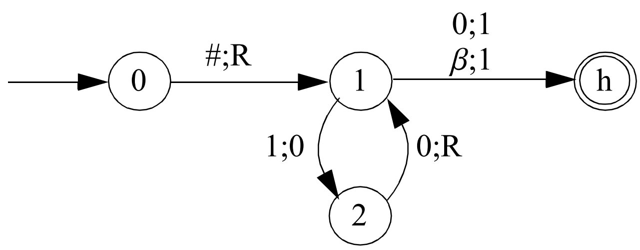
Modelos Avanzados de Computación

Ejercicios del Tema 4

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 $\beta\beta\beta$ …) y la salida debería ser (# 0 1 1 1 $\beta\beta\beta$ …).

SOLUCIÓN:

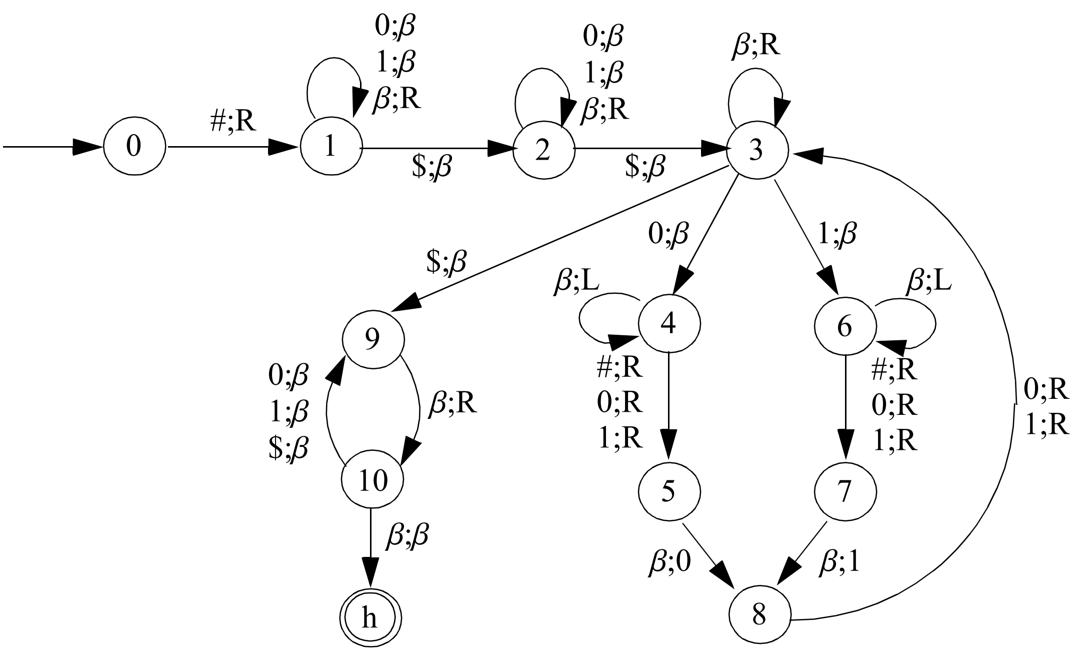


- La máquina comienza leyendo el símbolo # y desplazando el cabezal a la derecha para pasar al estado de incrementar (estado 1).
- Si la máquina lee un 0 lo sustituye por un 1 y para.
- Si se encuentra un 1 debe sustituirlo por un 0 e incrementar la siguiente cifra. Para ello escribe el 0, lo lee y se desplaza a la derecha volviendo al estado de incremento. De esta forma si la máquina lee "#1110001β" lo transforma en "#0001001β"
- Si la máquina sigue teniendo que incrementar y encuentra un blanco, escribe un 1 y para. De esta forma se convierte el valor "#11$\beta\beta$" en "#001β".

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 *β β β* …) la salida a generar debe ser (# 0 1 1 *β β β* …).

SOLUCIÓN:



La máquina funciona en tres fases:

a) En primer lugar borra los dos primeros números. Esto transforma la entrada "#**01$1101$**011$0$101*βββ*..." en "#***ββββββββ***011$0$101*βββ*...".

b) A continuación copia el tercer número al comienzo, sustituyendo cada dígito por un blanco mientras se copia. Esto transforma la entrada "#*ββββββββ***011**$0$101*βββ*..." en "#**011***ββββββββ*$0$101*βββ*...".

c) Para terminar borra el resto de la cinta. Esto transforma la entrada "#011*ββββββββ*$0$101*βββ*..." en "#011*ββββββββββββββββ*...".

La descripción con más detalle sería la siguiente:

- La máquina comienza leyendo el símbolo # y desplazandose a la derecha.
- En el estado 1 se borra el primer número de la entrada y se salta al estado 2 borrando el símbolo $ de separación.
- En el estado 2 se borra el segundo numero de la entrada y se salta al estado 3 borrando el símbolo $ de separación.
- En el estado 3 se lee un dígito del número a copiar. Si el dígito es el 0 se borra y se desplaza el cabezal a la izquierda hasta encontrar el último dígito copiado (estado 4), se

desplaza el cabezal a la derecha (estado 5) y se escribe el 0. Si desde el estado 3 se lee un 1, se borra, se desplaza el cabezal a la izquierda hasta encontrar el último dígito copiado (estado 6), se desplaza el cabezal a la derecha (estado 7) y se escribe un 1.

