UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 7 – Máquinas de Turing

Manuel Pegalajar Cuéllar
manupc@ugr.es

Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es

Objetivos del tema

- Entender el concepto de lenguaje recursivo y recursivamente enumerable.
- Entender el concepto de Máquina de Turing.
- Saber realizar programas en Máquinas de Turing.

Anotación sobre estas diapositivas:

El contenido de estas diapositivas es esquemático y representa un apoyo para las clases presenciales teóricas. No se considera un sustituto para apuntes de la asignatura.

Se recomienda al alumno completar estas diapositivas con notas/apuntes propios, tomados en clase y/o desde la bibliografía principal de la asignatura.

1. Introducción
2. Máquinas de Turing
3. Lenguajes recursivamente enumerables
4. Construcción de máquinas de Turing
5. Extensiones del modelo de MT

El problema de la parada

- **Entrada**: Un programa junto con sus datos de entrada
- **Salida:** SI (si el programa de entrada termina para esos datos), o NO (cuando no lo hace).

¿Por qué es difícil saber si un programa termina?

```
int exp(int i,n)
/* calcula i a la potencia n */
{
    int ans,j;
    ans = 1;
    for (j=1; j<=n; j++) ans *= i;
    return(ans);
}

main()
{
    int n, total, x,y,z;
    scanf("%d",&n);
    total = 3;
    while (1) {
        for(x=1; x<=total-2; x++)
            for(y=1; y<=total-x-1; y++) {
                z = total-x-y;
                if (exp(x,n) + exp(y,n) == exp(z,n))
                    printf("hola mundo");
            }
        total++;
    }
}
```

¿Por qué es difícil saber si un programa termina?

- El programa muestra "Hola, mundo" para una entrada *n* sólo si existen números enteros positivos x, y, z tales que $x^n+y^n=z^n$

Según la demostración del último teorema de Fermat, esto no es posible para n>2

Se ha tardado 300 años (desde que se planteó el teorema) hasta hoy demostrar que un único programa (el anterior) no muestra "Hola, mundo"…

… Pero nosotros queremos hacer un programa que nos diga si ***cualquier programa termina***.

Es un **problema indecidible. Nos ayudaremos de máquinas de Turing para demostrarlo**

1. Introducción
2. Máquinas de Turing
3. Lenguajes recursivamente enumerables
4. Construcción de máquinas de Turing
5. Extensiones del modelo de MT

Máquinas de Turing

Una Máquina de Turing (MT) es una sextupla (Q, A, B, δ, q_0, #, F) tal que:

- Q es un conjunto finito de estados.
- A es un alfabeto de entrada.
- B es un alfabeto de símbolos de la cinta, e incluye a A.
- δ es una función de transición que asigna a cada estado q en Q y símbolo b en B el valor δ(q,b), que puede ser vacío (no definido) o una tripleta (p,c,M), donde p es un estado en Q, c es un símbolo de B y M un valor en {I,D}, donde I=Izq y D=Dcha.
- q_0 es el estado inicial.
- # es un símbolo de B\A llamado “blanco”.
- F es el conjunto de estados finales.

Un ejemplo

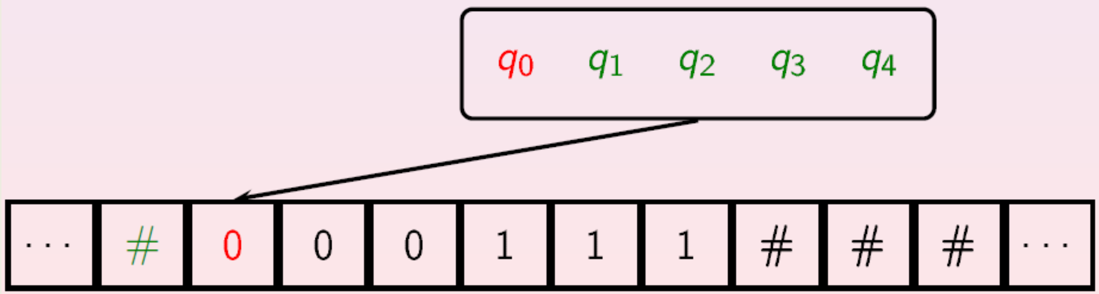
Consideremos la MT $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, \#\}, \delta, q_0, \#, \{q_4\})$ donde las transiciones no nulas son las siguientes:

$$
\begin{array}{ll}
\delta(q_0, 0) = (q_1, X, D) & \delta(q_0, Y) = (q_3, Y, D) \\
\delta(q_1, 0) = (q_1, 0, D) & \delta(q_1, 1) = (q_2, Y, I) \\
\delta(q_1, Y) = (q_1, Y, D) & \delta(q_2, 0) = (q_2, 0, I) \\
\delta(q_2, X) = (q_0, X, D) & \delta(q_2, Y) = (q_2, Y, I) \\
\delta(q_3, Y) = (q_3, Y, D) & \delta(q_3, \#) = (q_4, \#, D)
\end{array}
$$

¿Es la palabra 000111 aceptada?

Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



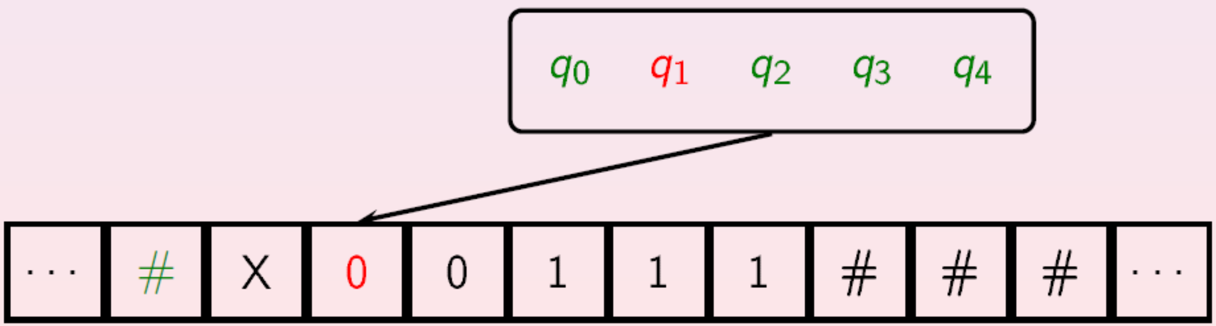
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\delta(q_0,Y)=(q_3,Y,D)$ $\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\delta(q_1,Y)=(q_1,Y,D)$ $\delta(q_2,0)=(q_2,0,I)$

$\delta(q_2,X)=(q_0,X,D)$ $\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\delta(q_3,\#)=(q_4,\#,D)$



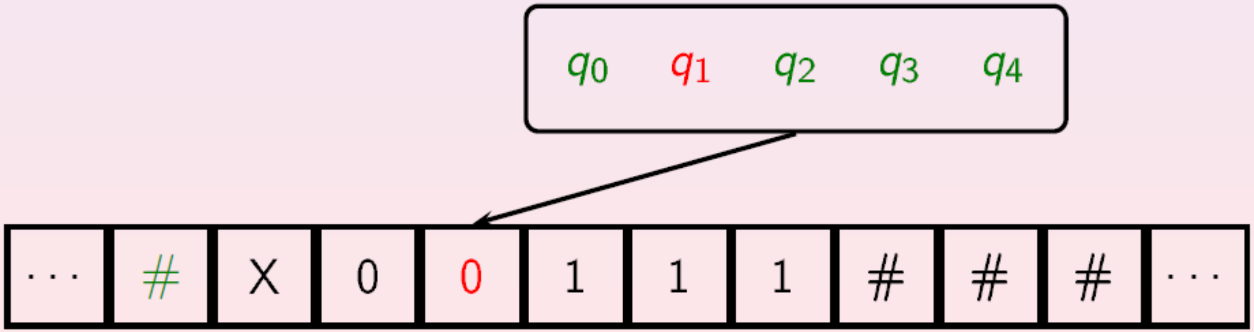
Ejemplo de funcionamiento

$\delta(q_0,0) = (q_1, X, D)$ $\quad$ $\delta(q_0,Y) = (q_3, Y, D)$ $\quad$ $\delta(q_1,0) = (q_1, 0, D)$

$\delta(q_1,1) = (q_2, Y, I)$ $\quad$ $\delta(q_1,Y) = (q_1, Y, D)$ $\quad$ $\delta(q_2,0) = (q_2, 0, I)$

