

Práctica 4 TALF

David Molina López

24 de diciembre de 2022

1. WHILE program diverge function

Para que un programa WHILE diverja se necesita que entre en bucle y que no salga de este. Para ello el siguiente programa WHILE nunca llega a cumplir la condición de salida.

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

2. Enumeración de todos los vectores

El siguiente programa escrito en Octave es capaz de generar todos los vectores, aunque está limitado a los N primeros vectores. Para ello se hace un bucle que dure N ciclos en el que se usa la codificación de Gödel desde N hasta N*, partiendo desde el 0 hasta el número de vectores que se quiera generar.

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

```
>> 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 0 1)
(2 0)
(4)
(0 0 0 0 0 0)
(1 0 0 0 0)
(0 0 0 1)
(0 1 0)
```

3. Enumeración de todos los programas WHILE

El siguiente programa escrito en Octave es capaz de generar todos los programas WHILE, aunque esta limitado a los N primeros programas WHILE. Para ello se hace un bucle que dure N ciclos en el que se usa el programa N2WHILE que transforma de N a programas WHILE , partiendo desde el 0 hasta el numero de programas que se quiera generar.

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

```
>> printNwhilePrograms(20)
(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)
```