- En el estado 8 la máquina acaba de copiar un dígito y el cabezal se encuentra sobre él. Desde este estado se desplaza el cabezal a la derecha y se salta al estado 3. En el estado 3 se recorren todos los blancos a la derecha hasta encontrar el siguiente dígito a copiar y se vuelve a ejecutar el punto anterior.

- Desde el estado 3, cuando se ha terminado de copiar todos los dígitos del tercer número se lee un $. En ese caso ha que borrar lo que queda de la cadena de entrada. Para ello se borra el $ y se hace un recorrido de la lista (estados 9 y 10) borrando todos los símbolos hasta llegar al final.

Ejercicio 4.3

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\$01001 $\beta\beta$ ...".

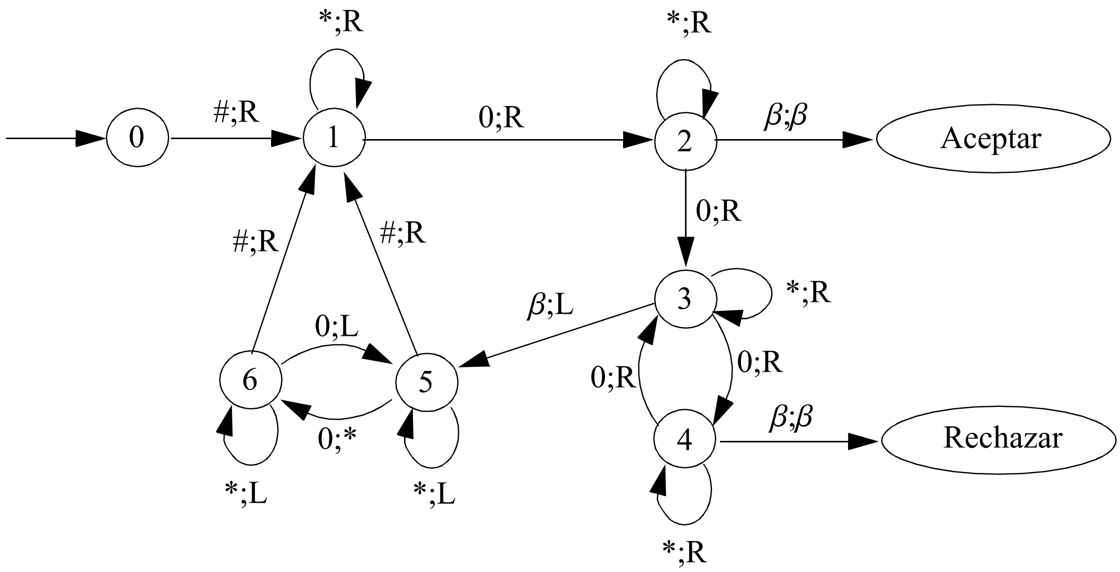
Ejercicio 4.4

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

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

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.

SOLUCIÓN:



Vamos a suponer que la cadena comienza con el símbolo # para marcar el principio. El comportamiento de la máquina consiste en realizar pasadas a la derecha contando el número de ceros. Si el número de ceros es 1 la cadena se acepta. Si el número de ceros es impar la cadena se rechaza. Si el número de ceros es par se recorre la cadena hacia atrás eliminando la mitad de ceros y se repite el estudio.

- El primer paso es leer el símbolo # y desplazarse a la derecha para comenzar a leer la cadena de entrada.

- En el estado 1 se recorre la cadena hacia la derecha desplazándose antes las marcas *. Cuando se lee el primer cero se pasa al estado 2.

- El estado 2 refleja que se ha leído un único cero. En este estado se sigue recorriendo la cadena leyendo las marcas. Si se llega al final de la cadena, se acepta. Si se encuentra un cero hay que estudiar la paridad y se desplaza al estado 3.

- El estado 3 refleja que el número de ceros es par. En este estado se recorre la cadena a la derecha consumiendo marcas. Si se encuentra un 0 es que el número de ceros leidos es impar y se desplaza al estado 4. Si se encuentra el final de la cadena es que quedan un número par de ceros y hay que saltar al estado 5 para eliminar la mitad de ceros.

- El estado 4 refleja que el número de ceros leídos es impar. Si se encuentra el final de la cadena hay que rechazar la entrada. Si se encuentra otro cero es que el numero de ceros leídos es par y se desplaza al estado 3.

