

Modelos Avanzados de Computación

Entrega 2

María del Mar Ruiz Martín
Doble Grado de Ingeniería Informática y Matemáticas
Universidad de Granada - UGR
18001 Granada, Spain
Curso 2016/2017



Ejercicio 2

Construir un programa Post-Turing que dado un número u en binario calcule $u+1$.

La idea para resolver este ejercicio será la siguiente: en primer lugar nos desplazaremos al final de la palabra u (mediante el conjunto de instrucciones que comienzan por la etiqueta $[A]$). A continuación pasaremos a buscar el primer símbolo distinto de 1. Si encontramos un símbolo diferente de 1, esto es, 0 o #, lo cambiaremos por un 1 y terminará el programa. En caso de encontrarnos con un 1 lo cambiaremos por un cero, nos movemos a la izquierda y procederemos de forma análoga. Esto último se corresponde con el bloque $[C]$. El programa sería el siguiente:

```
[A] RIGHT
    IF # GOTO C
    IF 0 GOTO A
    IF 1 GOTO A
[B] PRINT 0
[C] LEFT
    IF 1 GOTO B
    PRINT 1
    HALT
```

Ejercicio 4

Construir un programa con variables que concatene dos cadenas sobre $\{0,1\}$. Se supone que ambas cadenas están en las variables X_1 y X_2 y la salida en la variable Y .

Para resolver este problema vamos a implementar una macro auxiliar. En dicha macro la variable a_0 se corresponderá con el 0 y a_1 con el 1. i toma valores en $\{0,1\}$

```
Macro  $U \leftarrow VU$ :
     $Z \leftarrow \epsilon$ 
[A] IF  $V$  ENDS  $a_i$  GOTO  $B_i$ 
    GOTO C
[ $B_i$ ]  $V \leftarrow V-$ 
     $U \leftarrow a_i U$ 
     $Z \leftarrow a_i Z$ 
    GOTO A
[ $D_i$ ]  $Z \leftarrow Z-$ 
     $V \leftarrow a_i V$ 
    GOTO C
[C] IF  $Z$  ENDS  $a_i$  GOTO  $D_i$ 
```

Debemos notar que la macro termina tras la instrucción $[C]$ IF Z ENDS a_i GOTO D_i ($i=1,\dots,n$), dando paso a la siguiente instrucción a la que invocó a la macro. Supondremos que nuestro programa con variables cuenta con las variables de entrada X_1, X_2 , con las variables de trabajo Z, y V y con la variable de salida Y. Con estas variables y haciendo uso de la macro anterior el programa resultante quedaría como sigue:

```

Y ←  $X_2 Y$ 
Y ←  $X_1 Y$ 
HALT

```

Ejercicio 7

Construir un programa con variables que dada una cadena $u \in \{0,1\}^*$ calcule la cadena w formada por los símbolos que ocupan las posiciones impares de u y en el mismo orden que aparecen en u.

La idea para resolver este problema será la siguiente: obtendremos dos cadenas nuevas, W_1 y W_2 a partir de las letras de u. Iremos alternandolas, de forma que al final del proceso una de ellas tendrá los símbolos que ocupan una posición par y la otra los que tienen posición impar. Sin embargo, no sabremos cuál es cuál hasta el final. En el momento en que u se quede vacía, la última cadena a la que se añadió una letra es la que contiene los símbolos de posiciones impares, y nos quedaremos con esta. Por tanto, en cada paso añadimos la letra final de u a la cadena actual, borramos el último símbolo de u y pasamos a repetir el proceso con la cadena contraria.

Para realizar el programa contaremos con una variable de entrada, U, dos variables de trabajo, W_1 y W_2 y la variable de salida W. i tomará los valores 1 y 2 (representando la cadena activa) y j tomará valores en $\{0,1\}$, indicando qué valor se ha leído. Notaremos por \bar{i} al valor contrario al que tiene i, esto es, 1 si $i=2$ o 2 si $i=1$. Haremos uso de la macro vista en teoría $U \leftarrow V$.

```

IF U ENDS j GOTO  $A_{1,j}$ 
HALT
 $[A_{i,j}]$   $W_i \leftarrow j W_i$ 
 $U \leftarrow U -$ 
IF U ENDS  $\epsilon$  GOTO  $[C_i]$ 
IF U ENDS j GOTO  $A_{\bar{i},j}$ 
HALT
 $[C_i]$   $W \leftarrow W_i$ 

```

Ejercicio 8

Construir un programa con variables sobre $\{a,b\}$ que dadas dos cadenas $u_1, u_2 \in \{a,b\}^*$ calcule la cadena u cuyo número verifica $Z(u) = Z(u_1) + Z(u_2)$ (es decir, hacer la suma de números representados por cadenas de caracteres sobre $\{a,b\}$)

Para resolver este ejercicio contaremos con dos macros nuevas: una para sumar 1 y otra para restar 1. El procedimiento para resolver el problema será el siguiente: mientras que $Z(u_2)$ sea no nula le restaremos uno y le sumaremos uno a u_1 . Así, en $Z(u_2)$ iteraciones, conseguimos la cadena u que verifica $Z(u) = Z(u_1) + Z(u_2)$. En la definición de las macros entenderemos que $a_1 = a$ y $a_2 = b$.

Macro para sumar 1 ($U \leftarrow U + 1$):

```
[B] IF X ENDS  $a_i$  GOTO  $A_i$  (i=1,2)
     $Y \leftarrow a_1 Y$ 
    HALT
[ $A_1$ ]  $X \leftarrow X-$ 
     $Y \leftarrow a_2 Y$ 
    GOTO C
[ $A_2$ ]  $X \leftarrow X-$ 
     $Y \leftarrow a_1 Y$ 
    GOTO B
[C] IF X ENDS  $a_i$  GOTO  $D_i$  (i=1,2)
    HALT
[ $D_i$ ]  $X \leftarrow X-$  (i=1,2)
     $Y \leftarrow a_i Y$ 
    GOTO C
```

Macro para restar 1 ($U \leftarrow U - 1$):

```
[B] IF X ENDS  $a_i$  GOTO  $A_i$  (i=1,2)
    HALT
[ $A_2$ ]  $X \leftarrow X-$ 
     $Y \leftarrow a_1 Y$ 
    GOTO C
[ $A_1$ ]  $X \leftarrow X-$ 
    IF  $X \neq \epsilon$  GOTO  $C_2$ 
    HALT
[ $C_2$ ]  $Y \leftarrow a_2 Y$ 
    GOTO B
[C] IF X ENDS  $a_i$  GOTO  $D_i$  (i=1,2)
    HALT
[ $D_i$ ]  $X \leftarrow X-$  (i=1,2)
     $Y \leftarrow a_i Y$ 
```

GOTO C

Además usaremos la macro vista en clase para $U \leftarrow V$. Nuestro programa con variables contará con dos variables de entrada, U_1, U_2 , una variable de salida, U , y las variables de trabajo X e Y . Haciendo uso de las macros anteriores el programa tendría la siguiente forma:

```
     $U \leftarrow U_1$   
[A] IF  $X$  ENDS  $\epsilon$  GOTO B  
     $U \leftarrow U + 1$   
     $U_2 \leftarrow U_2 - 1$   
    GOTO A  
[B] HALT
```