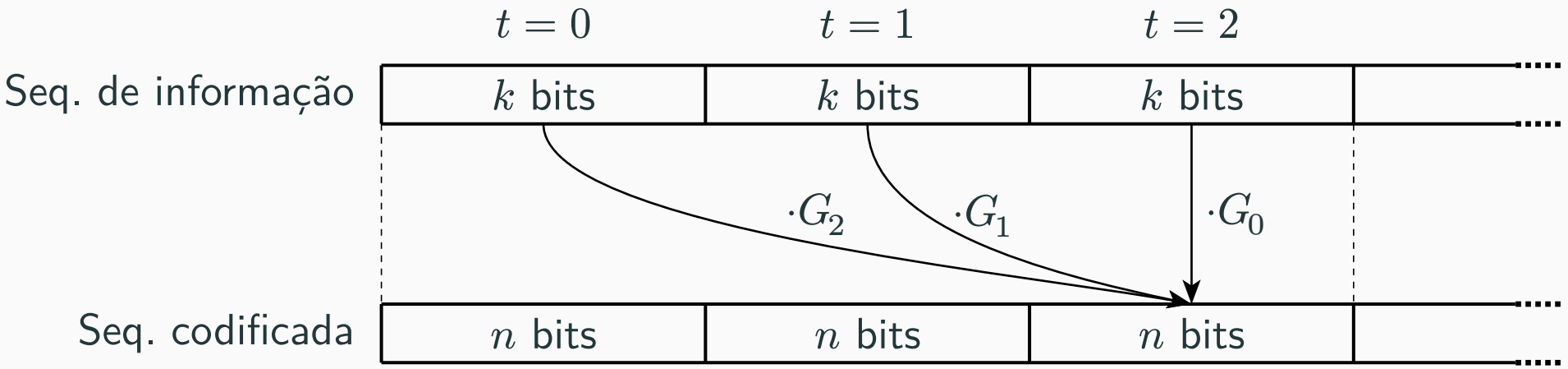
Códigos de bloco lineares:

$$v_t = u_t G.$$



Códigos convolucionais:

$$v_t = u_t G_0 + u_{t-1} G_1 + u_{t-2} G_2.$$



A “convolução”

De modo geral,

$$v_t = u_t G_0 + u_{t-1} G_1 + ... + u_{t-\mu} G_\mu,$$

onde μ é a **ordem de memória** do código e $G_0, G_1, ..., G_\mu$ são **matrizes geradoras**.

Ideia: Melhorar as propriedades do código aumentando μ, mantendo k e n pequenos.

Observações:

- Utilizaremos a nomenclatura “código (n, k, μ)”.
- As sequências u (informação) e v (codificada) são infinitas.
- A taxa continua sendo definida como $R = k/n$.

Notação

Convenção:

- Subscritos indicam o tempo.
- Sobrescritos entre parênteses indicam o índice do bit.

Assim:

$$u_t = \begin{bmatrix} u_t^{(0)} & u_t^{(1)} & \dots & u_t^{(k-1)} \end{bmatrix}$$
$$v_t = \begin{bmatrix} v_t^{(0)} & v_t^{(1)} & \dots & v_t^{(n-1)} \end{bmatrix}$$

Observação: Manteremos a convenção de indexação começando em 0.

Exercício 1

Considere o código convolucional com matrizes geradoras

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{e} \quad G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

(a) Determine os parâmetros (n, k, μ) do código.

(b) Determine os elementos de v_t em termos dos elementos de u_t e u_{t-1}.

(c) Esboce o diagrama de blocos correspondente.

(d) Codifique a sequência de informação $u = \texttt{11 01 10}\,\texttt{00 00 00}\ldots$

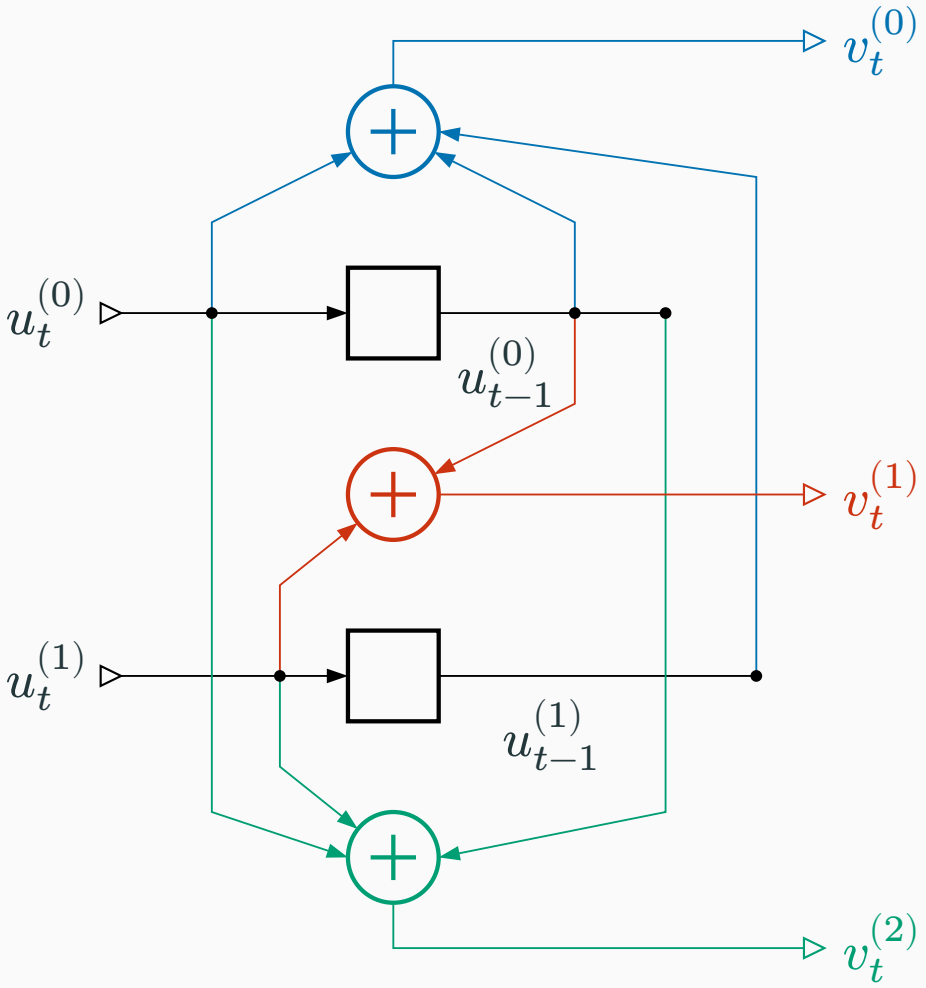
Exercício 1: Respostas

(a) $(n, k, \mu) = (3, 2, 1)$.

(b) $v_t^{(0)} = u_t^{(0)} + u_{t-1}^{(0)} + u_{t-1}^{(1)}$,
$v_t^{(1)} = u_t^{(1)} + u_{t-1}^{(0)}$,
$v_t^{(2)} = u_t^{(0)} + u_t^{(1)} + u_{t-1}^{(0)}$.

(c) Ao lado.

(d) $v = \texttt{110 000 001 111 000 000 000}\ldots$



Exercício 2

Considere o código convolucional com matrizes geradoras

$$G_0 = [1 \;\; 1], \quad G_1 = [1 \;\; 0] \quad \text{e} \quad G_2 = [1 \;\; 1].$$

(a) Determine os parâmetros (n, k, μ) do código.

(b) Determine os elementos de v_t em termos dos elementos de u_t, u_{t-1} e u_{t-2}.

(c) Esboce o diagrama de blocos correspondente.

(d) Codifique a sequência de informação $u = \mathtt{1011000}\ldots$

Exercício 2: Respostas

(a) $(n, k, \mu) = (2, 1, 2)$.

(b) $v_t^{(0)} = u_t^{(0)} + u_{t-1}^{(0)} + u_{t-2}^{(0)}$,
$v_t^{(1)} = u_t^{(0)} + u_{t-2}^{(0)}$.

(c) Ao lado.

(d) $v = \texttt{11 10 00 01 01 11 00 00 00}\ldots$

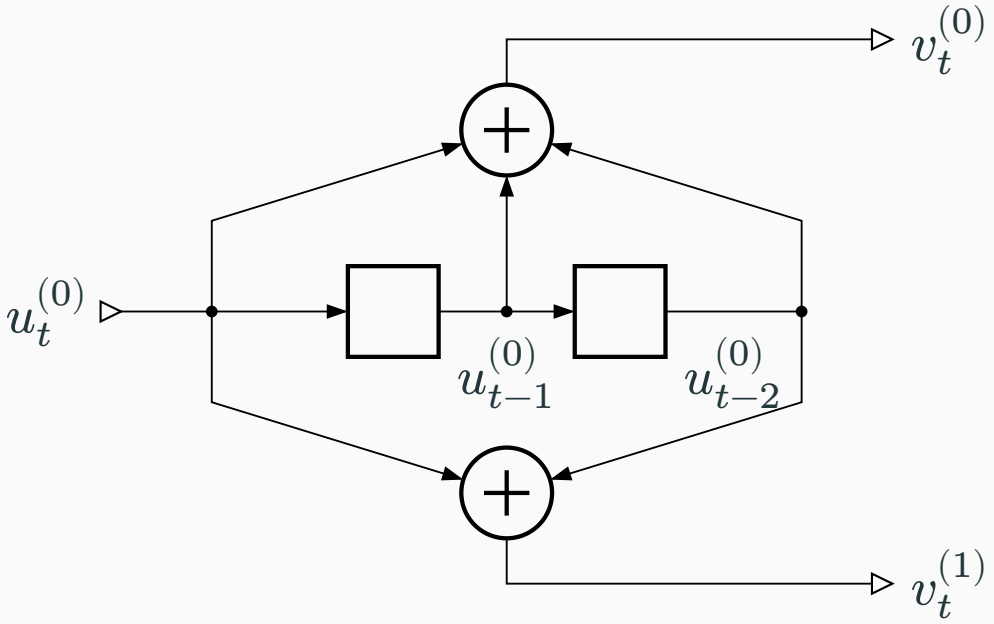


Diagrama de estados

É possível representar o codificador de um código convolucional através de uma **máquina de estados** (máquina de Mealy), com entrada u_t (k bits) e saída v_t (n bits).

