

Práctica 4

Mario Santos Heredia

26 de diciembre de 2022

Ejercicio 1

Para la realizacion de este ejercicio vamos a crear un programa WHILE que sea simple y que además diverge con cero argumentos de entrada:

```
Q = (0,s)
s:
  X2 := X1 + 1;
  while X2 ≠ 0 do
    X1 := 0
  od
```

Ejecutamos el codigo en octave para comprobar que realmente diverge:

```
octave:17> F_emulation("(1, X2:=X1+1; while X2!=0 do X1:=0 od)", 0)
^C
octave:18> █
```

La codificación de dicho código será la siguiente:

```
octave:22> CODE2N("X2:=X1+1;while X2!=0 do X1:=0 od")
ans = 10876
octave:23>
```

Ejercicio 2

Para este ejercicio crearemos un codigo con el que imprimiremos N vectores de tamaño variado. El codigo es el siguiente:

```
function retval = printNvectors (N)

  for i = 0: N-1
    disp ([ '( ' num2str(godeldecoding(i)) ' '])
  end

end
```

A continuación vamos a testear su comportamiento en Octave:

```
octave:12> printNVector(10)
()
(0)
(0 0)
(1)
(0 0 0)
(1 0)
(2)
(0 0 0 0)
(1 0 0)
(0 1)
octave:13>
```

Ejercicio 3

Para este último ejercicio, crearemos una función capaz de calcular N programas WHILE, cuyo código implementado sería:

```
function retval = printNWhileprograms (N)

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

end
```

Usamos Octave para comprobar el resultado de la función:

```
octave:13> printNWhilePrograms(10)
(0, X1:=0)
(1, X1:=0)
(0, X1:=0; X1:=0)
(2, X1:=0)
(1, X1:=0; X1:=0)
(0, X1:=X1)
(3, X1:=0)
(2, X1:=0; X1:=0)
(1, X1:=X1)
(0, X1:=0; X1:=0; X1:=0)
octave:14>
```

Como podemos ver la salida por pantalla da el número de programas WHILE que hemos pedido por pantalla.