Teoria da Informação (2002-2003)

Exame de recurso (21-7-03)

1 - $A = \{A_P, A_R\}$ $\qquad B = \{B_P, B_R\}$

A_P — mergulhador A pega em peça de Pyros

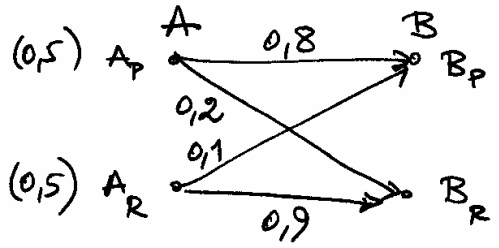
B_P — " B vê (ou pensa que vê) peça de Pyros

a)

$P(A_P) = P(A_R) = 0{,}5$

$P(B_P | A_R) = 10\%$ $\qquad P(B_R | A_P) = 20\%$

Diagrama do canal binário discreto s/ memória:



Pede-se a entropia da "fonte" A:

$$H(A) = \Omega(0{,}5) = 1 \text{ bit/peça}$$

b) O mergulhador B pensa ter visto uma peça de Pyros ($B = B_P$). O grau de incerteza é dado pela entropia condicional $H(A|B_P) = H[P(A_P|B_P), P(A_R|B_P)]$.

Do teorema de Bayes sabemos que

$$P(A_P, B_P) = P(A_P)\,P(B_P|A_P) = P(B_P)\,P(A_P|B_P)$$

$$\Rightarrow P(A_P|B_P) = \frac{P(A_P)\,P(B_P|A_P)}{P(B_P)}$$

Do diagrama tiramos $P(B_P|A_P) = 0{,}8$ e $P(A_P) = 0{,}5$.

Por outro lado $P(B_P) = 0{,}5 \times 0{,}8 + 0{,}5 \times 0{,}1 = 0{,}45$

Substituindo valores:

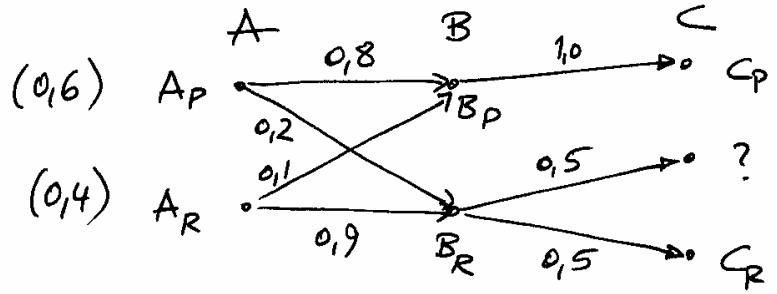
$$P(A_P \mid B_P) = \frac{0{,}8 \times 0{,}5}{0{,}45} = 8/9 \Longrightarrow P(A_R \mid B_P) = 1 - 8/9 = 1/9$$

Entropia pedida: $H(A \mid B_P) = H(8/9,\ 1/9) = \Omega(1/9) = 0{,}503$

c) Agora as probabilidades da fonte são outras:

$$P(A_P) = 0{,}6 \qquad P(A_R) = 0{,}4)$$

c1) Diagrama dos dois canais em série:



c2) $I(A,B) = H(A) - \underbrace{H(A \mid B)}_{\text{equivocação}} = H(B) - H(B \mid A)$

Agora é $P(B_P) = 0{,}6 \times 0{,}8 + 0{,}4 \times 0{,}1 = 0{,}52$

$H(B) = H(0{,}52\ ;\ 0{,}48) = \Omega(0{,}52) = 0{,}9988$

$H(B/A) = P(A_P)\, H(B \mid A_P) + P(A_R)\, H(B \mid A_R)$

$H(B \mid A_P) = \Omega(0{,}8) = 0{,}7219$

$H(B \mid A_R) = \Omega(0{,}1) = 0{,}4690$

$\Longrightarrow H(B \mid A) = 0{,}6 \times 0{,}7219 + 0{,}4 \times 0{,}4690 = 0{,}6208$

Finalmente, $I(A,B) = 0{,}9988 - 0{,}6208 = 0{,}3781$

c3) Matriz de probabil. de transição $A \to B$:

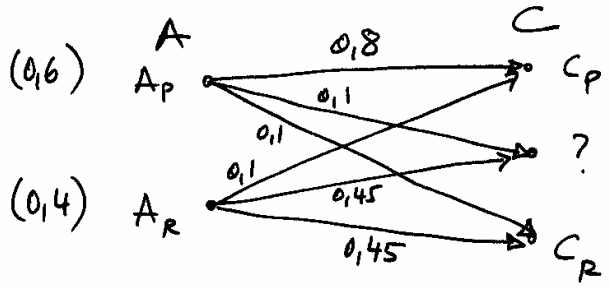
$$P_{AB} = \begin{bmatrix} 0{,}8 & 0{,}2 \\ 0{,}1 & 0{,}9 \end{bmatrix}$$

Matriz de probabil. de transição $B \to C$:

$$P_{BC} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0{,}5 & 0{,}5 \end{bmatrix}$$

Matriz do canal global $A \to C$:

$$P_{AC} = P_{AB}\, P_{BC} = \begin{bmatrix} 0{,}8 & 0{,}2 \\ 0{,}1 & 0{,}9 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0{,}5 & 0{,}5 \end{bmatrix} = \begin{bmatrix} 0{,}8 & 0{,}1 & 0{,}1 \\ 0{,}1 & 0{,}45 & 0{,}45 \end{bmatrix}$$



Probabilidade de o mergulhador C se "ver grego":

$$P(C = ?) = 0{,}6 \times 0{,}1 + 0{,}4 \times 0{,}45 = 0{,}24$$

2 — a) Codificação LZ78 c/ dicionário inicialm. vazio.

a/b/aa/c/ba/baa/ac/ab

Índice	Entrada do dicionário	Saída do codificador
1	a	(0, a)
2	b	(0, b)
3	aa	(1, a)
4	c	(0, c)
5	ba	(2, a)
6	baa	(5, a)
7	ac	(1, c)
8	ab	(1, b)

Logo, a sequência codificada é

(0,a) (0,b) (1,a) (0,c) (2,a) (5,a) (1,c) (1,b)

b) Descodificação LZ 78

Entrada do descodificador	Índice	Dicionário (= Saída do descodif.)
0 a	1	a
1 a	2	aa
0 b	3	b
0 c	4	c
1 b	5	ab
4 c	6	cc
3 c	7	bc
2 a	8	aaa

Saída descodificada: aaabcabccbcaaa.

c) Codificação LZ 77 de babcabcccaabcca

Corpo de janela: 7

"Look-ahead buffer": 4

Saída: apontador (i, j, <c>)

⇒ Saltos (deslocamentos) de j+1.

Passos de codificação:

1) ——— | babc | abcccaabcca — Deslocade 0+1=1 — Apontador (0,0,b)

nada no corpo — "buffer"

2) .....b | abca | bccc..... — 1 — (0,0,a)

3) ba | bcab | cccaa.... — 2 — (2,1,c)

4) babc | abcc | caabcca — 4 — (3,3,c)

5) abcabcc | caab | cca — 3 — (5,2,a)

6) abcccaa | bcca | — — 4 — (6,3,a)

7) caabcca | — | — (0,0,b) — Fim

Sequência de apontadores: (0,0,a) (2,1,c) (3,3,c) (5,2,a) (6,3,a)

Temos 6 apontadores. Cada apontador necessita de $\lceil \log_2 7 \rceil + \lceil \log_2 4 \rceil + \lceil \log_2 3 \rceil = 3+2+2 = 7$ bits.

Logo, precisaremos de $6 \times 7 = 42$ bits para representar a sequência ternária dada.

5- $Y = [y_1\ y_2 \cdots y_7]$

$$\left.\begin{array}{l} y_1 = u_1 \\ y_2 = u_2 \\ y_3 = u_3 \end{array}\right\} \text{mensagem} \qquad \left.\begin{array}{l} y_4 = u_1 + u_3 \\ y_5 = u_1 + u_2 + u_3 \\ y_6 = u_2 + u_3 \\ y_7 = u_1 + u_3 \end{array}\right\} \text{bits de paridade}$$

$\Rightarrow$ Trata-se de um código $(7,3)$ em que

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

$$S = ZH = [z_1\ z_2 \cdots z_7]\, H = [s_1\ \ s_2\ \ s_3\ \ s_4]$$

$$\xRightarrow[\text{(Inspecção de } H)]{} \left\{\begin{array}{l} s_1 = z_1 + z_3 + z_4 \\ s_2 = z_1 + z_2 + z_3 + z_5 \\ s_3 = z_2 + z_3 + z_6 \\ s_4 = z_1 + z_3 + z_7 \end{array}\right. \qquad \text{c.q.d.}$$

Resolução alternativa (sem recorrer à matriz H):