$\delta(q_2,X) = (q_0, X, D)$ $\quad$ $\delta(q_2,Y) = (q_2, Y, I)$

$\delta(q_3,Y) = (q_3, Y, D)$ $\quad$ $\delta(q_3,\#) = (q_4, \#, D)$



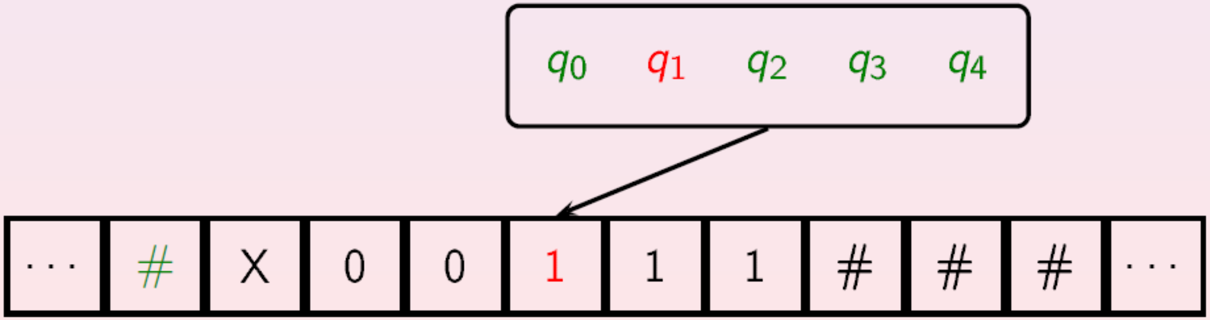
Ejemplo de funcionamiento

$\delta(q_0,0) = (q_1,X,D)$ $\quad$ $\delta(q_0,Y) = (q_3,Y,D)$ $\quad$ $\delta(q_1,0) = (q_1,0,D)$

$\delta(q_1,1) = (q_2,Y,I)$ $\quad$ $\delta(q_1,Y) = (q_1,Y,D)$ $\quad$ $\delta(q_2,0) = (q_2,0,I)$

$\delta(q_2,X) = (q_0,X,D)$ $\quad$ $\delta(q_2,Y) = (q_2,Y,I)$

$\delta(q_3,Y) = (q_3,Y,D)$ $\quad$ $\delta(q_3,\#) = (q_4,\#,D)$



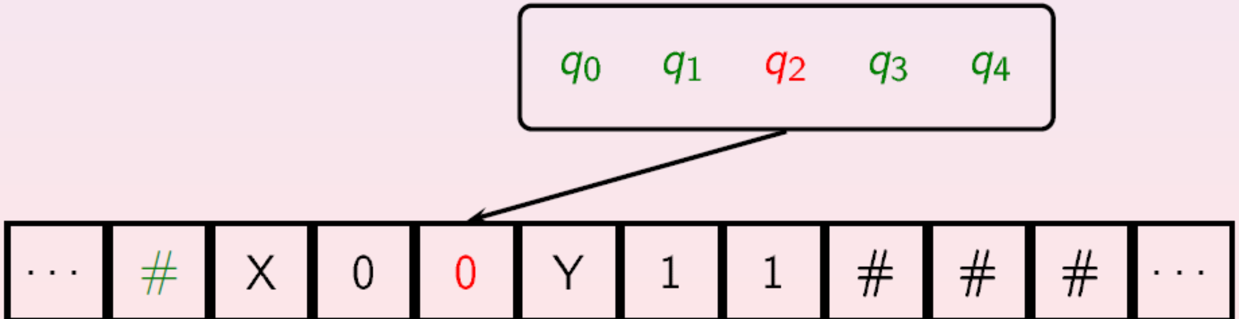
Ejemplo de funcionamiento

$\delta(q_0, 0) = (q_1, X, D)$ $\quad$ $\delta(q_0, Y) = (q_3, Y, D)$ $\quad$ $\delta(q_1, 0) = (q_1, 0, D)$

$\delta(q_1, 1) = (q_2, Y, I)$ $\quad$ $\delta(q_1, Y) = (q_1, Y, D)$ $\quad$ $\delta(q_2, 0) = (q_2, 0, I)$