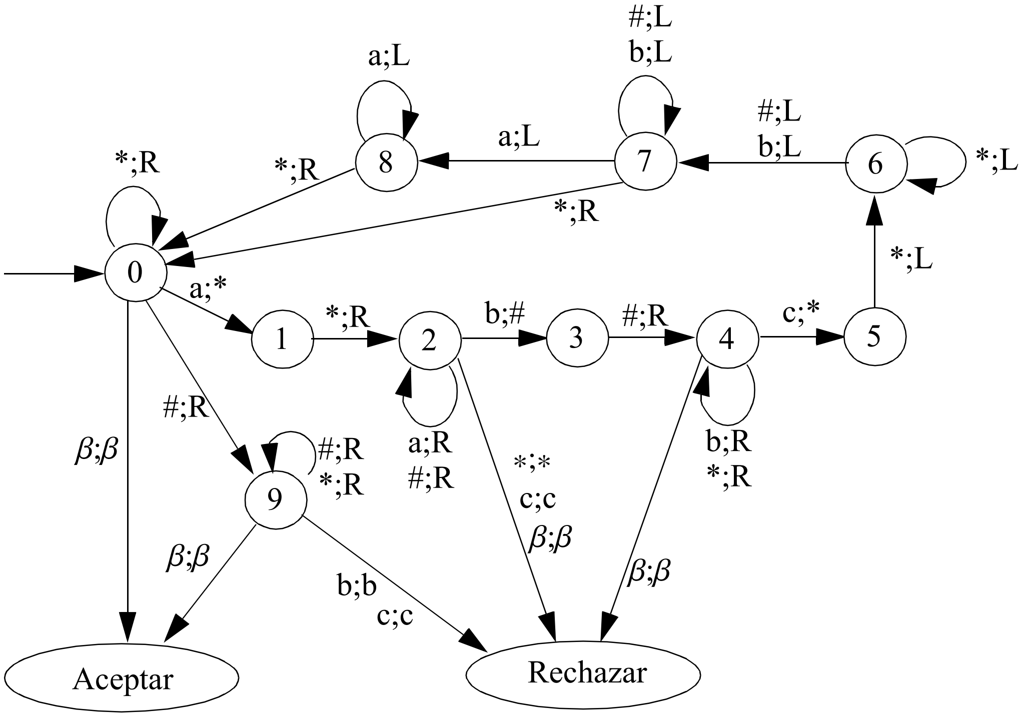
- En los estados 5 y 6 se recorre la cadena hacia atrás eliminando la mitad de ceros. En el estado 5 al encontrar un cero se marca y se pasa al estado 6. En el estado 6 al encontrar un cero se deseplaza a la izquierda dejando el cero y se pasa al estado 5. Tanto desde el estado 5 como desde el 6, si se encuentra el inicio de la cadena se salta al estado 1 y se vuelve a ejecutar el algoritmo.

Ejercicio 4.5

Desarrolle una Máquina de Turing que reconozca el siguiente lenguaje:

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

SOLUCIÓN:



El funcionamiento de la máquina se basa en hacer recorridos de la cadena marcando en cada recorrido una ‘a’, una ‘b’ y una ‘c’ y retroceder hasta la última ‘a’ marcada. Para distinguir las letras marcadas vamos a utilizar un caracter ‘*’ para marcar la ‘a’ y la ‘c’ y un carácter ‘#’ para marcar la ‘b’. Si en un recorrido se llega al final de la cadena sin marcar ninguna letra es porque todas han sido marcadas y el número de ‘a’, ‘b’y ‘c’coincide por lo que hay que aceptar la cadena. Si en algún recorrido no se pueden marcar las tres letras la cadena se rechaza.

- En el estado 0 nos situamos al comienzo de la cadena. Si se lee el caracter blanco es que se trata de la cadena vacía y se debe aceptar. Como vamos a marcar las ‘a’ con el símbolo ‘*’ desde este estado avanzamos en la cadena saltándonos todos las marcas. Si se lee un símbolo ‘a’ hay que emparejarlo con una ‘b’ y una ‘c’ por lo que se marca la ‘a’ y se salta al estado 1

- En el estado 1 se acaba de marcar una ‘a’. En este estado se lee la marca y se avanza al estado 2 desde el que se recorre la cadena buscando un carácter ‘b’. Si no se encuantra ninguna ‘b’ hay que rechazar la cadena. Si se encuentra una ‘b’ se marca y se salta al estado 3.

- En el estado 3 se acaba de marcar una ‘b’. En este estado se lee la marca y se avanza al estado 4. Desde ese estado se recorre la cadena buscando emparejar una ‘c’. Si no se encuentra la ‘c’ hay que rechazar la cadena. Si se encuentra la ‘c’ se marca y se salta al estado 5.

- En el estado 5 se acaban de emparejar una ‘a’, una ‘b’ y una ‘c’. Ahora hay que retroceder en la cadena desplazándose a la izquierda. Hay que recorrer las marcas de las ‘c’ (estado 6), las ‘b’ que puedan quedar y las marcas de las ‘b’ (estado 7) y las ‘a’ que puedan quedar hasta la última marca de las ‘a’ (estado 8).

- Al estado 9 se llega si después de emparejar varias veces las letras ya no quedan ‘a’. En este caso desde el estado 0 se leerá una marca de ‘b’. En este punto hay que comprobar que ya solo hay marcas pero no queda ninguna ‘b’ ni ninguna ‘c’. Si es así se acepta la cadena. Si existe alguna ‘b’ o alguna ‘c’ hay que rechazar la cadena.

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 | k ≥ 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~~bbbb~~...” 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 (#ababba~~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~~).