- O estado da máquina é formado pelas *saídas* dos blocos de atraso no diagrama de blocos.
- Portanto, o número de estados é 2^σ, onde σ é o número de blocos de atraso.
- O parâmetro σ também é chamado de **grau** do codificador.

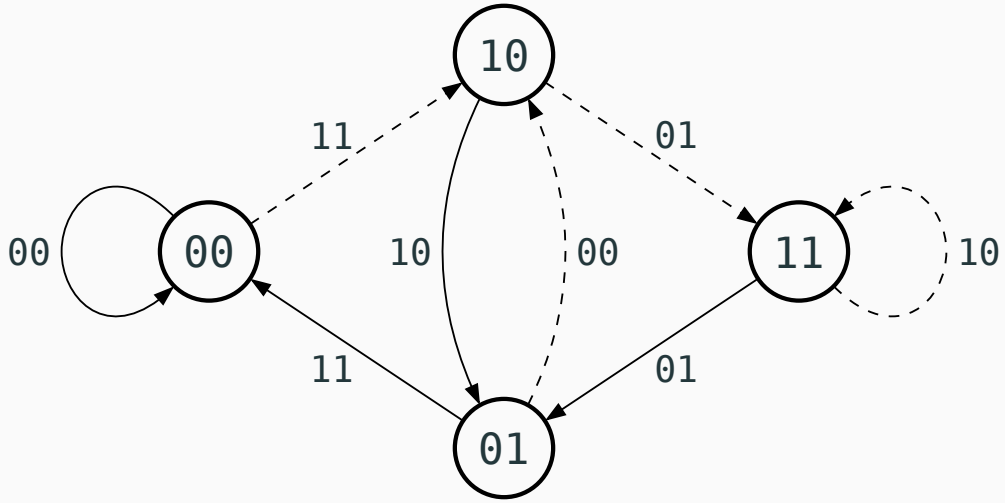
Exercício 3

Determine o diagrama de estados do código $(2, 1, 2)$ do Exercício 2 e codifique novamente a sequência de informação $u = \mathtt{1\,0\,1\,1}\,\mathtt{0\,0\,0}\ldots$

Exercício 3

Determine o diagrama de estados do código $(2, 1, 2)$ do Exercício 2 e codifique novamente a sequência de informação $u = \mathtt{1\,0\,1\,1}\,\mathtt{0\,0\,0}\ldots$

Resposta: Estado: $u_{t-1}u_{t-2}$; rótulo: v_t; linha cheia: $u_t = \mathtt{0}$; tracejada: $u_t = \mathtt{1}$.



$$v = \mathtt{11\,10\,00\,01\,01\,11}\,\mathtt{00\,00\,00}\ldots$$

Representação polinomial

Transformada D

A **transformada D** de uma sequência binária $x = (x_0, x_1, x_2, ...)$ é a série

$$x(D) = x_0 + x_1 D + x_2 D^2 + \cdots.$$

Multiplicar por D corresponde a atrasar a sequência em um instante. Por exemplo:

$$\begin{aligned} x_t: &\quad \texttt{1011}\texttt{000}\ldots &\leftrightarrow &\quad x(D) = 1 + D^2 + D^3, \\ x_{t-1}: &\quad \texttt{0101}\texttt{100}\ldots &\leftrightarrow &\quad Dx(D) = D + D^3 + D^4. \end{aligned}$$

Observações:

- D vem de *delay* (atraso).
- É a transformada z com $D = z^{-1}$, mas com aritmética módulo 2.
- Se a sequência é finita, $x(D)$ é um polinômio; se é infinita, é uma série de potências.

Formulação polinomial

Convertendo as sequências de entrada e de saída em polinômios:

$$u(D) = \begin{bmatrix} u^{(0)}(D) & u^{(1)}(D) & \cdots & u^{(k-1)}(D) \end{bmatrix},$$
$$v(D) = \begin{bmatrix} v^{(0)}(D) & v^{(1)}(D) & \cdots & v^{(n-1)}(D) \end{bmatrix}.$$

A **matriz geradora polinomial** é a matriz $k \times n$

$$G(D) = G_0 + G_1 D + \cdots + G_\mu D^\mu.$$

Convolução no tempo vira produto no domínio transformado:

$$v(D) = u(D)G(D).$$

Exercício 4

Considere novamente o código convolucional do Exercício 1, com matrizes geradoras

$$G_0 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{e} \quad G_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

(a) Determine a matriz geradora polinomial $G(D)$.

(b) Para a sequência de informação $u = \texttt{11 01 10 00 00 00}\ldots$, determine $u(D)$.

(c) Calcule $v(D) = u(D)G(D)$ e compare com o resultado do Exercício 1.

Exercício 4: Respostas

(a) $G(D) = \begin{bmatrix} 1+D & D & 1+D \\ D & 1 & 1 \end{bmatrix}$.

(b) $u = 11\,01\,10\,00\,00\,00\ldots$

$u^{(0)} = 101000\ldots$ e $u^{(1)} = 110000\ldots$

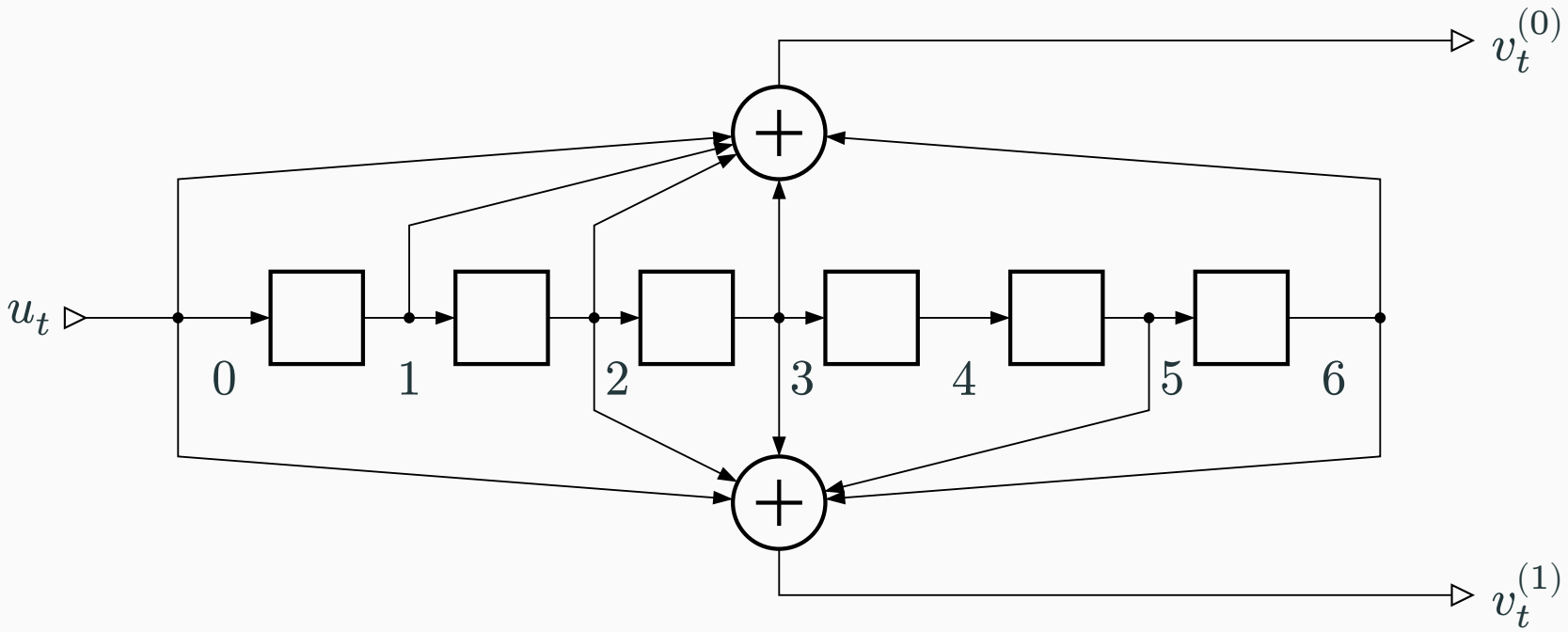
$u(D) = \begin{bmatrix} 1+D^2 & 1+D \end{bmatrix}$.

(c) $v(D) = \begin{bmatrix} 1+D^3 & 1+D^3 & D^2+D^3 \end{bmatrix}$.

$v^{(0)} = 1001000\ldots$, $v^{(1)} = 1001000\ldots$ e $v^{(2)} = 0011000\ldots$

$v = 110\,000\,001\,111\,000\,000\,000\ldots$

Código $(2, 1, 6)$ com $G(D) = \left[1 + D + D^2 + D^3 + D^6 \quad 1 + D^2 + D^3 + D^5 + D^6\right]$.



Aplicações: Telemetria espacial (Voyager 1 e 2, CCSDS), Wi-Fi (IEEE 802.11), TV digital (DVB-T).

Terminação de códigos convolucionais

Na prática, a sequência de informação u não é infinita e **técnicas de terminação** são necessárias, transformando um código convolucional (n, k, μ) em um código de bloco (n', k').

Motivação

Na prática, a sequência de informação u não é infinita e **técnicas de terminação** são necessárias, transformando um código convolucional (n, k, μ) em um código de bloco (n', k').

Seja h o número de blocos de informação (cada qual contendo k bits) a serem codificados.

- A *dimensão* k' do código de bloco resultante é $k' = hk$.
- Já o *comprimento* n' depende da técnica de terminação utilizada.

Motivação

Na prática, a sequência de informação u não é infinita e **técnicas de terminação** são necessárias, transformando um código convolucional (n, k, μ) em um código de bloco (n', k').

