

Práctica 8. Sesión 1 - GAP / GUAVA

1 de diciembre del 2022

Marcos Hidalgo Baños

NOTA. El archivo .log que recoge cada comando con su resultados se encuentra en este [enlace](#) a mi repositorio personal de GitHub.

0) Preliminares. Importación del paquete GUAVA para GAP.

```
#####
# Preliminares. Importacion del paquete #
#####
gap> LoadPackage("guava");

-----
SONATA Version 2.9.5
Erhard Aichinger
Franz Binder
Juergen Ecker
Peter Mayr
Christof Noebauer

System Of Nearrings And Their Applications
Info: https://gap-packages.github.io/sonata/

GUAVA Version 3.17
the GUAVA Group

Homepage: https://gap-packages.github.io/guava
Report issues at https://github.com/gap-packages/guava/issues
true
```

a) Construir el código C1 cuya matriz generadora G es:

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

```
#####
# Construccion del codigo C1 #
#####
gap> G := [[1,1,0,0,0,0,0],[0,1,1,0,1,1,0],[1,1,1,0,0,1,1],[1,1,1,0,0,1,1],[0,0,0,1,0,1,1]];
[ [ 1, 1, 0, 0, 0, 0, 0 ], [ 0, 1, 1, 0, 1, 1, 0 ], [ 1, 1, 1, 0, 0, 1, 1 ],
  [ 1, 1, 1, 0, 0, 1, 1 ], [ 0, 0, 0, 1, 0, 1, 1 ] ]

gap> C1 := GeneratorMatCode(G,GF(2));
a linear [7,4,1..2]1..3 code defined by generator matrix over GF(2)
```

b) Listar todas las palabras del código.
¿Cuál es la longitud de este código?

```
#####
# Listado de las palabras del codigo #
#####
gap> C := Elements(C1);
[ [ 0 0 0 0 0 0 0 ], [ 0 0 0 1 0 1 1 ], [ 0 0 1 0 0 1 1 ], [ 0 0 1 1 0 0 0 ],
  [ 0 1 0 0 1 0 1 ], [ 0 1 0 1 1 1 0 ], [ 0 1 1 0 1 1 0 ], [ 0 1 1 1 1 0 1 ],
  [ 1 0 0 0 1 0 1 ], [ 1 0 0 1 1 1 0 ], [ 1 0 1 0 1 1 0 ], [ 1 0 1 1 1 0 1 ],
  [ 1 1 0 0 0 0 0 ], [ 1 1 0 1 0 1 1 ], [ 1 1 1 0 0 0 1 ], [ 1 1 1 1 0 0 0 ] ]

#####
# Longitud de las palabras del codigo #
#####
gap> L := WordLength(C1);
7
```

c) Empleando GUAVA, calcular la dimensión de este código.

```
#####
# Dimension del codigo #
#####
gap> D := Dimension(C1);
4
```

- d) **Calcula la mínima distancia entre las palabras del código. ¿Cuántos errores se pueden corregir y cuantos se pueden detectar? ¿Por qué?**

```
#####  
# Minima Distancia entre palabras #  
#####  
gap> M := MinimumDistance(C1);  
2
```

Capacidad Correctora = $(\text{Min Dist.} - 1) / 2 = (2 - 1) / 2 = 0.5$

No se puede corregir ningún error puesto que la capacidad correctora no es mayor que 1.

- e) **Calcular la matriz de control del código.**

```
gap> Check := CheckMat(C1);  
<an immutable 3x7 matrix over GF2>  
  
gap> Codeword(Check);  
[ [ 1 1 0 0 1 0 0 ], [ 0 0 1 1 0 1 0 ], [ 1 1 1 1 0 0 1 ] ]
```

- f) **Visualizar la distribución de pesos de las palabras del código. ¿Cuántas palabras hay de cada peso?**

```
#####  
# Distribucion de pesos de las palabras #  
#####  
gap> Dist := WeightDistribution(C1);  
[ 1, 0, 2, 4, 5, 4, 0, 0 ]
```

- g) **Calcular el radio de recubrimiento del código.**

```
#####  
# Numero de palabras para cada peso #  
#####  
gap> Cover := CoveringRadius(C1);  
2
```

h) Comprobar si el código es perfecto o MDS (separable a máxima distancia).

```
#####  
# Comprobar si el código es perfecto #  
#####  
gap> perf := IsPerfectCode(C1);  
false  
  
#####  
# Comprobar si el código es MDS #  
#####  
gap> mds := IsMDSCode(C1);  
false
```

i) Codificar la información (1101).

```
#####  
# Codificar la información 1101 #  
#####  
brk> [1101] * C1;  
[ 1 1 0 0 0 0 0 ]
```

- Marcos Hidalgo Baños -