Das equações dos bits de paridade temos que

$$\left\{\begin{array}{l} y_1 + y_3 + y_4 = 0 \\ y_1 + y_2 + y_3 + y_5 = 0 \\ y_2 + y_3 + y_6 = 0 \\ y_1 + y_3 + y_7 = 0 \end{array}\right.$$

Se a palavra recebida $Z=[z_1 z_2 \dots z_7]$ não contiver erros é igual à palavra transmitida $Y=[y_1 y_2 \dots y_7]$, ou seja, numa palavra recebida s/ erros verifica-se que

$$\begin{cases} z_1+z_3+z_4=0 \\ z_1+z_2+z_3+z_5=0 \\ z_2+z_3+z_6=0 \\ z_1+z_3+z_7=0 \end{cases}$$

Se algum dos bits de Z estiver errado uma ou mais destas equações não é satisfeita, isto é, se alguma das somas em módulo 2 for diferente de zero significa que a palavra Z contém erro(s) $\Rightarrow$ cada uma das somas é um dos 4 elementos da síndrome S:

$$\begin{cases} s_1 = z_1+z_3+z_4 \\ s_2 = z_1+z_2+z_3+z_5 \\ s_3 = z_2+z_3+z_6 \\ s_4 = z_1+z_3+z_7 \end{cases}$$

Quer dizer que o circuito de geração de bits de paridade do codificador pode ser usado no descodificador para se obter a síndrome!

4 – $\sum_{i=1}^{10} i\, u_i = 0 \pmod{11} \equiv \underbrace{(u_1+2u_2+3u_3+\dots+10u_{10})}_{S} \bmod 11 = 0$

Ou seja, $S \bmod 11 = 0$

a) Vamos trocar o dígito u_k pelo dígito u_m:

$$S_e = u_1+2u_2+\dots+k\,u_m+\dots+m\,u_k+\dots+10\,u_{10} =$$

$$= \underbrace{\sum_i i\,u_i}_{S} + (k-m)\,u_m + (m-k)\,u_k = S + (k-m)(u_m-u_k)$$

Dividindo por 11 e calculando o resto:

$$S_e \bmod 11 = \left[S + (k-m)(u_m - u_k)\right] \bmod 11 = \underbrace{S \bmod 11}_{0} + \dots$$

A troca é detectada se $(k-m)(u_m-u_k) \bmod 11 \neq 0$.

Temos duas situações:

1) Dígitos trocados são consecutivos $\Rightarrow |k-m| = 1$. Como

$$\underset{(\text{isto é, } < 11)}{0 < |u_m - u_k| \leq 10} \Rightarrow \underbrace{(k-m)}_{\pm 1}(u_m-u_k) \bmod 11 = \pm \underbrace{(u_m-u_k)}_{<11} \neq 0$$

2) Dígitos trocados não são consecutivos $\Rightarrow |k-m| \geq 2$.

Mas como $0 < |u_m - u_k| \leq 10$ e $0 < |k-m| \leq 9$ (ou seja, estamos a incluir a situação 1)

$\Rightarrow$ o produto $(k-m)(u_m-u_k)$ nunca pode ser múltiplo de 11, que é primo.

$\Rightarrow$ Qualquer troca de dois dígitos quaisquer é detectada.

b) 1) Um único engano é sempre detectável:

$$S_1 = u_1 + 2u_2 + \dots + k\underset{\substack{\uparrow \\ \text{dígito errado}}}{u_r} + \dots = S + k(u_r - u_k)$$

$$S_1 \bmod 11 = \underbrace{S \bmod 11}_{0} + \underbrace{k(u_r-u_k)}_{\text{nenhum destes factores é } 11} \bmod 11$$

$\Rightarrow$ resto $\neq 0 \Rightarrow$ detectável.

2) Dois enganos podem não ser detectáveis:

$$S_2 = u_1 + 2u_2 + \dots + k u_r + \dots + m u_s + \dots = S + k(u_r - u_k) + m(u_s - u_m)$$

$$S_2 \bmod 11 = \underbrace{S \bmod 11}_{0} + \left[k\underbrace{(u_r-u_k)}_{\leq 10} + m\underbrace{(u_s-u_m)}_{\leq 10}\right] \bmod 11$$