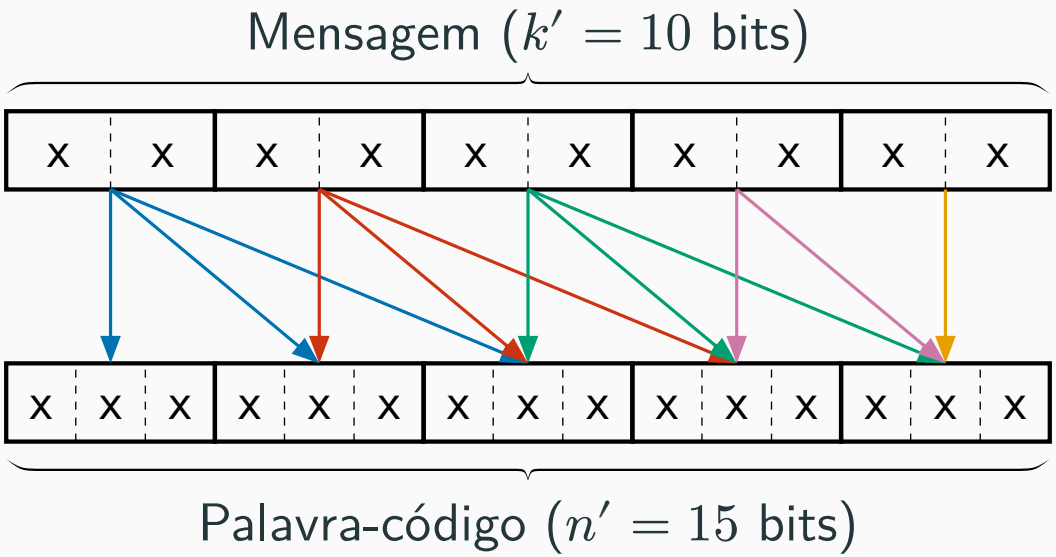
Seja h o número de blocos de informação (cada qual contendo k bits) a serem codificados.

- A *dimensão* k' do código de bloco resultante é $k' = hk$.
- Já o *comprimento* n' depende da técnica de terminação utilizada.

Três técnicas de terminação: **truncamento direto**, **estado zero** e **tail-biting**.

Terminação por truncamento direto

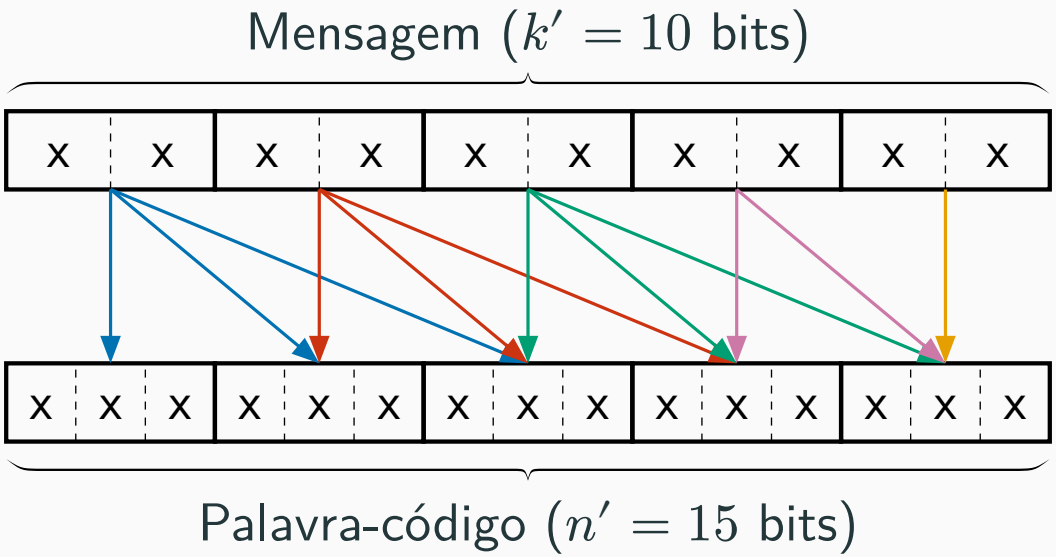
O codificador sempre inicia no estado zero e sua saída termina imediatamente após o último bloco de informação. O codificador *não necessariamente termina no estado zero*.



$$n' = hn \quad \text{e} \quad R' = \frac{k'}{n'} = \frac{hk}{hn} = R.$$

Terminação por truncamento direto

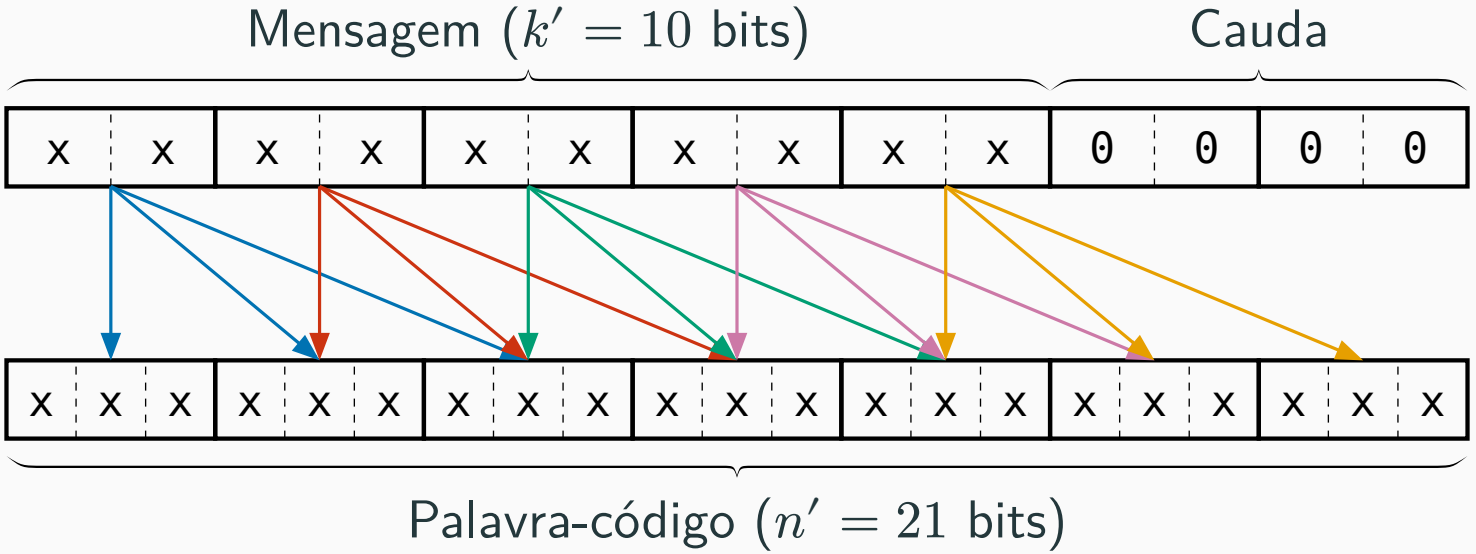
- *Problema:* Os bits do final da mensagem têm maior probabilidade de erro que os demais.



- *Motivo:* Os últimos blocos de informação influenciam menos blocos da palavra-código.

Terminação no estado zero

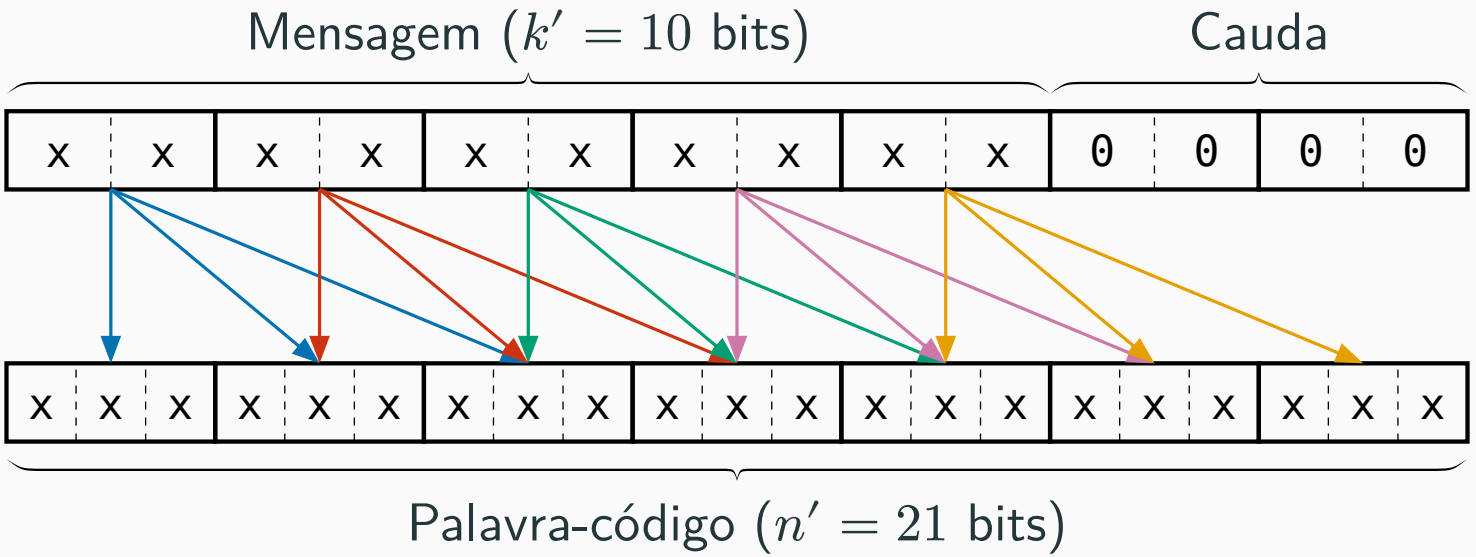
O codificador sempre *inicia e termina no estado zero*. Para isso, acrescenta-se uma **cauda** com μ blocos de zeros, que zera os registradores. Mensagem e cauda formam $h + \mu$ blocos.



$$n' = (h + \mu)n \quad \text{e} \quad R' = \frac{k'}{n'} = \frac{hk}{(h + \mu)n} = \frac{h}{h + \mu}R.$$

Terminação no estado zero

- Todo bloco de informação influencia $\mu + 1$ blocos da palavra-código: proteção uniforme.

