

Teoría de Autómatas y Lenguajes Formales

Practice 4: Program Numbering and EXWHILE

Pablo Fazio Arrabal

Ejercicio 1 - WHILE Coding

Ejercicio 1. Calculate the smallest number of a WHILE code that computes the *diverge* function (with zero arguments).

Construcción del código WHILE

El programa WHILE sin argumentos de entrada más pequeño será un bucle infinito definido por una variable que nunca alcanza el valor 0, de la forma:

```
diverge = (0, s)  
s:  
   $X_1 := X_1 + 1$ ;  
  while  $X_1 \neq 0$  do  
     $X_1 := X_1$   
  od
```

Ejercicio 2 - Gödel Decoding

Ejercicio 2. Create an Octave script to print all the vectors.

Construcción del script de Octave

En teoría, hemos visto que se puede establecer una biyección entre \mathbb{N} y el conjunto de todos los vectores en \mathbb{N}^* . Por tanto, será posible construir un programa que imprima por pantalla los N primeros vectores, de la forma:

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



```
pablofa02@pablofa02-Modern-14-A10RB: ~/talfuma/software/Whilelan...
octave:11> printNWhile(25)
(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)
(4, X1:=0)
(3, X1:=0; X1:=0)
(2, X1:=X1)
(1, X1:=0; X1:=0; X1:=0)
(0, X1:=X1; X1:=0)
(5, X1:=0)
(4, X1:=0; X1:=0)
(3, X1:=X1)
(2, X1:=0; X1:=0; X1:=0)
(1, X1:=X1; X1:=0)
(0, X1:=X1+1)
(6, X1:=0)
(5, X1:=0; X1:=0)
(4, X1:=X1)
(3, X1:=0; X1:=0; X1:=0)
octave:12> 
```