$\delta(q_2, X) = (q_0, X, D)$ $\quad$ $\delta(q_2, Y) = (q_2, Y, I)$

$\delta(q_3, Y) = (q_3, Y, D)$ $\quad$ $\delta(q_3, \#) = (q_4, \#, D)$



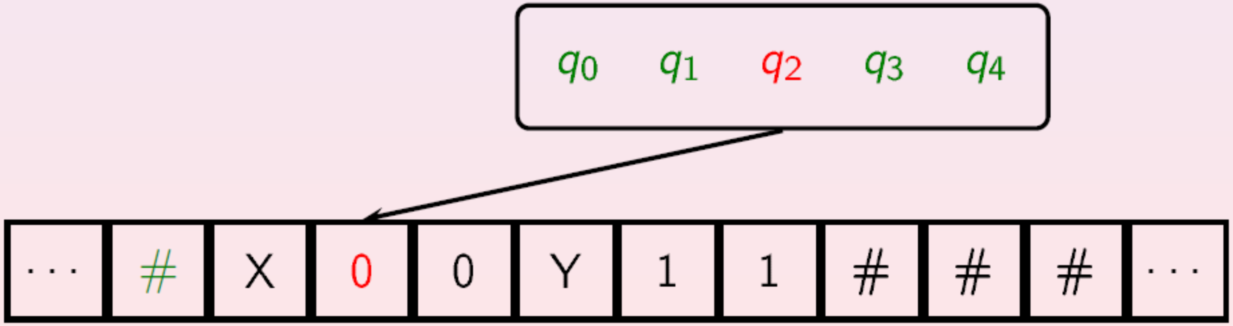
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\quad \delta(q_0,Y)=(q_3,Y,D)$ $\quad \delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\quad \delta(q_1,Y)=(q_1,Y,D)$ $\quad \delta(q_2,0)=(q_2,0,I)$

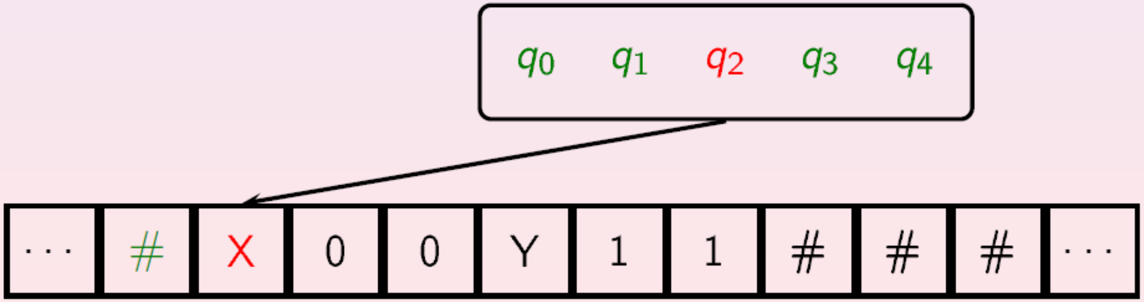
$\delta(q_2,X)=(q_0,X,D)$ $\quad \delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\quad \delta(q_3,\#)=(q_4,\#,D)$



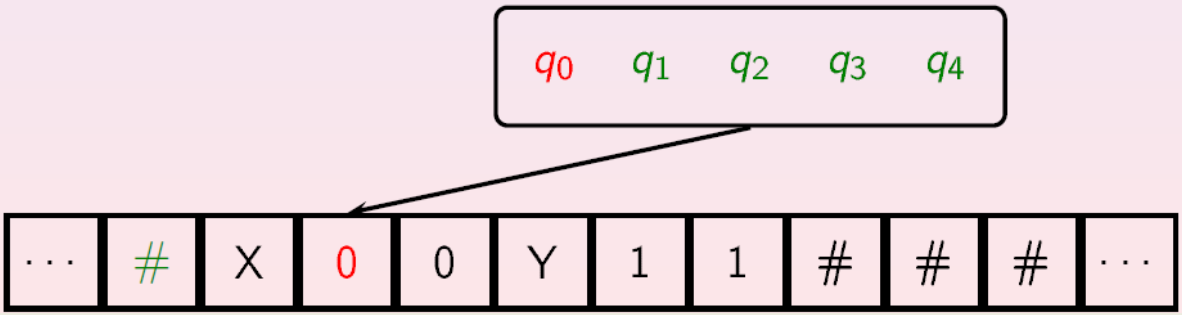
Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