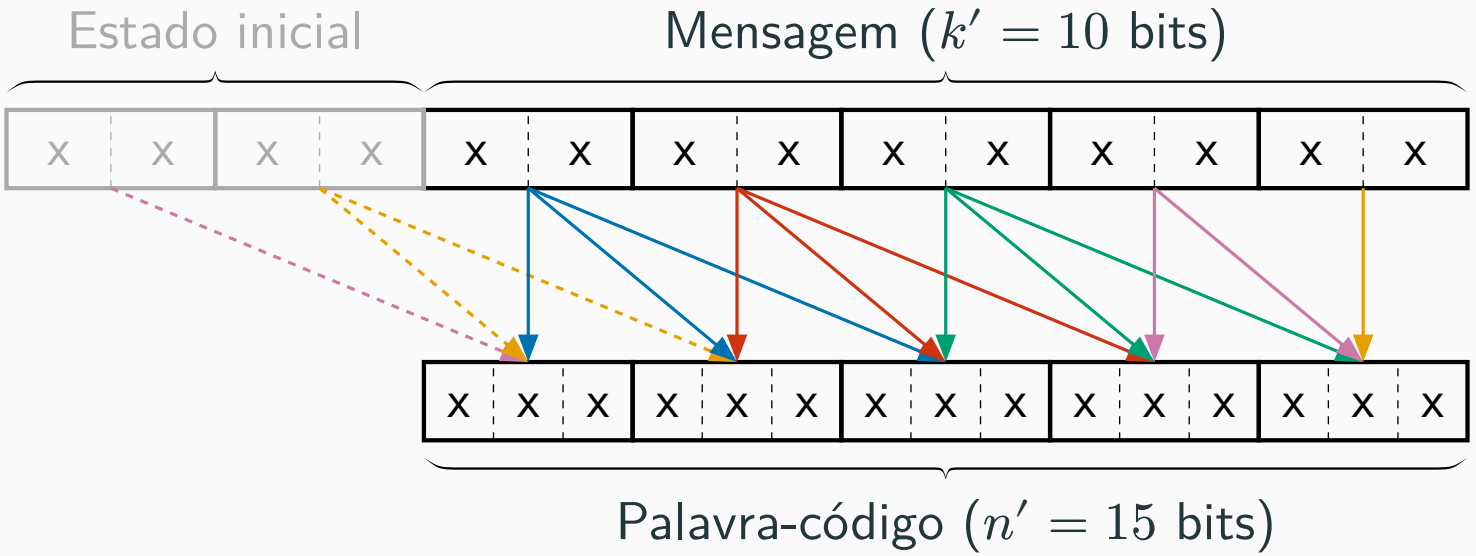


- A taxa diminui. No entanto, $R' \to R$ à medida que h cresce.

  Na figura, cai de $2/3 \approx 0.667$ para $10/21 \approx 0.476$. Para $h = 100$, vai a $100/153 \approx 0.653$.

Terminação por tail-biting

O codificador *inicia e termina no mesmo estado*, não necessariamente o zero. Para isso, antes de começar, carregam-se os registradores com os últimos μ blocos da mensagem. Ao final, eles conterão esses mesmos blocos.



Obtém-se proteção uniforme sem perda de taxa ($n' = hn$ e $R' = R$).

Terminação por tail-biting

- *Preço:* A decodificação fica mais complexa, pois o estado inicial depende da mensagem, a qual é desconhecida pelo decodificador.
- *Ressalva:* Para h pequeno, a distância mínima cai.

  No código $(2, 1, 2)$ do Exercício 2 com $h = 5$: tail-biting dá $d = 3$ e a terminação no estado zero dá $d = 5$; os dois só se igualam com $h \geq 9$.

Resumo

	Truncamento direto	**Estado zero**	**Tail-biting**
Estado inicial	zero	zero	últimos μ blocos
Estado final	qualquer	zero	igual ao inicial
Dimensão k'	hk	hk	hk
Comprimento n'	hn	$(h+\mu)n$	hn
Taxa R'/R	1	$h/(h+\mu)$	1
Proteção	desigual	uniforme	uniforme
Decodificação	simples	simples	mais complexa

Exemplos

Ambos os sistemas abaixo usam códigos com ordem de memória $\mu = 6$ (64 estados):

- No Wi-Fi, o "código da NASA" $(2, 1, 6)$ é terminado no estado zero. A cauda tem 6 bits, desprezível diante de um quadro com milhares de bits.
- No LTE (4G), os canais de controle usam tail-biting com um código $(3, 1, 6)$, que acrescenta uma terceira saída ao código da NASA. As mensagens têm algumas dezenas de bits, e uma cauda de 6 bits custaria caro.

Exercício 5

Considere novamente o código convolucional $(2, 1, 2)$ do Exercício 2, com $h = 5$ blocos de informação. Para cada uma das três técnicas de terminação:

(a) Determine os parâmetros (n', k') e a taxa R' do código de bloco resultante.

(b) Codifique a mensagem $u = \texttt{1 0 1 1 0}$.

(c) Codifique a mensagem $u = \texttt{0 0 0 0 1}$ e determine o peso de Hamming da palavra-código. Compare os resultados.

Exercício 5: Respostas

	Truncamento direto	**Estado zero**	**Tail-biting**
(a) (n', k')	$(10, 5)$	$(14, 5)$	$(10, 5)$
(a) R'	$1/2$	$5/14 \approx 0.357$	$1/2$
(b) v	11 10 00 01 01	11 10 00 01 01 11 00	00 10 00 01 01
(c) v	00 00 00 00 11	00 00 00 00 11 10 11	10 11 00 00 11
(c) $\mathrm{w_H}(v)$	2	5	5

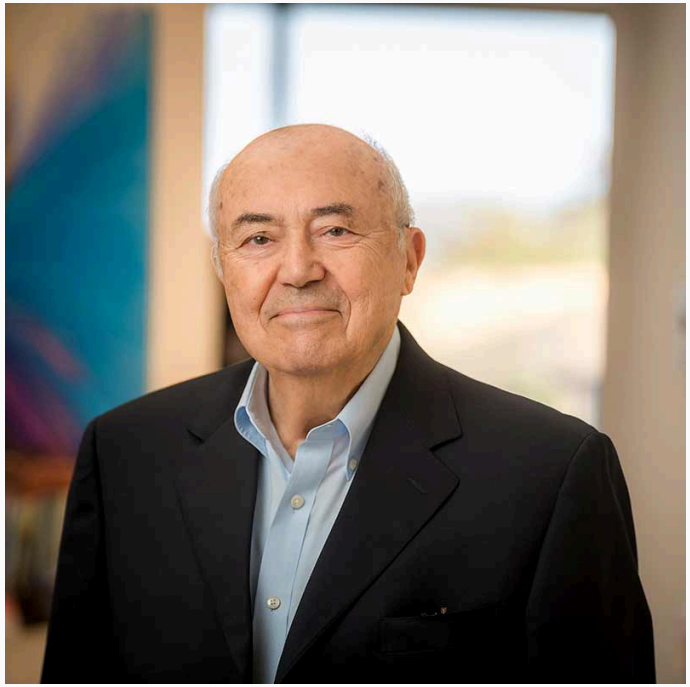
Observações:

- O truncamento produz o início da palavra-código do estado zero.
- O tail-biting produz a palavra-código do estado zero com a cauda somada aos primeiros blocos.

Algoritmo de Viterbi

Algoritmo de Viterbi

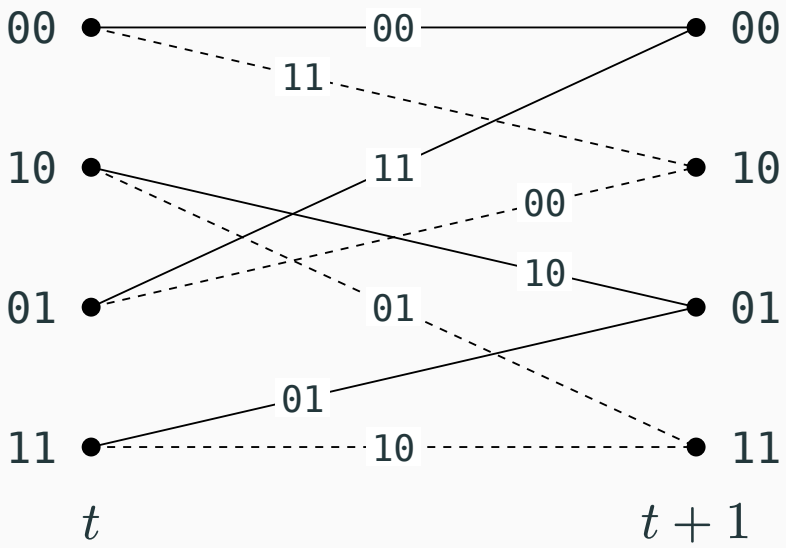
- Proposto por Andrew Viterbi em 1967.
- Decodifica códigos convolucionais por máxima verossimilhança, de forma eficiente.
- Viterbi foi cofundador da Qualcomm.



Andrew Viterbi (1935–).
Fonte: UC San Diego, The Chancellor's Medal.

Diagrama de treliça

O **diagrama de treliça** é o diagrama de estados "desdobrado no tempo". Cada coluna contém os 2^σ estados, e cada ramo liga um estado no instante t a outro no instante $t+1$.



No código $(2, 1, 2)$ do Exercício 2:

- Linha cheia: $u_t = 0$; linha tracejada: $u_t = 1$.
- Rótulo do ramo: saída v_t.
- Todas as seções são iguais a esta.

Diagrama de treliça

Com terminação no estado zero, todo caminho começa e termina em `00`. Cada palavra-código corresponde a um caminho na treliça.

