Teoria dos Códigos

Uma breve introdução

1 – Exemplos e definição

2- Canal de transmissão

3-Descodificação

4-Correção e detecção de erros

Primeiros exemplos e definições

Consideremos a seguinte situação: fulano X está perdido no meio de uma floresta mas está em contacto com fulano Y que consegue saber onde está X e qual o caminho que este deve tomar. A mensagem que Y gostaria de transmitir a X consiste numa sequência dos símbolos N (Norte), S (Sul), E (Este) e W (Oeste), no entanto o *canal de transmissão* entre Y e X apenas permite usar dois símbolos.

Trata-se portanto de *codificar* os quatro pontos cardeais através de um *código binário*.

Podemos escolher vários tipos de código.

Exemplo 1 Seja $C_1 = \{0, 1, 00, 11\}$ e consideremos a correspondência

$$N \mapsto 0 \qquad S \mapsto 1 \qquad E \mapsto 00 \qquad W \mapsto 11$$

O conjunto C_1 diz-se um *código* binário (em dois símbolos) e a aplicação entre $\{N, S, E, W\}$ e C_1 definida por diz-se uma *função de codificação*.

Neste exemplo o código não é *unicamente decifrável* pois a mensagem 00 tanto pode significar NN ou E.

Exemplo 2 Consideremos agora o código $C_2 = \{0, 01, 011, 0111\}$ e a correspondência

$$N \mapsto 0 \qquad S \mapsto 01 \qquad E \mapsto 011 \qquad W \mapsto 0111$$

Neste caso o código é unicamente decifrável, mas não é *instantâneo* pois é preciso esperar pela próxima palavra, ou pelo fim da mensagem, para se conseguir interpretar cada palavra.

Exemplo 3 Consideremos ainda um terceiro código $C_3 = \{0, 10, 110, 1110\}$ e a correspondência

$$N \mapsto 0 \qquad S \mapsto 10 \qquad E \mapsto 110 \qquad W \mapsto 1110$$

Neste caso o código é unicamente decifrável e instantâneo – uma palavra acaba quando se recebe o símbolo 0.

Exemplo 4 Consideremos ainda um quarto código $C_4 = \{00, 01, 10, 11\}$ e a correspondência

$$N \mapsto 00 \qquad S \mapsto 01 \qquad E \mapsto 10 \qquad W \mapsto 11$$

Trata-se de um código unicamente decifrável e instantâneo, pois todas as palavras têm o mesmo comprimento. Neste caso C_4 diz-se um *código uniforme*.

Exemplo 5 Para finalizar estes exemplos, consideremos o código $C_5 = \{000, 011, 101, 110\}$ e a correspondência

$$N \mapsto 000 \qquad S \mapsto 011 \qquad E \mapsto 101 \qquad W \mapsto 110$$

Tal como no exemplo anterior, C_5 é um código uniforme.

Nesta aulas iremos considerar apenas códigos uniformes, como os dos Exemplos 4 e 5. Entre estes, qual o melhor código, C_4 ou C_5?

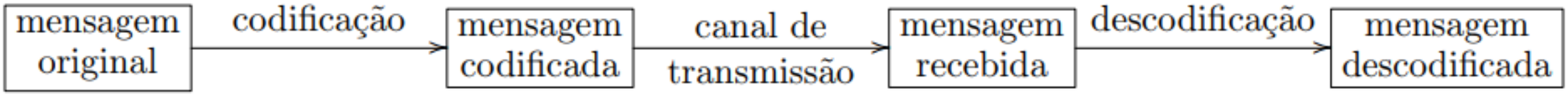
A resposta depende naturalmente do sentido que se der a "melhor".

Mas, mesmo sem especificar esse sentido, já podemos comparar C_4 e C_5 nos seguintes aspectos:

- C_4 é um código de comprimento menor do que C_5, portanto é mais rápido transmitir uma mensagem usando C_4.

- C_4 é o conjunto de todas as palavras binárias de comprimento 2 (i.e., $C_4 = (\mathbb{Z}_2)^2$), portanto qualquer palavra recebida é uma palavra de código e, por isso, C_4 não permite detectar erros que ocorram durante a transmissão. Por outro lado, $C_5 \neq (\mathbb{Z}_2)^3$ e portanto C_5 vai permitir detectar alguns erros. Mas será possível corrigi-los?

