

Teoría de la Computación 2022

Lab 08

26.octubre.2022

1. Diseñar máquinas de Turing para resolver los siguientes problemas:

a) Aceptar las cadenas del alfabeto $\{a, b, c\}$ que contengan a la subcadena aaa .

b) Hacer un corrimiento a la derecha *right-shift* de la cadena input (cadenas binarias).

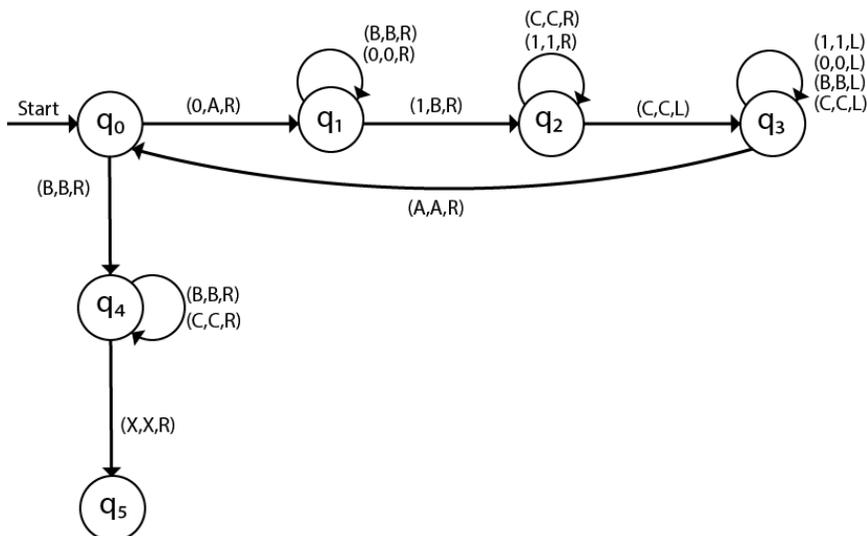
c) Duplicar la cadena input (cadenas binarias).

d) Restar dos números enteros positivos en formato unario.

En este caso el input es de la forma $1^* - 1^*$, y se asume que el primer número siempre es mayor o igual que el segundo.

2. En clase discutimos que el lenguaje $L_1 = \{0^n 1^n : n \geq 1\}$ no es regular, pero es libre de contexto. También vimos que el lenguaje $L_2 = \{0^n 1^{n^2} : n \geq 1\}$ ya no es libre de contexto, de modo que no puede ser producido mediante un autómata de pila.

Sin embargo, L_2 si es recursivamente enumerable, y que puede generarse mediante una máquina de Turing. Verifique que la siguiente máquina genera a L_2 :



Nota, aquí el alfabeto input es $\Sigma = \{0, 1, 2\}$, el alfabeto de cinta es $\Gamma = \{0, 1, 2, A, B, C, X\}$, y el símbolo en blanco es X .

3. Construir una máquina de Turing que verifique si una cadena binaria ($\Sigma = \{0, 1\}$) es un palíndromo, para dos casos por separado:

a) Cuando la longitud del input es par,

b) Cuando la longitud del input es impar.

Explique qué se debería hacer si quisieramos una máquina de Turing que verifique si la cadena binaria input es un palíndromo, independientemente si su longitud es par o impar.