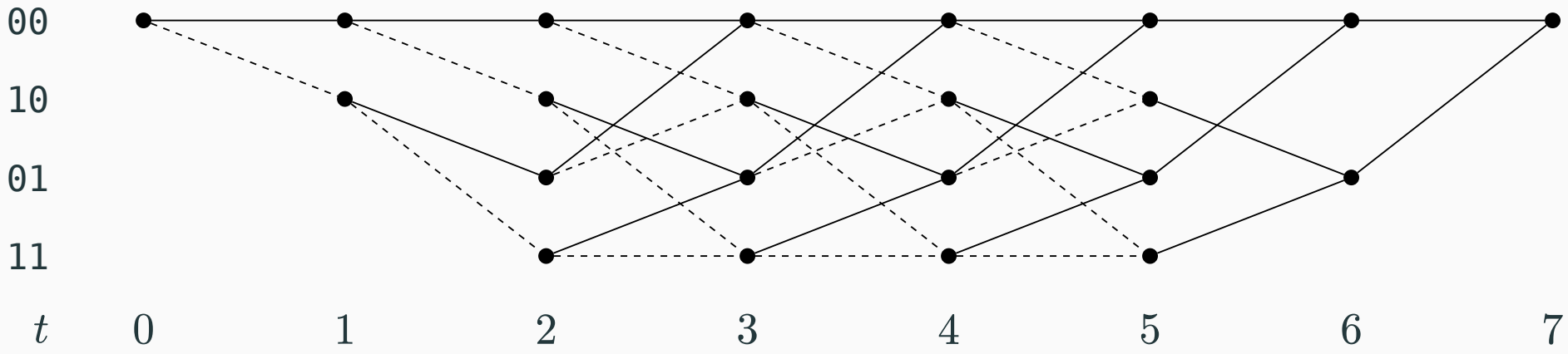
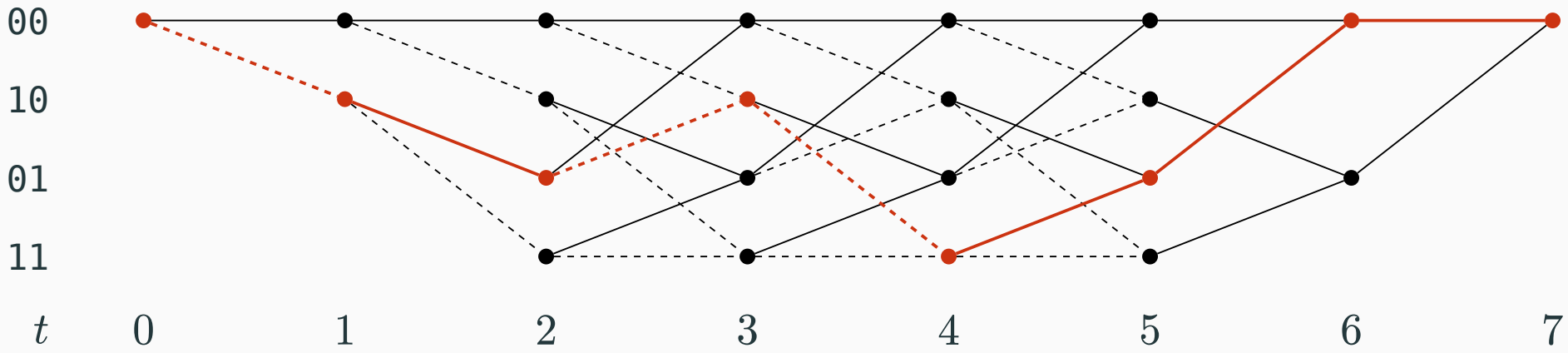


Diagrama de treliça

Com terminação no estado zero, todo caminho começa e termina em `00`. Cada palavra-código corresponde a um caminho na treliça.



Em destaque: $u = \texttt{1 0 1 1 0}$ e $v = \texttt{11 10 00 01 01 11 00}$ (Exercício 5).

Decodificação de máxima verossimilhança

Considere HDD no $\mathrm{BSC}(p)$, com $0 \leq p \leq 1/2$. O decodificador de máxima verossimilhança escolhe o caminho da treliça mais próximo de b:

$$\hat{v} = \operatorname*{argmin}_{v} \mathrm{d_H}(b, v).$$

- *Força bruta:* Comparar b com todas as $2^{k'}$ palavras-código. Inviável para k' grande.
- *Ideia:* A métrica é uma soma ao longo da treliça,

$$\mathrm{d_H}(b, v) = \sum_{t} \mathrm{d_H}(b_t, v_t).$$

Se dois caminhos chegam ao mesmo estado no mesmo instante, o de maior métrica pode ser descartado: qualquer continuação de um serve também ao outro.

Descrição do algoritmo

Cada estado s no instante t guarda uma **métrica acumulada** $M_t(s)$ e um **sobrevivente**: o melhor caminho que chega a ele.

1. *Inicialização:* $M_0(\texttt{00}) = 0$.
2. *Recursão:* para cada instante t e cada estado s,

$$M_{t+1}(s) = \min_{s' \to s} \{ M_t(s') + \mathrm{d_H}(b_t, v_t) \},$$

   onde v_t é a saída do ramo $s' \to s$. O sobrevivente de s estende o do s' vencedor.
3. *Decisão:* $\hat{v}$ é o sobrevivente do estado final de menor métrica.

Observações:

- Em caso de empate, escolhe-se qualquer um dos ramos.
- A complexidade é exponencial em σ, mas apenas linear em h (na força bruta, é exponencial em h).

Exercício 6

Considere novamente o código $(2, 1, 2)$ do Exercício 2, com terminação no estado zero e $h = 5$. Após a transmissão por um BSC, recebe-se

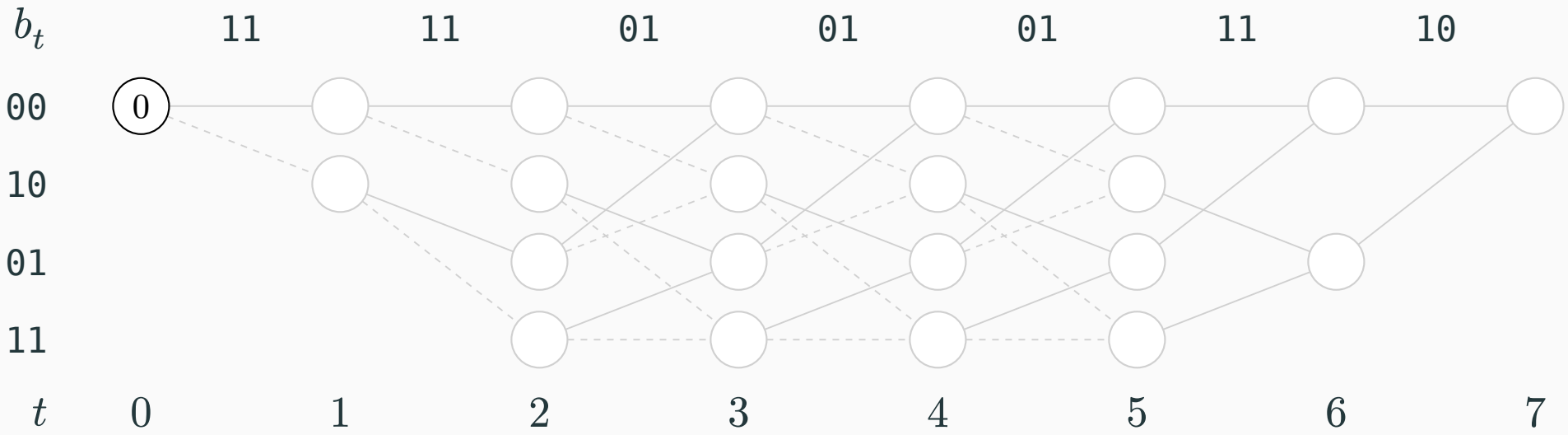
$$b = \texttt{11 11 01 01 01 11 10}.$$

(a) Utilize o algoritmo de Viterbi para determinar $\hat{v}$ e $\hat{u}$.

(b) Determine $\mathrm{d_H}(b, \hat{v})$ e compare $\hat{v}$ com as respostas do Exercício 5.

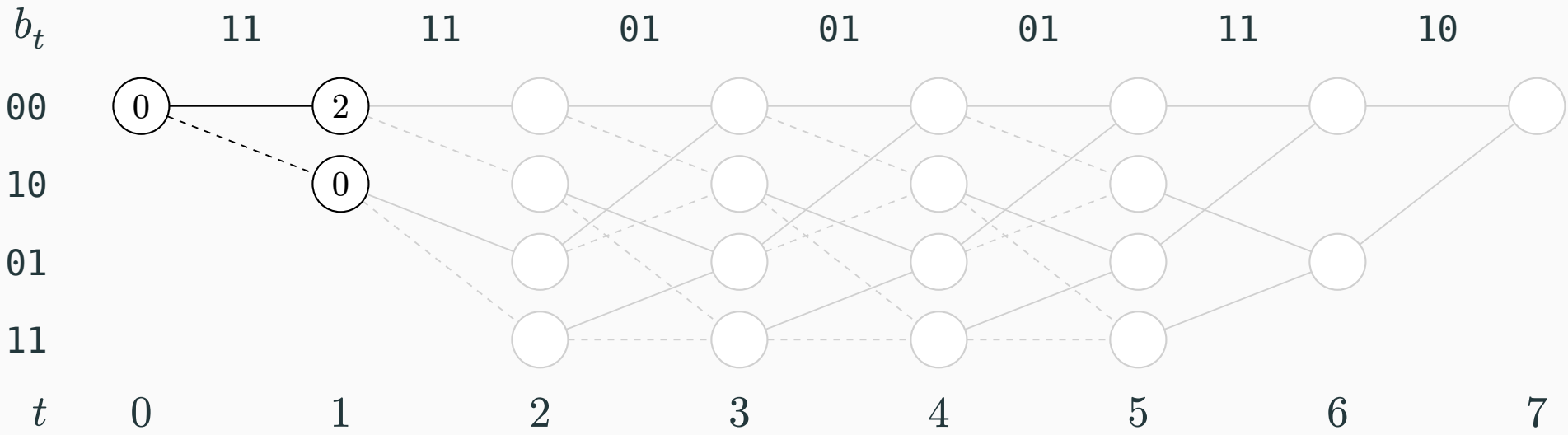
Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



