

Escuela técnica superior

FACULTAD DE INGENIERÍA INFORMÁTICA

PRÁCTICA 4

PROGRAM NUMBERING AND EXWHILE

Ignacio Fernández Contreras
3º Informática A

1. El desarrollo del cálculo de la menor codificación del programa WHILE "diverger".

El programa While más simple sin argumentos que computa la función de divergencia es:

```
Q = (0,s)
s:
    X1 := X1 + 1;
    while X1 != 0 do
        X1 := X1
    od
```

Para asegurar que un programa siempre diverge, es necesario utilizar un bucle. El cuerpo del bucle es el mínimo requerido para que el programa pueda diverger. Esta primera instrucción es fundamental para garantizar que el programa siempre diverja en lugar de converger.

La codificación del código s es:

```
> CODE2N("X2:=X1+1; while X2!=0 do X1:=0 od")
ans = 10876
```

2. El código Octave que hace un print de todos los vectores

Sabemos que podemos establecer una biyección entre todos los vectores y N , por lo que solo necesitamos un programa con un bucle que pueda imprimir todo el conjunto de vectores. El siguiente código imprime los primeros N vectores.

```
function printNvectors(N)
    for i=0: N-1
        dis(["(" num2str(godeldecoding(i)) ")"])
    end
end
```

La salida por pantalla del programa anterior para una entrada $N = 20$:

```
printNvectors(20)
()
(0)
(0 0)
(1)
(0 0 0)
(1 0)
(2)
(0 0 0 0)
(1 0 0)
(0 1)
(3)
(0 0 0 0 0)
(1 0 0 0)
(0 1 0)
(2 0)
(4)
(0 0 0 0 0 0)
(1 0 0 0 0)
(0 1 0 0)
(2 0 0)
```

3. El código Octave que hace un print de todos los programas WHILE

Este caso es muy similar al caso de la actividad 2, ya que existe una biyección entre los programas WHILE y N. Esto nos permite utilizar un script de Octave para generar todos los programas WHILE posibles, simplemente recorriendo todos los posibles valores de N. El script de Octave para lograr esto es el siguiente:

```
function printNwhilePrograms(N)
    for i=0: N-1
        disp(N2WHILE(i))
    end
end
```

En este script, el bucle `for` recorre todos los valores de N, imprimiendo cada uno de ellos. De esta manera, podemos generar todos los programas WHILE posibles utilizando un script de Octave. Dado que el código mencionado no ejecuta en mi maquina, he generado un código alternativo haciendo llamadas más concretas a funciones:

```
function printNwhilePrograms(N)
    for i=0: N-1
        n = cantordecoding(i, 2, 1);
        code = N2CODE(cantordecoding(i, 2, 2));
        program = strcat('(' , num2str(n) , ' , ' , code , ')');
        program;
    end
end
```

```
printNwhilePrograms(N)
program = (0, X1:=0)
program = (1, X1:=0)
program = (0, X1:=0; X1:=0)
program = (2, X1:=0)
program = (1, X1:=0; X1:=0)
program = (1, X1:=X1)
program = (3, X1:=0)
program = (2, X1:=0; X1:=0)
program = (1, X1:=X1)
program = (0, X1:=0; X1:=0; X1:=0)
```