Código Binario Gray

Gonzalo Mena Mendoza

mena@computer.org
http://mena.com.mx/gonzalo/maestria/algor/CodigoBinarioGray.nb

Proyecto de la materia de Análisis de Algoritmos y Estructuras de Datos Avanzadas
Profesor:Dr.Jaime Rangel Mondragón

Facultad de Informática
Universidad Autónoma de Querétaro
marzo de 2005

Dada una $n \in \mathbb{I}$, $n \geq 0$ enumerar las 2^n cadenas binarias que componen el Código Binario Gray de n bits.

Referencias

[1] The Art of Computer Programming, Volumen 4 (borrador)
D. E. Knuth,
http://www-cs-faculty.stanford.edu/~knuth/news.html

[2] Wikipedia, The Free Encyclopedia
http://en.wikipedia.org/wiki/Gray_code

Introducción

Consideremos el problema de obtener todas las 2^n cadenas que consisten de n dígitos binarios, una solución absurdamente sencilla consiste en comenzar con el número binario $(0. .00)_2 = 0$ y sumar 1 repetidamente hasta llegar a $(1. .11)_2 = 2^n-1$. Sin embargo hay situaciones en que preferiríamos visitar dichas 2^ncadenas en un orden distinto. La más famosa alternativa es el llamado **Código Binario Gray** que enumera todas las 2^n cadenas de n bits de tal manera que sólo un bit cambie entre cualesquiera dos cadenas contiguas [1].

Por ejemplo, el código binario Gray para n = 4 es:

0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000

Table 1

En realidad puede haber más de un código Gray para un tamaño dado de cadena. Sin embargo el término fue utilizado por vez primera para un código binario específico, el código binario reflejado Gray, BRGC [2].Como puede verse en la Tabla 1, la propiedad de cambio de un sólo bit entre cadenas adyacentes también se cumple entre el primero y el último elemento.

En la siguiente figura se muestra del lado izquierdo el órden lexicográfico para n = 4; y del lado derecho, el código binario Gray correspondiente[1].

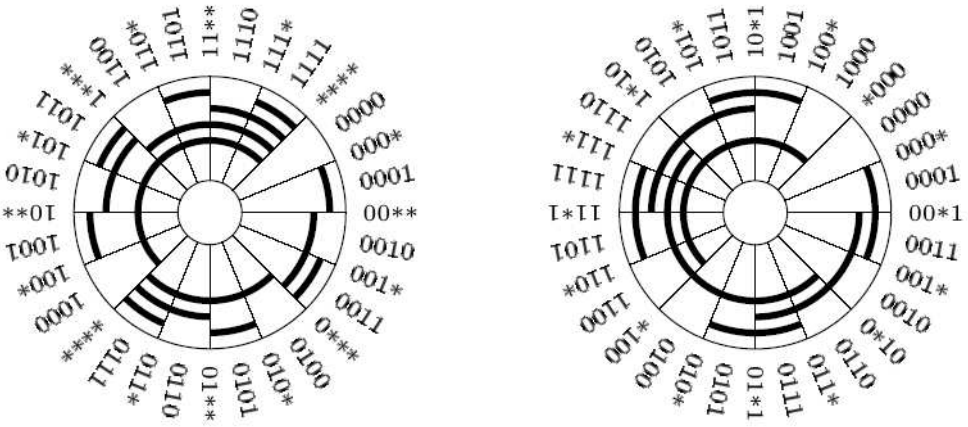

Figure 1

Este código es importante en aplicaciones en las que información analógica es convertida en digital o viceversa. Por ejemplo, supóngase que queremos identificar la posición de un disco rotatorio que ha sido dividido en 16 sectores utilizando 4 sensores, cada uno de los cuales puedes distinguir entre blanco y negro. Si empleamos el orden lexicográfico para marcar cuatro pistas concéntricas, una para cada sensor, como en la figura de la izquierda, corremos el riesgo de obtener mediciones altamente imprecisas cuando los sensores se encuentran entre las separaciones de los sectores. Sin embargo, la figura de la derecha nunca dará una mala lectura.

Análisis

El caso para n = 1 es trivial pues se trata de la misma secuencia que en el orden lexicográfico.

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Para n = 2 notamos que la mitad superior de la secuencia difiere de la inferior en el bit más significativo. Además el orden del bit menos significativo de la mitad inferior de la secuencia se invierte con respecto a la primera mitad.

$$\begin{pmatrix} 0 & \mathbf{0} \\ \underline{0} & \underline{\mathbf{1}} \\ 1 & \mathit{1} \\ 1 & \mathit{0} \end{pmatrix}$$

Si abajo del código Gray para n = 2 añadimos una copia del mismo invertida en el eje vertical se cumple la propiedad de cambio en un solo bit, salvo en las dos filas centrales y entre la primera y la última fila. Entonces añadimos un 0 al comienzo de cada fila de la primera mitad y un 1 a cada fila de la seguna mitad. El resultado es el código binario Gray para n = 3.

$$\begin{pmatrix} 0 & \mathbf{0} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} \\ \underline{0} & \underline{\mathbf{1}} & \underline{\mathbf{0}} \\ 1 & \mathit{1} & \mathit{0} \\ 1 & \mathit{1} & \mathit{1} \\ 1 & \mathit{0} & \mathit{1} \\ 1 & \mathit{0} & \mathit{0} \end{pmatrix}$$

Definición

Podemos entonces definir el código binario Gray de manera recursiva. Si decimos que Γ_n representa la secuencia binaria Gray de n bits, entonces podemos definir a Γ_n con estas dos reglas [1]:

$$\Gamma_0 = \epsilon;$$
$$\Gamma_{n+1} = 0\,\Gamma_n,\ 1\,\Gamma_n{}^R \qquad (1)$$

Donde ϵ representa una cadena vacía, $0\,\Gamma_n$ denota la secuencia Γ_n con prefijo 0 añadido a cada cadena, y $1\,\Gamma_n{}^R$ simboliza el reverso de la secuencia Γ_n con prefijo 1 agregado a cada cadena.

Programación

Si representamos el conjunto de cadenas binarias como una lista compuesta a su vez por 2^n listas, y en donde cada elemento de las segundas es un dígito, podemos definir las siguientes dos funciones que corresponden a las reglas en (1):

```
CódigoGray[0] := {{}};
CódigoGray[n_] := Join[
  Map[Prepend[#, 0] &, CódigoGray[n - 1]],
  Map[Prepend[#, 1] &, Reverse[CódigoGray[n - 1]]]
 ]
```

```
CódigoGray[3]
```

```
{{0, 0, 0}, {0, 0, 1}, {0, 1, 1}, {0, 1, 0},
 {1, 1, 0}, {1, 1, 1}, {1, 0, 1}, {1, 0, 0}}
```

Si lo ponemos en forma de matriz para hacerlo más legible:

```
MatrixForm[CódigoGray[3]]
```

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

■ Ejemplos

```
MatrixForm[CódigoGray[1]]
```

```
MatrixForm[CódigoGray[2]]
```

```
MatrixForm[CódigoGray[3]]
```

```
MatrixForm[CódigoGray[4]]
```