A situação geral considerada em Teoria de Códigos pode ser esquematizada na seguinte figura:



As mensagens codificada e recebida são ambas formadas por sequências de símbolos do mesmo alfabeto. O canal de transmissão poderá ter ruído, de modo que a mensagem recebida poderá conter erros ou símbolos apagados e não será igual à mensagem enviada. O objectivo é estudar códigos tendo em conta certas características como a rapidez de transmissão, facilidade e eficiência de codificar e descodificar, capacidades detectoras e correctoras de erros, etc.

Elementos básicos para construção de um código

- Um conjunto finito, $\mathcal{A}$ que chamaremos **alfabeto**. Denotaremos por $q = |\mathcal{A}|$ o número de elementos de A. Quando o número de elementos do alfabeto de um código é q, diz-se que o código é q-*ário*. Nos exemplos da seção anterior vimos códigos cujo alfabeto era o conjunto $\mathbb{Z}_2 = \{0, 1\}$, que são os chamados códigos *binários*.

- Seqüências finitas de símbolos do alfabeto, que chamaremos **palavras**. O número de letras de uma palavra chama-se o seu **comprimento**. Para termos um código com o qual seja fácil trabalhar com um certo rigor, faremos a convenção de que todas as palavras que iremos considerar para compor o código terão o mesmo comprimento n. Por esta razão, estes códigos dizem-se *em blocos* mas, como todos os códigos que estudaremos serão em blocos, daqui em diante omitiremos esta palavra.

- Um **código q-ário de comprimento** n será então um subconjunto qualquer (a nossa escolha) de palavras de comprimento n, i.e., um código $\mathcal{C}$ é um subconjunto

$$\mathcal{C} \subset \mathcal{A}^n = \underbrace{A \times A \times \cdots \times A}_{n \text{ vezes}}.$$

Exemplo 1 *Quando o alfabeto utilizado é o conjunto* $\mathbb{Z}_2 = \{0, 1\}$ *o código diz-se binário. O conjunto*

$$\mathcal{C}_1 = \{00000, 01011, 10110, 11101\}$$

é um código em blocos, binário, de comprimento 5.

Se consideremos como alfabeto o conjunto $\mathbb{Z}_3 = \{0, 1, 2\}$. *O conjunto*

$$\mathcal{C}_2 = \{00012, 11022, 10101, 10201, 20202\}$$

é um código em blocos, ternário, de comprimento 5.

Notação Um código $(n, M)_q$ significa um código uniforme q-ário com M palavras de comprimento n. Também usamos (n, M) para denotar o mesmo tipo de códigos quando o número de símbolos q está subentendido.

Definição Um *esquema de codificação* é um par (C, f) onde

- C é um código,
- $f : \mathcal{S} \to C$ é uma aplicação injectiva, chamada *função de codificação*,
- $\mathcal{S}$ diz-se o *alfabeto fonte*.

O alfabeto fonte pode ou não ser o mesmo do código C. Em todos os exemplos anteriores, o conjunto $\{N, S, E, W\}$ é o alfabeto fonte e o alfabeto do código é $\{0, 1\}$.

Exemplos como esses definem funções de codificação.

Um alfabeto pode ser qualquer conjunto finito de símbolos à nossa escolha. O conjunto das letras $\{a, b, c, \ldots, x, y, z\}$ é naturalmente um alfabeto, e o conjunto de todas as palavras portuguesas formam um código que não é uniforme.

A partir de agora, iremos condiderar apenas códigos uniformes, assim "código" significará sempre "código uniforme".

Exemplo 1 Fixemos um alfabeto $\mathcal{A}_q$ de q elementos, por exemplo, $\mathcal{A}_q = \mathbb{Z}_q$. O *código de repetição q-ário de comprimento n* é o conjunto formado por q palavras em que os símbolos de cada palavra são todos iguais. Concretamente, $\{0000, 1111\}$ é o código de repetição binário de comprimento 4 e tem parâmetros $(4, 2)$, $\{000, 111, 222, 333, 444\}$ é o código de repetição quinquenário de comprimento 3 e tem parâmetros $(3, 5)$, etc.

Exemplo 2 Os parâmetros de um código não o definem univocamente. Seja $C_1 = \{0000, 1111\}$ o código de repetição binário e seja $C_2 = \{1010, 0101\}$. Estes dois códigos têm parâmetros $(4, 2)$, mas $C_1 \neq C_2$.