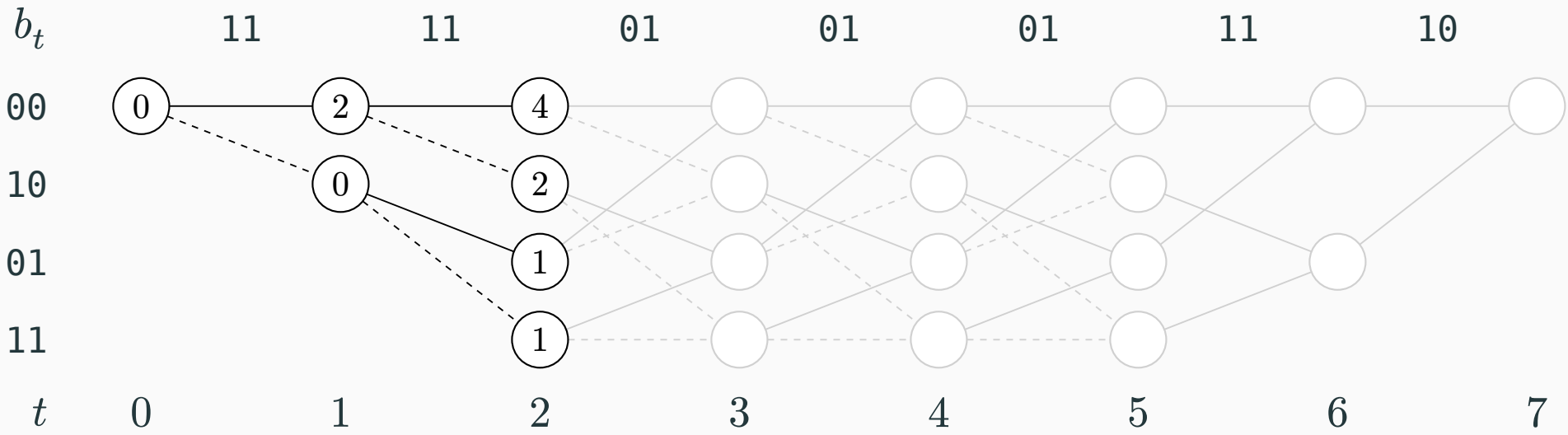
Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



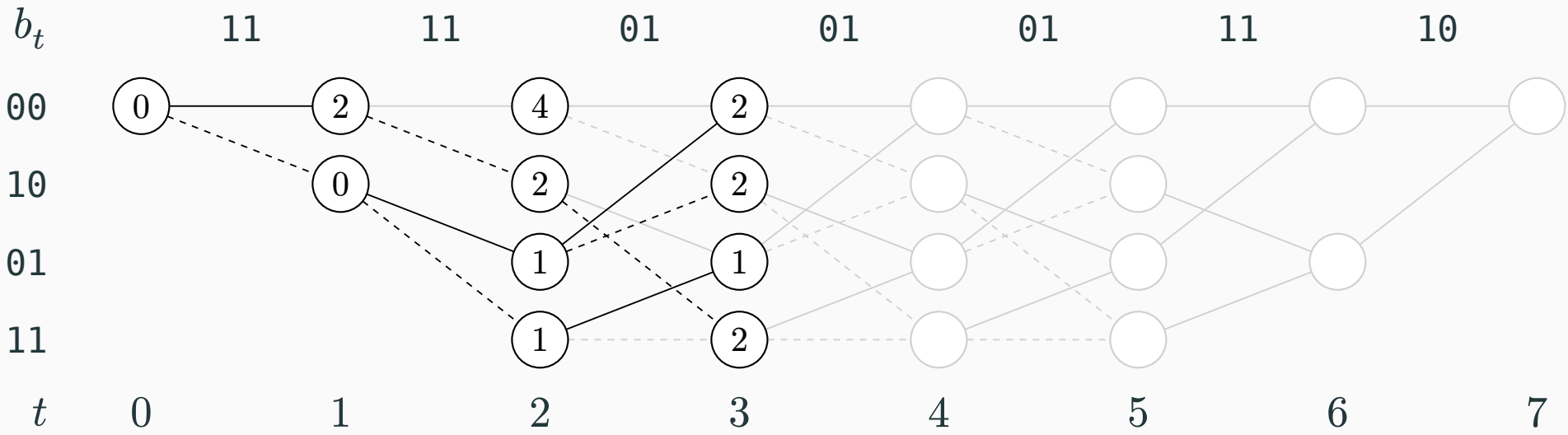
Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



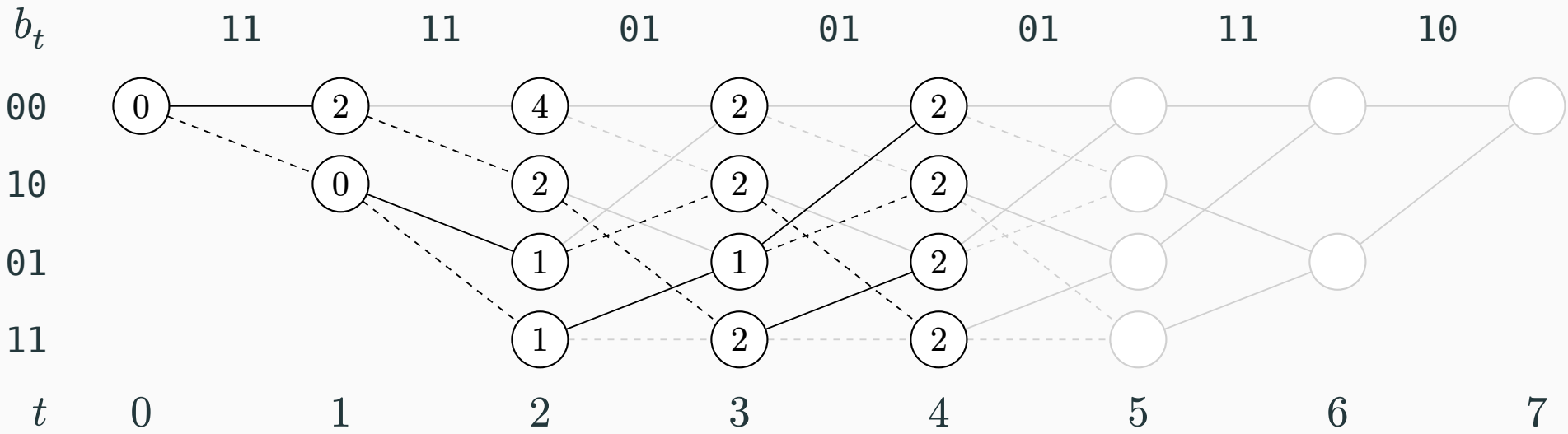
Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



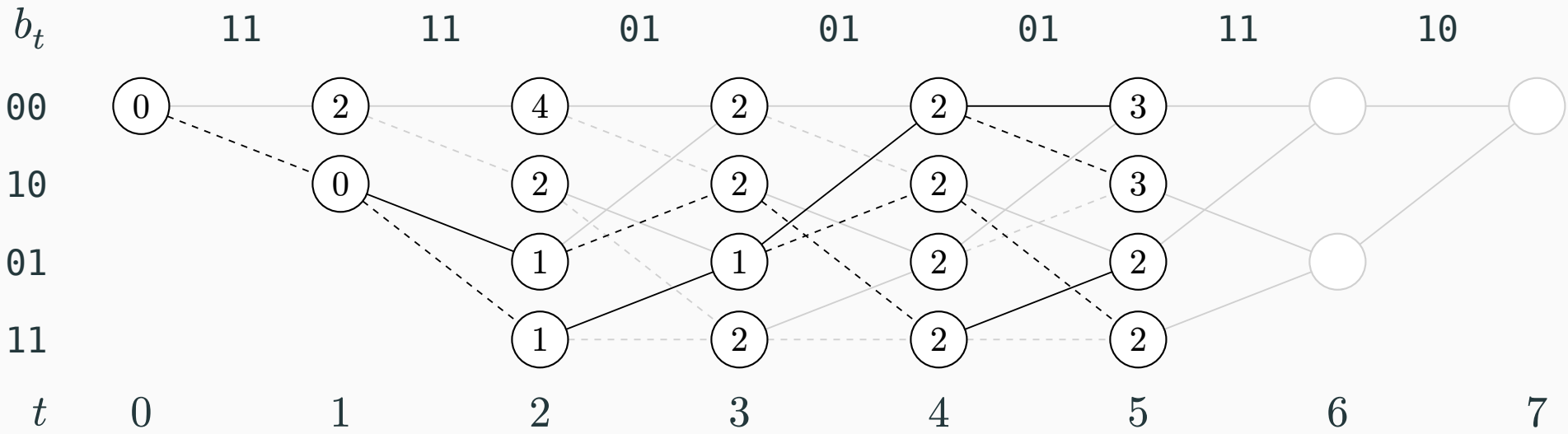
Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



Exercício 6: Respostas

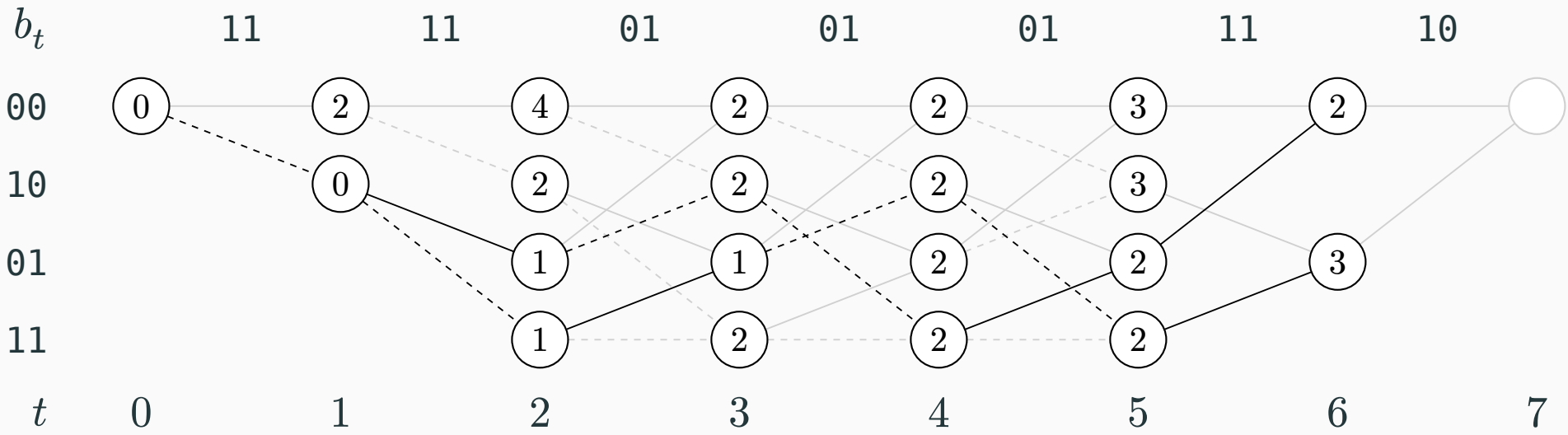
Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



Observação: Em $t = 5$, há empate nos estados `00` e `10`; os dois sobreviventes são descartados adiante.

