

# Ejercicios Relación 3 y 4

Alejandro Manzanares Lemus  
alexmnzlm@correo.ugr.es

15 de mayo

## 1 Ejercicio 1

Diseñar un programa con variables que dadas dos cadenas  $u$  y  $w$  compuestas por ceros y unos, calcule  $u^{|w|}$ . Diseña un programa Post-Turing que calcule la misma función, suponed que la entrada es  $ucv$  donde  $c$  es un simbolo diferente de 0 y 1.

### 1.1 Programa con variables

La idea de este programa es cargar en  $Z$  la  $u$  para poder ir copiándolo en  $Y$  tantas veces como elementos haya en  $w$ . Se va restando 1 elemento a  $w$  cada vez que se vacía  $Z$  lo que significa que  $u$  se ha copiado una vez, hasta que  $w$  se queda sin elementos.

Suponemos que el programa con variables tiene la entrada de la forma:

$X1 = u$   
 $X2 = w$

Y la salida de la siguiente forma:

$Y = u^{|w|}$

[R]

$Z \leftarrow X1$   
IF  $X2 \neq \#$  GOTO NP  
HALT

[NP]

$X2 \leftarrow X2 -$

[SC]

IF  $Z$  ENDS 0 GOTO C0  
IF  $Z$  ENDS 1 GOTO C1  
GOTO R

[C0]

$Z \leftarrow Z -$   
 $Y \leftarrow 0Y$

```

GOTO SC
[C1]
Z <- Z-
Y <- 1Y
GOTO SC

```

## 1.2 Máquina Post-Turing

La versión del programa Post-Turing es algo más compleja, la idea es dividir la palabra de la siguiente forma:  $ucwdu^{|w|}$ . Lo primero que hacemos es añadir una  $d$  al final de la palabra  $ucw$ . Así después de la  $d$  se irá copiando  $u$  tantas veces como elementos haya en  $w$ . Desde  $c$ , se recorre la palabra y el primer elemento distinto de  $\#$  se sustituye por  $\#$ . Hecho esto nos movemos al principio de la palabra y cada vez que leamos un  $0$ , ponemos una  $a$  y escribimos un  $0$  al final (en el primer  $\#$  después de  $d$ ) y si leemos un  $1$ , escribiremos una  $b$  y escribiremos un  $1$  al final volviendo al principio después. Cuando se llegue a  $c$ , volvemos al principio y cambiamos  $a$  por  $0$  y  $b$  por  $1$  para devolver  $u$  a su estado original, y desde  $c$  se vuelve a seguir la secuencia de recorrer la palabra buscando el primer elemento distinto de  $\#$ . Si ya no quedasen elementos distintos de  $\#$  entre  $c$  y  $d$ , hemos terminado y nos movemos al principio de la palabra y borramos todos los elementos antes de la  $d$  y la propia  $d$ . El resultado será  $u^{|w|}$

```

RIGHT TO NEXT BLANK
PRINT d
LEFT TO NEXT C
[EW]
RIGHT
IF # GOTO EW
IF D GOTO BT
PRINT #
LEFT TO NEXT C
LEFT TO NEXT BLANK
[S]
IF 0 GOTO E0
IF 1 GOTO E1
IF c GOTO RW
RIGHT
GOTO S
[E0]
PRINT a
RIGHT TO NEXT D
RIGHT TO NEXT BLANK
PRINT 0
LEFT TO NEXT D
LEFT TO NEXT C
LEFT TO NEXT BLANK

```

```

      GOTO S
[E1]  PRINT b
      RIGHT TO NEXT D
      RIGHT TO NEXT BLANK
      PRINT 1
      LEFT TO NEXT D
      LEFT TO NEXT C
      LEFT TO NEXT BLANK
      GOTO S
[RW]  LEFT TO NEXT BLANK
[C]   RIGHT
      IF a GOTO R0
      IF b GOTO R1
      IF c GOTO EW
[R0]  PRINT 0
      GOTO C
[R1]  PRINT 1
      GOTO C
[BT]  LEFT TO NEXT C
      LEFT TO NEXT BLANK
[D]   PRINT #
      RIGHT
      IF D GOTO H
      GOTO D
[H]   PRINT #
      HALT

```

## 2 Ejercicio 2

Diseñar un programa con variables numéricas que calcule el máximo común divisor de dos enteros.

Para resolver este ejercicio voy a implementar el algoritmo de Euclides para calcular el máximo común divisor de dos números enteros. Para ellos necesito implementar las siguientes macros:

- La macro resta: Resta V a U y almacena en U el resultado

U  $\leftarrow$  U - V:

```

    Z  $\leftarrow$  V
[A]  Z  $\leftarrow$  Z-1
    U  $\leftarrow$  U-1
    IF Z != 0 GOTO A

```

- La macro menor que: Si se cumple que U < V salta a la instrucción E

IF U < V GOTO E:

```

    Z1  $\leftarrow$  U
    Z2  $\leftarrow$  V
[A]  Z1  $\leftarrow$  Z1-1
    Z2  $\leftarrow$  Z2-1
    IF Z1 != 0 GOTO A
    IF Z2 != 0 GOTO E

```

- La macro modulo: Almacena en W la operación V mod U

W  $\leftarrow$  V % U:

```

    Z1  $\leftarrow$  U
    Z2  $\leftarrow$  V
    IF Z2 < Z1 GOTO A
[B]  W  $\leftarrow$  Z1
[A]  Z1  $\leftarrow$  Z1 - Z2
    IF Z1 < Z2 GOTO B
    GOTO A

```

## 2.1 Programa con variables

Realizamos la operación modulo hasta que este sea 0 y entonces devolvemos Z2 que es el mcd de X1 y X2.

```

    Z1  $\leftarrow$  X1
    Z2  $\leftarrow$  X2
[A]  Z3  $\leftarrow$  Z1 % Z2
    Z1  $\leftarrow$  Z2
    Z2  $\leftarrow$  Z3

```

```
IF Z3 != 0 GOTO A  
Y <- Z2
```