Algoritmia y Complejidad

Grupo 01 - Leibniz

Integrantes:

- **María del Carmen López Gómez**
- **Claudia Paola García Nocetti**
- **Javier González Rodríguez**
- **Marcos Hidalgo Baños**

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA

Análisis del lenguaje

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

➔ Ejemplos de cadenas aceptadas:

0101#0101	110#110	01#01	0#0	1#1	#

➔ Ejemplos de cadenas rechazadas:

1001#11	101#100	101#010	110100	#1010#	##

Máquina de Turing Unicinta

➔ Formalización de la Máquina de Turing $\boldsymbol{M}$ que decide $\mathscr{L}$

Σ	Alfabeto de la cadena de entrada (input)	$\Sigma = \{0, 1, \#\}$
Γ	Alfabeto de la cinta en la Máquina de Turing	$\Gamma = \{0, 1, \#, x, ƀ\}$
Q	Conjunto de estados de la Máquina de Turing	Q = {q0, … , q7, q_acc, q_rej}
δ	Reglas de transición entre estados	Descritas en la siguiente diapositiva

Máquina de Turing Unicinta

➔ Listado de las reglas de transición entre estados ***δ***

δ(q0,0) ⟶ (q1,x,R)
δ(q0,1) ⟶ (q2,x,R)
δ(q0,#) ⟶ (q7,R)

δ(q1,0/1) ⟶ (q1,R)
δ(q1,#) ⟶ (q3,R)

δ(q2,0/1) ⟶ (q2,R)
δ(q2,#) ⟶ (q4,R)

δ(q3,x) ⟶ (q3,R)
δ(q3,0) ⟶ (q5,x,L)

δ(q4,x) ⟶ (q4,R)
δ(q4,1) ⟶ (q5,x,L)

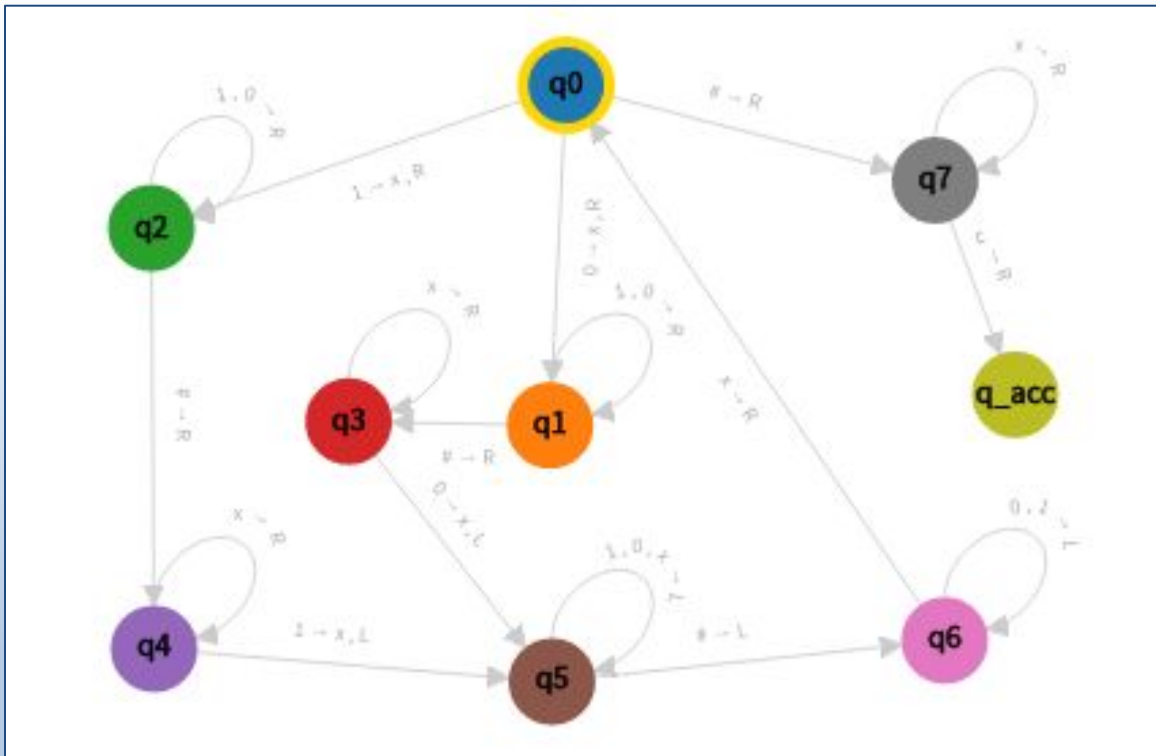
δ(q5,0/1/x) ⟶ (q5,L)
δ(q5,#) ⟶ (q6,L)