2. Canal de transmissão

Definição Um *canal de transmissão* consiste num alfabeto $\mathcal{A}_q = \{a_1, a_2, \ldots, a_q\}$ e nas probabilidades de canal $P(\text{recebido } a_j \mid \text{enviado } a_i)$, para $i, j \in \{1, \ldots, q\}$, verificando a seguinte condição

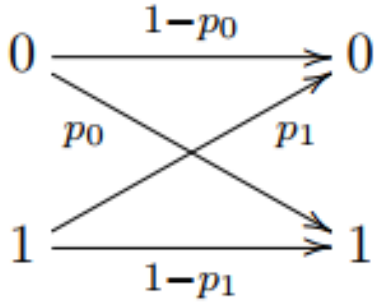
$$\sum_{j=1}^{q} P(\text{recebido } a_j \mid \text{enviado } a_i) = 1 \quad \text{, para cada } i \text{ fixo.}$$

Para simplificar a notação, por vezes escrevemos $P(a_j|a_i)$ para denotar a probabilidade condicionada $P(\text{recebido } a_j \mid \text{enviado } a_i)$, e indicamos as probabilidades do canal através de um grafo onde cada seta representa uma das probabilidades condicionadas da definição

$$a_i \xrightarrow{P(a_j|a_i)} a_j \quad .$$

Exemplo 1

Um *canal de transmissão binário* ($q = 2$) é definido pelos dois valores $p_0 = P(1|0)$ (a probabilidade de troca do símbolo 0) e $p_1 = P(0|1)$ (a probabilidade de troca do símbolo 1), e pode ser representado pelo seguinte esquema

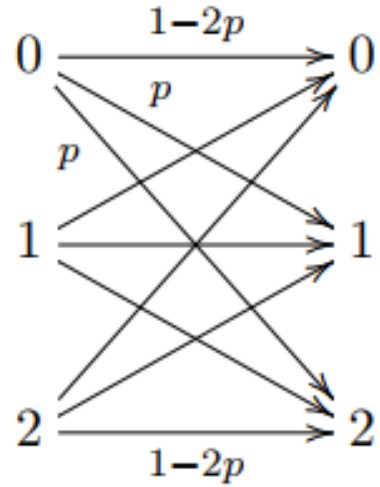


onde o número em cada seta é a probabilidade do símbolo da ponta da seta ser recebido dado que o símbolo da cauda da seta foi enviado. Portanto, neste exemplo, $P(0|0) = 1 - p_0$, $P(1|0) = p_0$, $P(0|1) = p_1$ e $P(1|1) = 1 - p_1$.

Se $p_0 = p_1$, obtém-se um *canal binário simétrico*

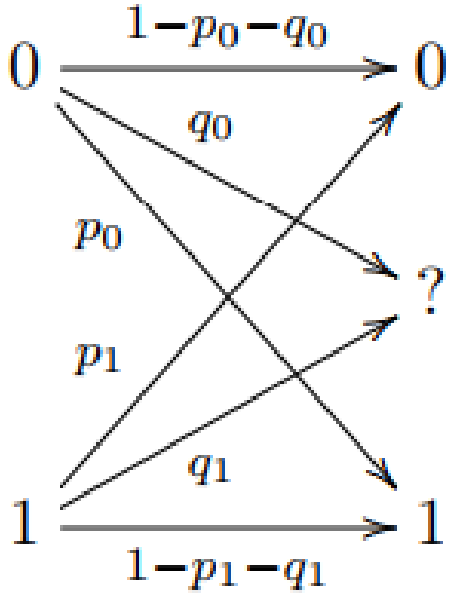
Neste caso, o número $p := p_0 = p_1$ diz-se a *probabilidade de troca de símbolos*.

Para $q = 3$, temos o caso particular de um *canal símetrico ternário* com probabilidade de troca $p \in]0, 1[$ definido pelo esquema



onde as setas diagonais têm todas probabilidade p e, portanto, as setas horizontais têm probabilidade $1 - 2p$ (a figura acima está incompleta), ou seja, $P(a_j|a_i) = p$ se $j \neq i$, e $P(a_i|a_i) = 1 - 2p$.