A deticção dos dois enganos pode não estar garantida pois o que está dentro do parêntesis recto pode ser múltiplo de 11, como se vê no seguinte exemplo:

$$\underset{(ex.:\ 3\times 5)}{k(u_r-u_k)=15} \Longrightarrow 15 \bmod 11 = 4$$

$$\underset{(ex.:\ 6\times 3)}{m(u_s-u_m)=18} \Longrightarrow 18 \bmod 11 = 7$$

$$\overset{+}{\overline{11}}$$

$$(15+18) \bmod 11 = (15 \bmod 11 + 18 \bmod 11) \bmod 11 = (4+7) \bmod 11 =$$
$$= 11 \bmod 11 = 0$$

ou

$$(15+18) \bmod 11 = 33 \bmod 11 = 0$$

3) Duas trocas e um engano podem não ser detectáveis:

u_k e u_m trocados; u_r — engano

$$S_3 = u_1 + 2u_2 + \dots + k u_m + \dots + m u_k + \dots + r u_r + \dots =$$
$$= S + [(k-m)(u_m-u_k) + r(u_r-u_n)]$$

Os erros são detectáveis se $[\dots + \dots] \bmod 11 = 0$

O primeiro resto é $\neq 0$ $\left([(k-m)(u_m-u_k)] \bmod 11 \neq 0\right)$ e o segundo também. Os erros não são detectáveis se a soma dos dois restos der 11, o que pode acontecer.

Exemplo:

Se $(k-m)(u_m-u_k)=15 \Longrightarrow$ resto 4

$r(u_r-u_n) = 18 \Longrightarrow$ resto 7

$\overset{+}{\overline{}}$

$11 \rightarrow$ resto 0.

5 — a) $(15,10) \Rightarrow$ Singleton : $d_{min} \leq \underbrace{n-k+1}_{6} = d_s$

Plotkin: $d_{min} \leq \dfrac{n \cdot 2^{k-1}}{2^k - 1} = 15 \dfrac{2^9}{2^{10}-1} =$

$= 7{,}51 \Rightarrow 7 = d_p$

Assim, $d_{min} = \min(d_s, d_p) = \min(6,7) - 1 = 5$

$d_{min} \geq 2t+1 \Rightarrow t = \left\lfloor \dfrac{d_{min}-1}{2} \right\rfloor = 2$

$d_{min} \geq l+1 \Rightarrow l = \lfloor d_{min} - 1 \rfloor = 4$

b) ARQ SR: $R'_{SR} = \dfrac{k}{n}(1-p_R) \approx \dfrac{k}{n}(1-np) = \dfrac{10}{15}\underbrace{(1-0{,}15 \cdot 10^{-2})}_{\approx 1}$

$r_b = 100\,\text{kbits/s} \Rightarrow T_b = \dfrac{1}{r_b} = 10^{-5}\,s$

Seja $n_b = 10^6$ bits $\Rightarrow$ duração de 10^6 bits : $n_b T_b = 10^6 \cdot 10^{-5} =$
$= 10\,s$

Com retransmissões demora mais tempo:

$$T_{total} = \frac{n_b T_b}{R'_{SR}} = \frac{\cancel{10} \times 15}{\cancel{10}(1-0{,}15 \cdot 10^{-2})} \approx 15 \text{ segundos.}$$

6 — a) Somando a 2ª e a 3ª palavras obtemos

$$\begin{array}{r} 0111010101 \\ + \quad 0100101101 \\ \hline 0011111000 \end{array}$$

← 2ª linha da matriz G

← 3ª linha

A mensagem 101 é obtida somando a 1ª palavra dada c/ a palavra-soma obtida atrás:

$$\begin{array}{r} 1001010101 \\ + \quad 0011111000 \\ \hline 101\,010110\,1 \end{array}$$

b) Matriz geradora do código (10, 3):

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

$\Downarrow$ código encurtado (9,2)

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

c) Matriz H do código original:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline & & & I_7 & & & \end{bmatrix}$$

$\Longrightarrow$ a soma das três primeiras linhas é nula $\Longrightarrow d_{min} = 3$.

Quanto ao código encurtado basta consultar as suas quatro palavras de código:

0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 0 0 0
1 0 1 0 1 0 1 0 1 1
1 1 0 1 0 1 1 0 1

$\Rightarrow d_{min} = 5$

d) (10,3): Precisamos de um polinómio de grau $10-3=7$ que seja factor de $p^{10}+1$. Consultando a tabela de factorização vemos que $p^{10}+1 = (p+1)^2(p^4+p^3+p^2+p+1)^2$. Ora não é possível obter um polinómio de grau 7 (nem 3, aliás) com estes factores!

Conclusão: não existe código cíclico (10,3)!!