Exercício 6: Respostas

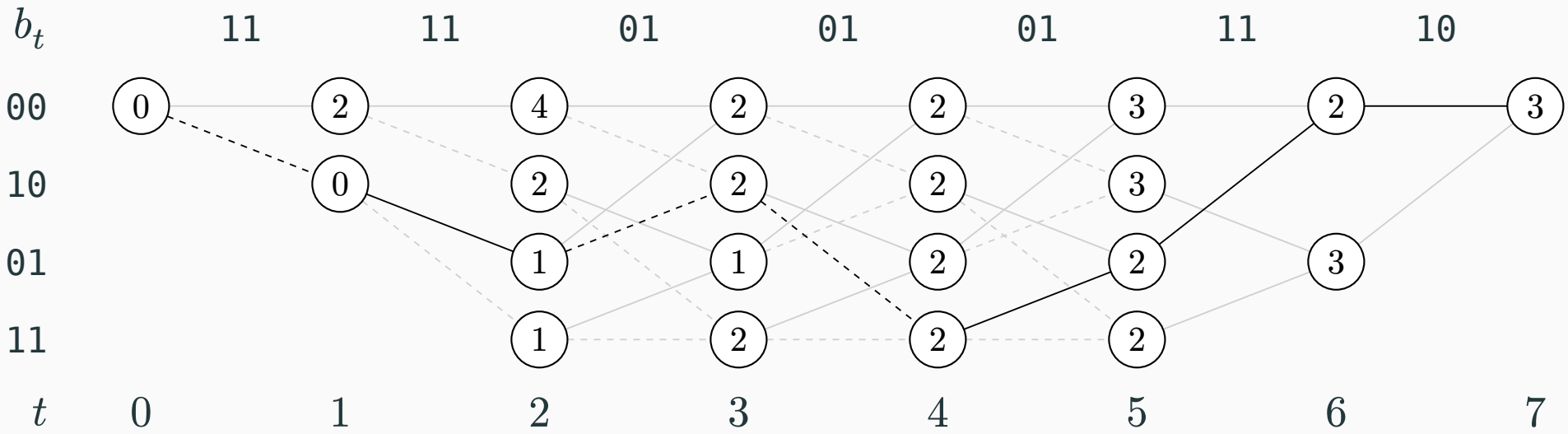
Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



Observação: Em $t = 5$, há empate nos estados `00` e `10`; os dois sobreviventes são descartados adiante.

Exercício 6: Respostas

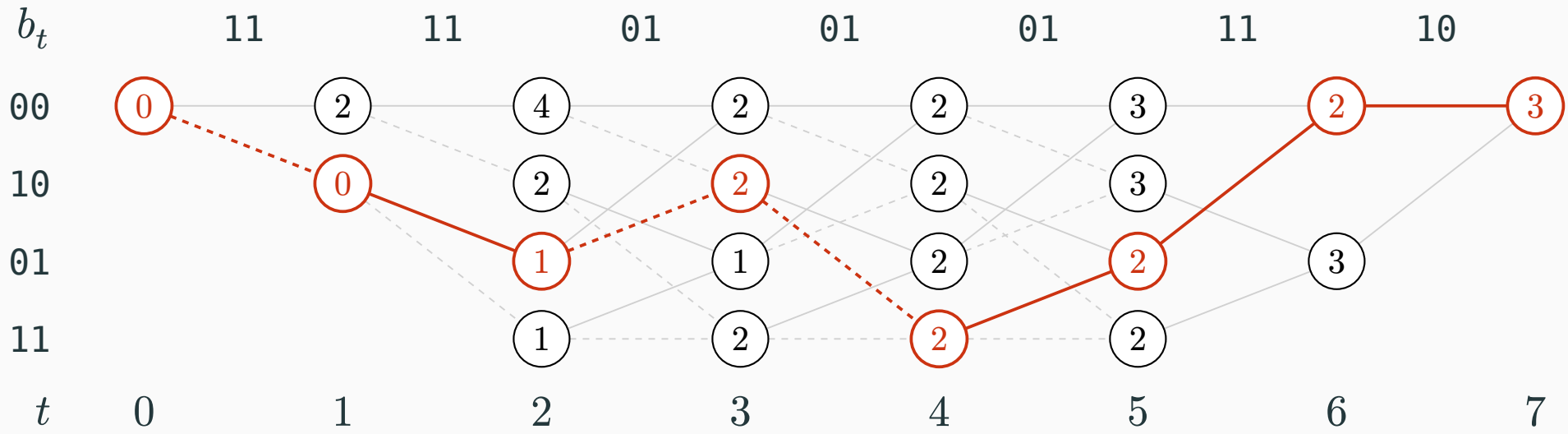
Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



Observação: Em $t = 5$, há empate nos estados `00` e `10`; os dois sobreviventes são descartados adiante.

Exercício 6: Respostas

Nos nós, as métricas acumuladas; em preto, os caminhos sobreviventes.



(a) $\hat{v} = \texttt{11 10 00 01 01 11 00}$ e $\hat{u} = \texttt{1 0 1 1 0}$.

(b) $\mathrm{d_H}(b, \hat{v}) = 3$, a métrica final.

Observação: Em $t = 5$, há empate nos estados `00` e `10`; os dois sobreviventes são descartados adiante.

Adaptação para SDD

O algoritmo é o mesmo; muda apenas a métrica de ramo.

- *HDD:* Distância de Hamming:

$$\mathrm{d_H}(b_t, v_t).$$

- *SDD no AWGN com BPSK:* Distância euclidiana quadrática:

$$\|r_t - x_t\|^2,$$

onde r_t é o vetor recebido e x_t é o sinal BPSK correspondente a v_t.

Alternativamente, como já visto antes, minimizar a norma $\|r_t - x_t\|^2$ equivale a maximizar a correlação $r_t \cdot x_t$.

Adaptação para SDD: Com valores L

Lembrando... Para bits equiprováveis, mapeamento $\texttt{0} \mapsto -1$ e $\texttt{1} \mapsto +1$ e AWGN com variância $N_0/2$, o valor L de cada bit recebido é

$$L = \ln \frac{P(v = \texttt{1} \mid r)}{P(v = \texttt{0} \mid r)} = \frac{4r}{N_0}.$$

O sinal de L é a decisão b; o módulo $|L|$, a confiabilidade dela.

Maximizar a correlação equivale a minimizar a soma das confiabilidades dos bits em que v_t discorda de b_t:

$$\sum_{j : v_t^{(j)} \neq b_t^{(j)}} \left| L_t^{(j)} \right|.$$

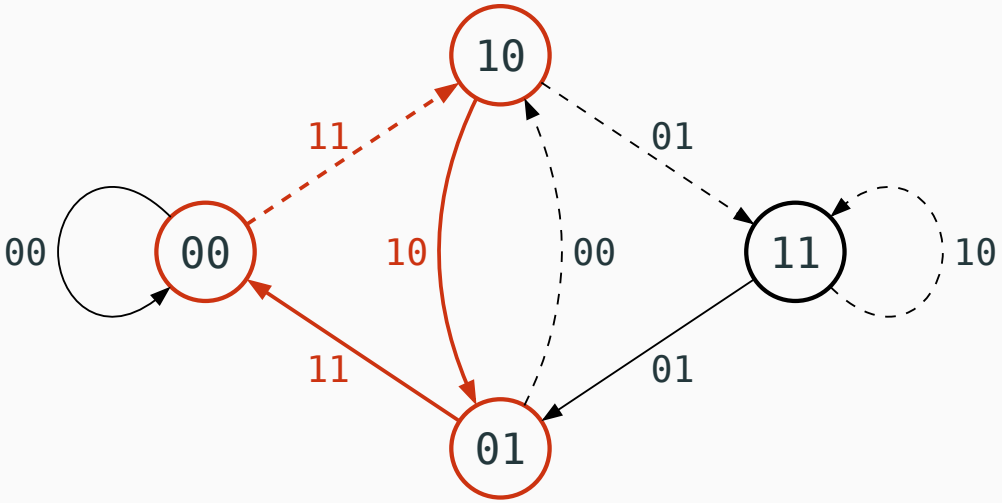
Observação: É uma distância de Hamming ponderada pela confiabilidade. Com todos os $|L|$ iguais, vira a distância de Hamming comum do HDD.

Desempenho de códigos convolucionais

Eventos de erro

Por linearidade, supõe-se transmitida a sequência toda nula. Um **evento de erro** ocorre quando o decodificador escolhe um caminho que sai de `00` e volta a `00`. Cada caminho tem:

- *Peso de saída:* distância de Hamming até a sequência toda nula.
- *Peso de entrada:* número de bits de informação errados.



