

Teoría de Autómatas y Lenguajes Formales

Práctica 4: Program Numbering y EXWHILE

Álvaro, Luque Torres

25 de diciembre de 2022

Actividades

1. Crea el programa **WHILE** más simple que computa la función *diverge* (con 0 argumentos) y computa la codificación de su código.
El programa más simple que *diverge* es:

Q = (0, s)

s:

```
X2 := X1 + 1;  
while X2 ≠ 0 do  
  X1 := 0  
od
```

La codificación del código es:

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

$$code2N(c) = \Gamma(sent2N(s_1), \dots, sent2N(s_m)) - 1$$

sent2N(X₂ := X₁ + 1) = 7

sent2N(while X₂ ≠ 0 do X₁ := 0 od) = $5\sigma_1^2(i-1, code2N(X_1 := 0)) + 4 = 9$

code2N(c) = $\Gamma(2, 44) - 1 = 10877 - 1 = 10876$

2. Cree un script de Octave que enumere todos los vectores.

```
function Nvectors(N)

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

```
>> Nvectors(15)
()
(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 0 1)
(2 0)
```

3. Cree un script de Octave que enumere todos los programas WHILE.

```
function Nwhiles(N)

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

end
```

```
>> Nwhiles(15)
(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)
```