Outro exemplo interessante é o *canal binário de apagamento* definido por



Para além de cada símbolo do alfabeto $\mathcal{A}_2 = \{0, 1\}$ poder ser trocado durante a transmissão, pode ainda ser apagado, o que corresponde a ser enviado para o novo símbolo '?'. É equivalente a usar o alfabeto $\{0, 1, ?\}$ em que o símbolo de apagamento '?' não é usado em nenhuma palavra de código.

3. Descodificação

Fixemos um código q-ário C de comprimento n, isto é, $C \subset \mathcal{A}_q^n$ onde $\mathcal{A}_q$ é um alfabeto com q símbolos.

Definição 1 Um *método de descodificação* é uma correspondência entre palavras de $\mathcal{A}_q^n$ (vistas como as palavras recebidas) e palavras do código C. Caso esta correspondência não esteja definida em todas as palavras de $\mathcal{A}_q^n$, a descodificação diz-se *incompleta*.

Vamos considerar dois métodos de descodificação.

1. *Descodificação por máxima verosimilhança*

2 *Descodificação por distância mínima*

1 *Descodificação por máxima verosimilhança:* recebido $y \in \mathcal{A}_q^n$, procurar $x' \in C$ tal que

$$P(\text{recebido } y \mid \text{enviado } x') = \max_{x \in C}\{P(\text{recebido } y \mid \text{enviado } x)\} .$$

Como C é finito, o conjunto $\{P(\text{recebido } y \,|\, \text{enviado } x) \ : \ x \in C\}$ também é finito e, portanto, o máximo na definição anterior existe sempre, embora possa não ser único.

Exemplo 1. Seja $C = \{110, 111\}$ e considere-se um canal binário simétrico com probabilidade de troca $p = 0,03$. Suponhamos que recebemos a palavra 011. Como $011 \notin C$, sabemos que ocorreram erros durante a transmissão. Vamos usar o método de descodificação por máxima verosimilhança.

$$P(011 \text{ recebida} \,|\, 110 \text{ enviada}) = P(0|\,1)P(1|\,1)P(1|\,0)$$

$$= p(1-p)p = (0,03)^2 \times 0,97 = 0,000873$$

$$P(011 \text{ recebida} \,|\, 111 \text{ enviada}) = P(0|\,1)P(1|\,0)^2$$

$$= p(1-p)^2 = 0,03 \times (0,97)^2 = 0,028227$$

Como a última probabilidade é maior, concluimos que 111 é a palavra de código que provavelmente foi enviada, portanto descodificamos 011 por 111.

Exemplo 2. Consideremos a mesma situação do exemplo anterior, mudando apenas o código para $C = \{010, 111\}$. Continuamos a ter um canal simétrico binário e a mesma palavra recebida 011.

$$P(011 \text{ recebida} \,|\, 010 \text{ enviada}) = P(0|\,0)P(1|\,1)P(1|\,0)$$

$$= (1-p)^2 p$$

$$P(011 \text{ recebida} \,|\, 111 \text{ enviada}) = P(0|\,1)P(1|\,1)^2$$

$$= p(1-p)^2$$

Como as duas probabilidades são iguais (e nem dependem do valor de p), o método de descodificação por máxima verosimilhança não nos permite tirar conclusões acerca de qual a palavra enviada com maior probabilidade. Temos então duas alternativas. Ou optamos por uma descodificação incompleta, o que quer dizer que não descoficamos a palavra recebida 011; ou escolhemos uma das palavra de código para descodificar 011 sempre que esta seja recebida. Neste último caso, se decidirmos descodificar 011 por 010, por exemplo, da próxima vez que 011 for recebida, teremos que descodificá-la novamente pela mesma palavra $010 \in C$.

Há esquemas de decisão ou descodificação que não envolvem probabilidades, mas usam uma noção de proximidade.

Definição 1 Sejam $x, y \in \mathcal{A}_q^n$. Define-se a *distância de Hamming* entre as palavras x e y por

$$\mathrm{d}(x,y) = \#\{i : x_i \neq y_i\} .$$

Ou seja, $\mathrm{d}(x,y)$ é o número de coordenadas em que x e y diferem, ou ainda, $\mathrm{d}(x,y)$ é o número mínimo de trocas de símbolos necessárias para obter y a partir de x. Por exemplo, $\mathrm{d}(00,01) = 1$ e $\mathrm{d}(111000, 112012) = 3$.