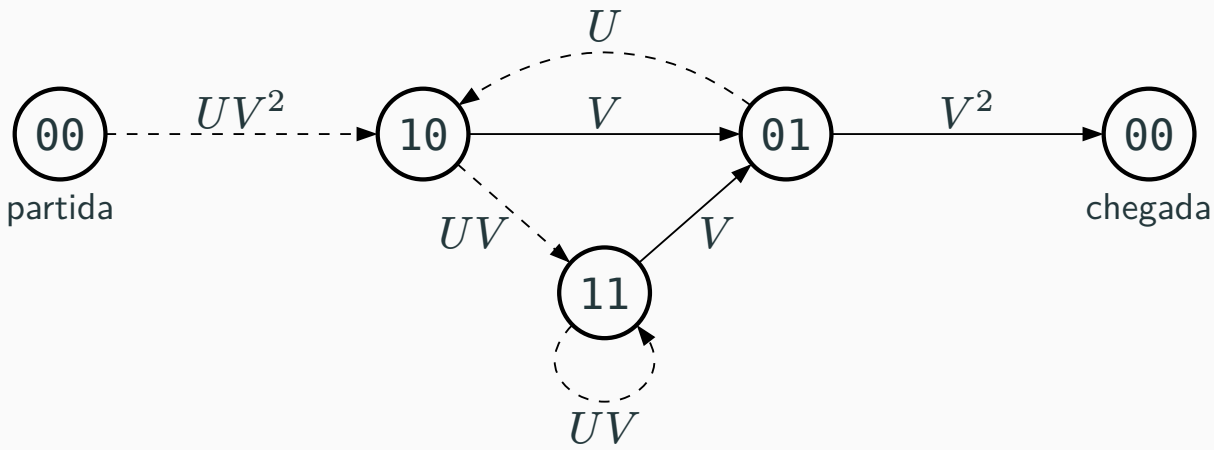
No código $(2, 1, 2)$ do Exercício 2:

- *Estados:* `00` $\to$ `10` $\to$ `01` $\to$ `00`.
- *Entrada:* `1 0 0` (peso 1).
- *Saída:* `11 10 11` (peso 5).

Diagrama de estados modificado

Para contar os caminhos que saem de `00` e voltam a `00`:

- Remove-se o loop de `00`, e esse estado é dividido em dois: partida e chegada.
- Cada ramo recebe o rótulo $U^{\mathrm{w_H}(u_t)} V^{\mathrm{w_H}(v_t)}$.

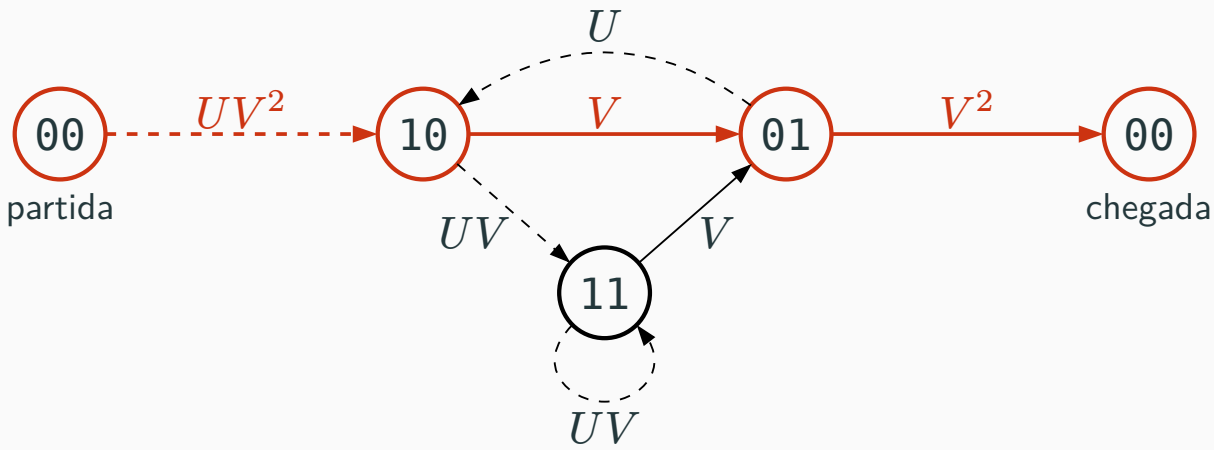


O rótulo de um caminho é o produto dos rótulos dos seus ramos.

Diagrama de estados modificado

Para contar os caminhos que saem de `00` e voltam a `00`:

- Remove-se o loop de `00`, e esse estado é dividido em dois: partida e chegada.
- Cada ramo recebe o rótulo $U^{\mathrm{w_H}(u_t)} V^{\mathrm{w_H}(v_t)}$.



O rótulo de um caminho é o produto dos rótulos dos seus ramos.

Exemplo: No caminho em destaque, $UV^2 \cdot V \cdot V^2 = UV^5$: peso de entrada 1, peso de saída 5.

Função enumeradora de caminhos

A **função enumeradora de caminhos** é a soma dos rótulos de todos os caminhos da partida até a chegada:

$$A(U, V) = \sum_{i,j} A_{i,j} U^i V^j,$$

onde $A_{i,j}$ é o número de caminhos com peso de entrada i e peso de saída j.

Observação: U e V só contam pesos, com aritmética usual (e não módulo 2).

Função enumeradora de caminhos

A **função enumeradora de caminhos** é a soma dos rótulos de todos os caminhos da partida até a chegada:

$$A(U,V) = \sum_{i,j} A_{i,j} U^i V^j,$$

onde $A_{i,j}$ é o número de caminhos com peso de entrada i e peso de saída j.

Observação: U e V só contam pesos, com aritmética usual (e não módulo 2).

Para obtê-la, resolvem-se as *equações de estado*:

$$X_s = \sum_{r \to s} \lambda_{rs} X_r, \quad \text{com} \quad X_{\text{partida}} = 1 \quad \text{e} \quad A(U,V) = X_{\text{chegada}},$$

onde X_s é a soma dos rótulos dos caminhos da partida até o estado s e λ_{rs} é o rótulo do ramo $r \to s$.

Exercício 7

Considere novamente o código $(2, 1, 2)$ do Exercício 2.

(a) Escreva as equações de estado.

(b) Resolva o sistema e determine a função enumeradora de caminhos $A(U, V)$.

Exercício 7

Considere novamente o código $(2, 1, 2)$ do Exercício 2.

(a) Escreva as equações de estado.

(b) Resolva o sistema e determine a função enumeradora de caminhos $A(U, V)$.

Respostas:

(a) Equações de estado:

$$
\begin{aligned}
X_{10} &= UV^2 + UX_{01} \\
X_{01} &= VX_{10} + VX_{11} \\
X_{11} &= UVX_{10} + UVX_{11} \\
A(U, V) &= V^2 X_{01}
\end{aligned}
$$

(b) Função enumeradora de caminhos:

$$
\begin{aligned}
A(U, V) &= \frac{UV^5}{1 - 2UV} \\
&= UV^5 + 2U^2V^6 + 4U^3V^7 + 8U^4V^8 + \cdots
\end{aligned}
$$

Distribuições de pesos

Definem-se A_j e B_j:

- *Número de caminhos de peso j:* $A_j = \sum_i A_{i,j}$.
- *Total de bits 1 de informação nos caminhos de peso j:* $B_j = \sum_i iA_{i,j}$.

Em termos de funções:

$$A(V) = \quad A(U,V)\Big|_{U=1} = \sum_j A_j V^j;$$

$$B(V) = \partial_U A(U,V)\Big|_{U=1} = \sum_j B_j V^j.$$

A **distância livre** d_{free} é o menor peso de saída de um caminho que sai de `00` e volta a `00`, ou seja, o menor j com $A_j \neq 0$.

Como o código é linear, é também a menor distância de Hamming entre duas sequências-código distintas.

Exercício 8

Considere novamente o código $(2, 1, 2)$ do Exercício 2. Determine $A(V)$, $B(V)$ e d_{free}.

Exercício 8

Considere novamente o código $(2, 1, 2)$ do Exercício 2. Determine $A(V)$, $B(V)$ e d_{free}.

Respostas:

$$A(V) = \frac{V^5}{1-2V} \quad = V^5 + 2V^6 + 4V^7 + 8V^8 + \cdots$$

$$B(V) = \frac{V^5}{(1-2V)^2} = V^5 + 4V^6 + 12V^7 + 32V^8 + \cdots$$

Distância livre: $d_{\text{free}} = 5$.

Limitante da união para a BER

É possível mostrar que a probabilidade de erro de bit de códigos convolucionais com SDD no AWGN utilizando BPSK é limitada superiormente por

$$P_{\mathrm{b}} \leq \frac{1}{k} \sum_{j} B_j \, \mathrm{Q}\Big(\sqrt{2jRE_{\mathrm{b}}/N_0}\Big).$$

O limitante se aproxima do valor verdadeiro para alta SNR.

Em alta SNR, domina o termo $j = d_{\mathrm{free}}$. Comparado à transmissão sem codificação, onde $P_{\mathrm{b}} = \mathrm{Q}\Big(\sqrt{2E_{\mathrm{b}}/N_0}\Big)$, o **ganho de codificação assintótico** para SDD é

$$\mathrm{ACG} = 10 \log_{10}(Rd_{\mathrm{free}}) \ \mathrm{dB}.$$

Observação: Nos códigos convolucionais, P_{B} (BLER) cresce com h; por isso, P_{b} (BER) é a métrica mais apropriada.

Exercício 9

Considere novamente o código $(2, 1, 2)$ do Exercício 2, com SDD e $E_{\mathrm{b}}/N_0 = 6$ dB.

(a) Estime P_{b} usando apenas três termos do limitante da união.

(b) Compare com a transmissão sem codificação.

(c) Determine o ganho de codificação assintótico.

Exercício 9

Considere novamente o código $(2, 1, 2)$ do Exercício 2, com SDD e $E_{\mathrm{b}}/N_0 = 6$ dB.

(a) Estime P_{b} usando apenas três termos do limitante da união.

(b) Compare com a transmissão sem codificação.

(c) Determine o ganho de codificação assintótico.

Respostas:

(a) $P_{\mathrm{b}} \approx 6.89 \times 10^{-6}$. (Com todos os termos: 7.28×10^{-6}.)

(b) Sem codificação, $P_{\mathrm{b}} \approx 2.39 \times 10^{-3}$, cerca de 350 vezes maior.

(c) $\mathrm{ACG} = 10 \log_{10}\left(\frac{1}{2} \cdot 5\right) \approx 3.98$ dB.

Observação: Sem codificação, obter $P_{\mathrm{b}} = 6.89 \times 10^{-6}$ exigiria $E_{\mathrm{b}}/N_0 \approx 9.75$ dB: o ganho efetivo é de 3.75 dB, um pouco abaixo do assintótico.

Referências

Referências

[1] S. Haykin, *Communication Systems*, 4th ed. John Wiley & Sons, 2001.

[2] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed. Pearson Prentice Hall, 2004.