δ(q6,0/1) ⟶ (q6,L)
δ(q0,x) ⟶ (q0,R)

δ(q7,x) ⟶ (q7,R)
δ(q7,ƀ) ⟶ (q_acc,ƀ,R)

Máquina de Turing Unicinta

$$L = \{ w \# w \mid w \in \{0,1\}^* \}$$




```
input: '#'
blank: ' '
start state: q0
table:
  q0:
    [0]: {write: x, R: q1}
    [1]: {write: x, R: q2}
    '#': {R: q7}
  q1:
    [1,0]: R
    '#': {R: q3}
  q2:
    [1,0]: R
    '#': {R: q4}
  q3:
    [x]: R
    [0]: {write: x, L: q5}
  q4:
    [x]: R
    [1]: {write: x, L: q5}
  q5:
    [1,0,x]: L
    '#': {L: q6}
  q6:
    [0,1]: L
    [x]: {R: q0}
  q7:
    [x]: R
    ' ': {R: q_acc}
  q_acc:
```

Diseño de la Máquina de Turing

UNIVERSIDAD DE MÁLAGA

LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN UNIVERSIDAD DE MÁLAGA

➔ Nos encontramos tres zonas destacadas:

- <u>Ciclo "**w** empieza en 1".</u> (q0 - q2 - q4 - q5 - q6 - q0)

  Sabemos que si la subcadena **w** empieza por 1, necesariamente su homóloga tras el carácter **#** también lo debe hacer. Procedemos a buscar esa pareja por lo que obviamos todo tipo de carácter de Γ que no se encuentre tras **#**.

  Una vez localizado, hemos de obviar los caracteres **x** que seguramente nos encontremos en las sucesivas iteraciones del ciclo. Condicionamos que el siguiente carácter sea un 1 mediante una única transición entre q4 y q5.

  Durante las transiciones entre q5 - q6 - q0 realizamos el viaje de vuelta hacia la celda inmediatamente a la derecha de la original sin realizar escrituras.

- <u>Ciclo "**w** empieza en 0".</u> (q0 - q1 - q3 - q5 - q6 - q0)

  Equivalente de la idea anterior aplicado a las cadenas que empiezan por 0.

- <u>Rama de aceptación.</u> (q0 - q7 - q_acc)

  Accederemos únicamente tras haber procesado ambas subcadenas **w**.
  Comprobaremos que el carácter **#** se encuentra solo o seguido de **x** || ƀ

Complejidad Temporal

➔ Estimación del Orden de Complejidad de $\boldsymbol{M}$:

1	0	1	#	1	0	1
x	0	1	#	**x**	0	1
x	**x**	1	#	**x**	**x**	1
x	**x**	**x**	#	**x**	**x**	**x**

n = 2w+1
w = (n-1)/2

TIEMPO EMPLEADO = w·(2(w+1)+1)

Conclusión

Sea *t(n)* la función que mide el tiempo necesario para $\boldsymbol{M}$ en determinar si una cadena de tamaño *n* pertenece o no a $\mathscr{L}$:

$$t(n) = ((n-1)/2) \cdot (2((n-1)/2)+1)+1) \in O(n^2)$$

Propuesta Versión Multicinta

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

➔ Cinta 1: Lectura de datos

1	0	1	#	1	0	1

$\underbrace{\quad 1 \quad 0 \quad 1 \quad}_{\mathbf{w}}$

➔ Cinta 2: Copia de la subcadena **w**

1	0	1				

Proceso de aceptación de la cadena

Una vez recorrida la entrada hasta el carácter **#** se procede a comparar el contenido de la celda apuntada por la cabeza lectora de la Cinta 1 con su equivalente en la Cinta 2. Si se detecta discrepancia se transita al estado de rechazo. Si ambas celdas contienen el carácter **ƀ** se acepta la cadena.