

### Ejercicio 4.1

Desarrolle una máquina de Turing que calcule el siguiente de un número natural descrito en notación binaria de izquierda a derecha. Por ejemplo, si la entrada fuera el número 13 la cinta tendría inicialmente el contenido (# 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...) y la salida debería ser (# 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...).

### Ejercicio 4.2

Desarrolle una máquina de Turing que reciba como entrada una lista de números en notación binaria separados por el símbolo \$ y genere como salida el tercer número de la lista. Por ejemplo, si la entrada es (# 0 1 \$ 1 1 0 1 \$ 0 1 1 \$ 0 \$ 1 0 1 1 1 1 1 1 1 1 ...) la salida a generar debe ser (# 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...).

### Ejercicio 4.3 → Diferentes

Desarrolle una máquina de Turing que compruebe que dos cadenas separadas por el símbolo \$ son iguales, es decir, la máquina debe aceptar entradas de la forma “#01001\$01001111...”.

### Ejercicio 4.4

Desarrolle una Máquina de Turing que reconozca el lenguaje formado por cadenas de 0 cuya longitud sea potencia de dos:

$$L = \{0^{2^n}; n \geq 0\}$$

NOTA: Para comprobarlo hay que hacer varias pasadas a la cadena eliminando la mitad de 0's en cada pasada hasta alcanzar una cadena de longitud 1. Si en alguna pasada el número de 0's es impar hay que rechazar la cadena.

### Ejercicio 4.5

Desarrolle una Máquina de Turing que reconozca el siguiente lenguaje:

$$L = \{a^n b^n c^n; n \geq 0\}$$

### Ejercicio 4.6

Desarrolle una Máquina de Turing que calcule la suma de dos números en formato binario. Los números se expresarán de izquierda a derecha. Se utilizará el símbolo “#” para marcar el comienzo de la cinta, la separación de los sumandos y el final de la entrada. La máquina debe generar como salida el valor de la suma, escrito de izquierda a derecha.

Por ejemplo, el número 13 se escribe en binario como “1101” y el número 20 se escribe como “10100”. Para sumar 13+20 la cadena de entrada debe ser “#1011#00101#” y la de salida debe ser “#100001”.

### Ejercicio 4.7

El siguiente algoritmo permite reconocer el lenguaje  $L = \{ 0^k 1^k \mid k \geq 0 \}$

- 1.- Recorrer la cinta. Si se encuentra un 0 a la derecha de un 1, **rechazar**.
- 2.- Repetir mientras queden 0s y 1s:
- 3.- Recorrer la lista verificando si el número de 0s y 1s es par o impar.  
Si es impar, **rechazar**.
- 4.- Recorrer la lista marcando (quitando) la mitad de 0s y la mitad de 1s.
- 5.- Si no quedan 0s ni 1s, **aceptar**. En otro caso, **rechazar**.

Desarrolle una Máquina de Turing que implemente este algoritmo.

### Ejercicio 4.8

Diseñar una Máquina de Turing que haga una copia de una cadena de símbolos  $\{A,B,C\}$ . Por ejemplo, para la entrada “#AABCA~~bbb~~...” devuelve en la cinta “#AABCAAABCA~~bb~~..”.

NOTA: Tenga en cuenta que no existe ningún espacio entre la cadena inicial y la copia.

### Ejercicio 4.9

Diseñar una Máquina de Turing que tome como entrada una palabra formada por los símbolos del alfabeto  $\{a,b\}$  y devuelve la longitud de la palabra expresada en código binario. Por ejemplo, para la entrada (#ababb~~bb~~) devuelve el número 6 (#011~~bb~~).

NOTA: El número binario está escrito de izquierda a derecha, es decir, la cifra menos significativa a la izquierda.

### Ejercicio 4.10

Diseñar una Máquina de Turing que tome como entrada una palabra formada por los símbolos del alfabeto  $\{A,B\}$  y devuelve la misma palabra escrita en orden inverso. Por ejemplo, para la entrada (#ABABB~~bb~~) devuelve (#BBABA~~bb~~).