

Teoría de Autómatas y Lenguajes Formales

Práctica 4

José Manuel Rodríguez Chicano

December 26, 2022

1 Ejercicio 1

Create the simplest WHILE program that computes the diverge function (with zero arguments) and compute the codification of its code.

La forma más fácil de realizar un programa WHILE sin argumentos de entrada que compute una divergencia es aquel cuyo bucle nunca llegue a 0 y la condición para que este termine sea cuando una variable X llegue a 0.

```
Q: (0,0,diverge)
```

```
diverge:
```

```
X2:= X1 + 1;
```

```
while X2 != 0 do
```

```
    X1 := 0
```

```
od
```

Y la computación de su código realizado en el octave con la función CODE2N sería la siguiente:

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

2 Ejercicio 2

Create an Octave script that enumerates all the vectors.

Al ser posible establecer una biyección de N con el conjunto de vectores N^* . Se puede crear un programa que imprima los N vectores por pantalla.

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

Y podemos comprobar su funcionamiento con el propio OCTAVE ejecutando el código con una N cualquiera en este caso 40.

```
>> printNvectors(40)
()
(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)
(4)
(0 0 0 0 0 0)
(1 0 0 0 0)
(0 0 0 1)
(0 1 0)
(1 1)
(5)
(0 0 0 0 0 0 0)
(1 0 0 0 0 0)
(0 0 0 0 1)
(0 0 1 0)
(1 0 1)
(0 2)
(6)
(0 0 0 0 0 0 0 0)
(1 0 0 0 0 0 0)
(0 0 0 0 0 1)
(0 0 0 1 0)
(1 0 0 1)
(0 0 2)
(3 0)
(7)
(0 0 0 0 0 0 0 0 0)
(1 0 0 0 0 0 0 0)
(0 0 0 0 0 0 1)
(0 0 0 0 0 0 0 1)
```

3 Ejercicio 3

Create an Octave script that enumerates all the WHILE programs.

Los programas WHILE también pueden establecer una biyección con el conjunto N

Así podemos crear un código que imprima en pantalla los N primeros programas WHILE asociados a cada número natural de la siguiente manera:

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

Y usando OCTAVE podemos comprobar el funcionamiento del mismo:

```
>> printNwhilePrograms(35)
(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)
(2, X1=X1; X1=0)
(1, X1=X1+1)
(0, X1=0; X1=0; X1=0; X1=0)
(7, X1=0)
(6, X1=0; X1=0)
(5, X1=X1)
(4, X1=0; X1=0; X1=0)
(3, X1=X1; X1=0)
(2, X1=X1+1)
(1, X1=0; X1=0; X1=0; X1=0)
```