COMBINATÓRIA E TEORIA DE CÓDIGOS
Ficha 1

11/3/2009

Exercícios 1.2 - 1.5 (R. Hill)

Problema 1. (Generalização de 1.2) Quais as capacidades de correcção e detecção *simultâneas* de erros de um código de distância mínima d. Discuta e comente separadamente os casos de d par e ímpar. Dê exemplos ilustrativos.

Problema 2. (Generalização de 1.5) Estabeleça e demonstre um majorante para o número máximo de palavras M de um código $q-$ário (n, M, d). Dê exemplos ilustrativos.

Problema 3. O que se poderá fazer e dizer quanto às capacidades correctoras de erros de apagamento e de erros de troca e de apagamento simultaneamente de um código de distância mínima d. Estabeleça conjecturas e Teoremas, experimente-os e demonstre-os. Dê exemplos ilustrativos.

Problema 4. Considere um canal binário simétrico com probabilidades de troca de símbolos

$$P\{recebido\mathbf{1}|enviado\mathbf{0}\} = 0,3 \qquad ; \qquad P\{recebido\mathbf{0}|enviado\mathbf{1}\} = 0,2.$$

Se for usado o código binário $\{000, 100, 111\}$ para enviar uma mensagem através desse canal, descodifique, usando a máxima verosimilhança, as palavras recebidas:

a) 010 ; b) 011 ; c) 001 .

Problema 5. Na palavra binária

$$0111100000?001110000?00110011001010111000000000?01110$$

codificou-se uma data. O sistema utilizado consistiu em escrevê-la primeiro na forma de 6 dígitos decimais seguidos (por exemplo, 290296 quer dizer 29 de Fevereiro de 1996) e passar esse número para a base 2 (no exemplo acima 290296 transforma-se em 1000110110111111000) e em seguida codificar de acordo com a regra

$$\{0,1\}^2 \longrightarrow \mathcal{C} \subseteq \{0,1\}^6$$

$$00 \longrightarrow 000000$$

$$01 \longrightarrow 001110$$

$$10 \longrightarrow 111000$$

$$11 \longrightarrow 110011$$

Na palavra recebida há 3 bits que não se conhecem (foram apagados) e possivelmente outros que estão trocados.

a) Encontrar os 3 bits apagados;
b) Dizer quantos bits e em que posições estão errados;
c) De que data se trata?
d) Repetir o problema trocando os bits das posições 15 e 16.

Problema 6. (Um Código de HAMMING) Codifica-se um *vector mensagem* de 4 componentes binárias $\mathbf{m} = m_1 m_2 m_3 m_4$, $m_i \in \{0,1\}$ numa *palavra de código* com 7 componentes binárias $\mathbf{c} = c_1 c_2 c_3 c_4 c_5 c_6 c_7$, $c_j \in \{0,1\}$, definidas por

$$c_3 = m_1 \; ; \; c_5 = m_2 \; ; \; c_6 = m_3 \; ; \; c_7 = m4$$

e as restantes componentes escolhidas

$$c_4: \text{ tal que } \alpha = c_4 + c_5 + c_6 + c_7 \text{ seja par}$$

$$c_2: \text{ tal que } \beta = c_2 + c_3 + c_6 + c_7 \text{ seja par}$$

$$c_1: \text{ tal que } \gamma = c_1 + c_3 + c_5 + c_7 \text{ seja par.}$$

Verifique que com este esquema de codificação se constrói um código que permite corrigir um erro em qualquer posição.

Recebido um vector $\mathbf{x} = x_1x_2x_3x_4x_5x_6x_7$ calculam-se

$$\left.\begin{array}{l} \alpha = x_4 + x_5 + x_6 + x_7 \\ \beta = x_2 + x_3 + x_6 + x_7 \\ \gamma = x_1 + x_3 + x_5 + x_7 \end{array}\right\} \mathrm{mod} 2;$$

$\alpha\beta\gamma$ representa em binário a componente j onde se deu o erro. Se $\alpha\beta\gamma = 000$ assume-se que não há erro.

Estude este exemplo com cuidado.