Exemplo 1 . Considere-se o alfabeto $\mathcal{A}_4 = \{1,2,3,4\}$ e sejam $x = 1234$, $y = 2341$ e $z = 1243$. Então

$$\mathrm{d}(x,y) = 4 , \qquad \mathrm{d}(x,z) = 2 \qquad \text{e} \qquad \mathrm{d}(y,z) = 3 .$$

Definição 2 . Seja C um código contendo pelo menos duas palavras. Define-se a *distância mínima de* C por

$$\mathrm{d}(C) = \min\{\mathrm{d}(x,y) : x, y \in C, x \neq y\} .$$

Este parâmetro $\mathrm{d}(C)$ vai ter bastante importância quando discutirmos as capacidades de detecção e correcção de erros de um código C.

Notação · Se C é um código q-ário com M palavras de comprimento n e distância mínima $\mathrm{d}(C) = d$, dizemos que C é um código $(n, M, d)_q$, ou (n, M, d). Os números n, M e d dizem-se os *parâmetros* de C.

Exemplo Consideremos o código $C_5 = \{000, 011, 101, 110\}$

A distância entre $000 \in C_5$ e qualquer outra palavra (de comprimento 3, claro) é o número de símbolos não nulos nessa palavra, portanto $\mathrm{d}(000, x) = 2$ para qualquer $x \in C \setminus \{000\}$. Calculando a distância entre os restantes pares de palavras de código:

$$\mathrm{d}(011, 101) = 2\ , \qquad \mathrm{d}(011, 110) = 2\ , \qquad \mathrm{d}(101, 110) = 2\ ,$$

conclui-se que $\mathrm{d}(C_5) = 2$ e portanto $(3, 4, 2)_2$ são os parâmetros deste código.

2 *Descodificação por distância mínima:* recebida a palavra $y \in \mathcal{A}_q^n$, procurar $x' \in C$ tal que

$$\mathrm{d}(x', y) = \min\{\mathrm{d}(x, y) \,:\, x \in C\} \,,$$

ou seja, descodificamos y pela palavra de código mais próxima.

Tal como no caso da descodificação por máxima verosimilhança, por C ser finito, o conjunto $\{\mathrm{d}(x, y) \,:\, x \in C\}$ também é finito e o mínimo na definição anterior existe sempre, embora possa não ser único.

Exemplo 1. Consideremos o código binário $C = \{0010, 0101, 1010, 1110\}$ e suponhamos que recebemos a palavra 0100.

Como $\mathrm{d}(0100, 0010) = 2$, $\mathrm{d}(0100, 0101) = 1$, $\mathrm{d}(0100, 1010) = 3$, $\mathrm{d}(0100, 1110) = 3$,

usando o método de descodificação por distância mínima, descodificamos 0100 por 0101.

Exemplo 2. Seja $C = \{0000, 1111\}$ o código de repetição de comprimento 4 e consideremos um canal de transmissão binario simétrico com propabilidade de troca $p = \frac{1}{4}$. Pretende-se descodificar a palavra recebida $y = 0010$ pelo dois métodos definidos.

Descodificação por máxima verosimilhança: Temos de calcular as probabilidades condicionadas $P(\text{recebido } y \,|\, \text{enviado } x)$ para $x \in C$. Otém-se

$$P(\text{recebido } y \,|\, \text{enviado } 0000) = (1-p)^3 p = \frac{3^3}{4^4}$$

$$P(\text{recebido } y \,|\, \text{enviado } 1111) = p^3(1-p) = \frac{3}{4^4}$$

Como $\frac{3^3}{4^4} > \frac{3}{4^4}$, descodificamos y por 0000.

Descodificação por distância mínima: Temos de calcular as distâncias entre y e cada uma das palavras do código C. Obtêm-se

$$\mathrm{d}(y, 0000) = 1 \qquad \text{e} \qquad \mathrm{d}(y, 1111) = 3 \ ,$$

portanto descodificamos y por 0000, a mesma que se obteve pelo outro método. Não se trata de uma coincidência uma vez que as probabilidades calculadas apenas dependem no número de coordenadas em que x e y diferem, i.e., da distância $\mathrm{d}(x, y)$.

