

# Taller 1 - Detección y corrección de errores

## Objetivos

Profundizar en los códigos correctores de errores, considerando códigos básicos basados en paridad y la presentación del código de Hamming (7,4), y el método de checksum.

## Introducción

Todos los sistemas de codificación que permiten detección y corrección de errores se basan en una misma idea: agregar redundancia a la información.

Los códigos pueden ser:

- *Detectores*, cuando solamente detectan la presencia de errores.
- *Correctores*, cuando con la información que cuentan son capaces de corregir el o los errores detectados. Todos los códigos correctores son detectores.

La mayoría de los códigos correctores y detectores se basan en la aplicación del código de paridad. Un código de paridad agrega un bit de redundancia a la cadena codificada, que obliga a que la paridad (calculada mediante un XOR  $\oplus$  de todos los bits enviados) sea par (XOR igual a 0) o impar (XOR igual a 1).

Como ejemplo se presenta un código de paridad simple de 8 bits, de los cuales 7 son bits de datos ( $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ ) y el restante bit es de paridad ( $p_1$ ). El código corresponde a

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$p_1$
-------	-------	-------	-------	-------	-------	-------	-------

donde  $p_1 = a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7$

Este tipo de códigos puede extenderse agregando la verificación de más elementos del mensaje original que se desea preservar.

## Resolver los siguientes problemas

- Considerando un método de paridad simple indicar:
  - ¿Cuántos errores puede detectar la paridad simple? Explique su argumentación.
  - ¿Cuántos errores puede corregir la paridad simple? Explique su argumentación.
  - ¿Cuál es la distancia de este sistema de codificación?
- Se plantea el siguiente código basado en paridad (conocido como Hamming [7,4]).

Dados 4 bits a codificar ( $a_4 a_3 a_2 a_1$ ) se envía la palabra  $a_4 a_3 a_2 p_3 a_1 p_2 p_1$  donde  $p_1, p_2, p_3$  se calculan en base a las siguientes fórmulas:

$$p_3 = a_4 \oplus a_3 \oplus a_2$$

$$p_2 = a_4 \oplus a_3 \oplus a_1$$

$$p_1 = a_4 \oplus a_2 \oplus a_1$$

$a_4$	$a_3$	$a_2$	$p_3$	$a_1$	$p_2$	$p_1$
-------	-------	-------	-------	-------	-------	-------

- ¿Cuál es la distancia de éste sistema de codificación?
- Indique y justifique si el código es corrector o solamente detector.
- Describa cómo se comportan los bits  $s_3 s_2 s_1$  para todos los casos de errores simples posibles. ¿Qué característica tiene el resultado?
- Suponga que desea almacenar en memoria la palabra 1010 ( $a_4 a_3 a_2 a_1$ ) con redundancia. Calcule los valores de  $p$  y presente la palabra final que será almacenada.
- Suponga que ahora se leyó de memoria la misma palabra pero con un error en uno de los bits (elegir un bit cualquiera y cambiarlo). Calcule los valores de  $S$  ( $s_3 s_2 s_1$ ) para detectar si hubo error y su posición.
- Repita las partes iii y iv para la palabra 1001.

- c. La suma de comprobación (checksum) es una técnica simple para detectar errores en bloques de información. El método consiste en agrupar el mensaje a transmitir en tiras de  $l$  bits, luego se suma cada una de las tiras de  $l$  bits y se agrega el resultado módulo  $2^l$  al final del mensaje.
- i. Implemente en C un algoritmo de codificación y de decodificación de mensajes utilizando suma de comprobación para  $l = 8$ . Asuma que recibe una cadena de char conteniendo los caracteres y un entero con el largo del arreglo.
  - ii. Calcule el checksum de la palabra "Arqui" representada en ASCII (ver ejemplo en las notas de teórico, página 9).
  - iii. Discuta que pasaría si en entre el proceso de codificar y el de decodificar, existen errores en la transmisión del tipo:
    1. aislados
    2. de ráfagas