Teorema 1. *Para um canal simétrico binário com probabilidade de troca $p < \frac{1}{2}$ os esquemas de descodificação por máxima verosimilhança e por distância mínima coincidem.*

Notas de Combinatória e Teoria de Códigos (2011, revistas e aumentadas em 2013)
Joana Ventura. Disponível em:
https://www.math.tecnico.ulisboa.pt/~jventura/CTC/CTCnotas.pdf

4. Correcção e detecção de erros

Seja $C = \{000, 111\}$ o código de repetição binário de comprimento 3. Se usarmos a descodificação por distância mínima, cada palavra em $\mathcal{A}_2^3$ é descodificada de acordo com a seguinte tabela

recebido	descodificado por
000	000
100, 010, 001	000
011, 101, 110	111
111	111

Caso 1: Se 000 (ou 111) é a palavra enviada e ocorrem erros de transmissão em uma ou duas coordenadas, a palavra recebida y contém exactamente um ou dois símbolos 1. Embora não tenhamos informação para corrigir o erro (admitindo que não conhecemos a palavra enviada), podemos ainda concluir que ocorreram erros pois y não pertence ao código. Dizemos que C *detecta até dois erros.*

Seja $C = \{000, 111\}$ o código de repetição binário de comprimento 3. Se usarmos a descodificação por distância mínima, cada palavra em $\mathcal{A}_2^3$ é descodificada de acordo com a seguinte tabela

recebido	descodificado por
000	000
100, 010, 001	000
011, 101, 110	111
111	111

Caso 2: Se a palavra enviada foi 000 e ocorreu um erro na transmissão de um dos símbolos, a palavra recebida foi uma das da segunda linha da tabela, portanto é descodificada correctamente por ela própria. Ou seja, o *erro foi corrigido*. Analogamente para o caso de ocorrer um erro numa das coordenadas de 111. Caso ocorram dois erros na transmissão de 000, a palavra recebida é descodificada incorrectamente por 111 (terceira linha da tabela). Dizemos que C *corrige um erro*, mas não corrige dois.

Definição 1.29. Seja C um código e sejam s e t números inteiros positivos.

- Diz-se que C *detecta* s *erros* se e só se, quando ocorrem s erros ou menos, a palavra obtida não pertence ao código C.
- Diz-se que C *corrige* t *erros* se e só se o método de descodificação por distância mínima corrige t, ou menos, erros.

Em particular, "corrigir" quer dizer que há unicidade de mínimo na definição de descodificação, i.e., está-se a usar um método de descodificação incompleta em que não se descodifica a palavra recebida em caso de "empate".

Seja C um código de distância mínima $\mathrm{d}(C) = d$. Então C detecta precisamente $d-1$ erros, ou corrige precisamente $\left\lfloor \frac{d-1}{2} \right\rfloor$ erros.

Exemplo:

(Um Código de Hamming Binário) Codifica-se um vector mensagem de 4 componentes binárias $m = m_1m_2m_3m_4$, com $m_i \in \{0,1\}$, numa palavra de código com 7 componentes binárias $c = c_1c_2c_3c_4c_5c_6c_7$, com $c_j \in \{0,1\}$, definidas por

$$c_3 = m_1 \quad ; \qquad c_5 = m_2 \quad ; \qquad c_6 = m_3 \quad ; \qquad c_7 = m_4$$

e as restantes componentes escolhidas

$$\begin{aligned} &c_4 \text{ tal que } \alpha = c_4 + c_5 + c_6 + c_7 \text{ seja par,} \\ &c_2 \text{ tal que } \beta = c_2 + c_3 + c_6 + c_7 \text{ seja par,} \\ &c_1 \text{ tal que } \gamma = c_1 + c_3 + c_5 + c_7 \text{ seja par.} \end{aligned}$$

Verifique que com este esquema de codificação se constrói um código que permite corrigir um erro em qualquer posição.
Recebido um vector $x = x_1x_2x_3x_4x_5x_6x_7$, calculam-se

$$\left.\begin{aligned} \alpha &= x_4 + x_5 + x_6 + x_7 \\ \beta &= x_2 + x_3 + x_6 + x_7 \\ \gamma &= x_1 + x_3 + x_5 + x_7 \end{aligned}\right\} \quad \text{mod } 2 \ ;$$

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

1) Uma palavra de 7 bits em código Hamming é recebida. Verificar seu estado, detectando os erros e apresentar a palavra original enviada, corrigindo o(s) bit(s) incorretos.

- Palavra recebida: 1010111

2) Uma palavra de 9 bits em código Hamming é recebida. Verificar seu estado, detectando os erros e apresentar a palavra original enviada, corrigindo o(s) bit(s) incorretos.

- Palavra recebida: 101010101