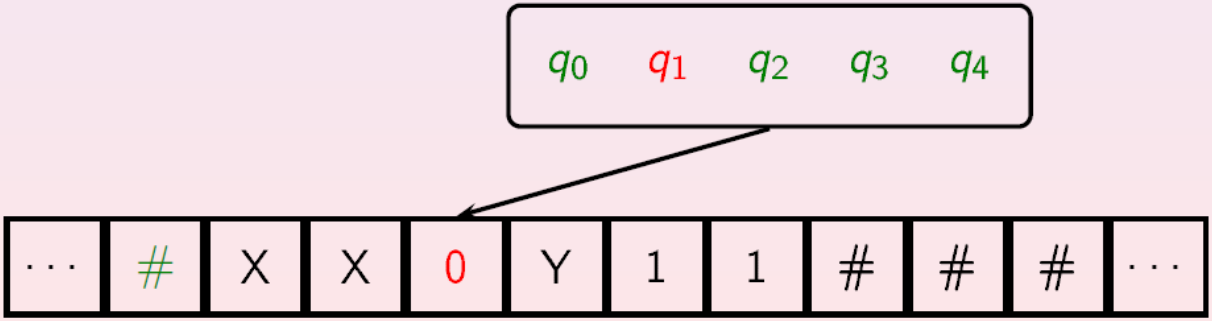
Ejemplo de funcionamiento

$\delta(q_0, 0) = (q_1, X, D)$ $\quad \delta(q_0, Y) = (q_3, Y, D)$ $\quad \delta(q_1, 0) = (q_1, 0, D)$

$\delta(q_1, 1) = (q_2, Y, I)$ $\quad \delta(q_1, Y) = (q_1, Y, D)$ $\quad \delta(q_2, 0) = (q_2, 0, I)$

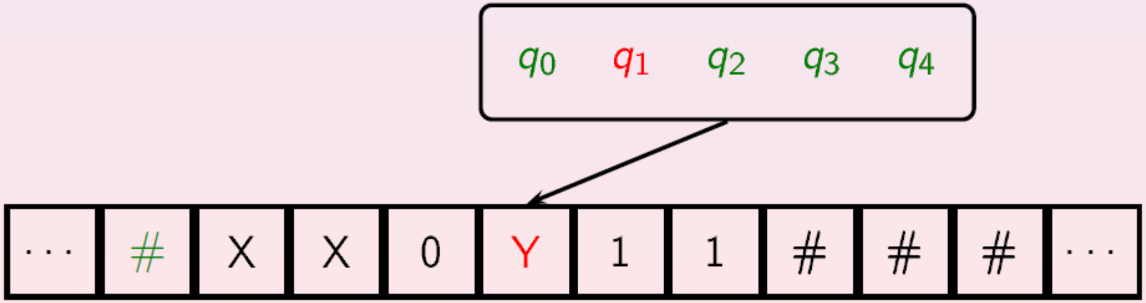
$\delta(q_2, X) = (q_0, X, D)$ $\quad \delta(q_2, Y) = (q_2, Y, I)$

$\delta(q_3, Y) = (q_3, Y, D)$ $\quad \delta(q_3, \#) = (q_4, \#, D)$



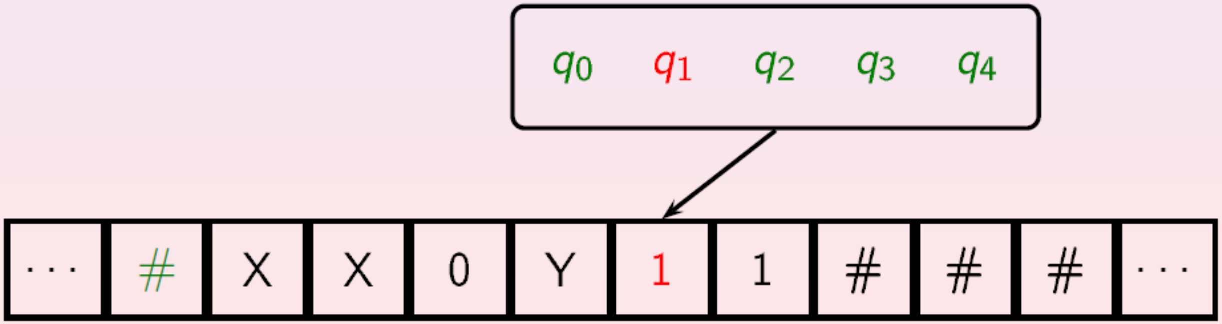
Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



Ejemplo de funcionamiento

$$\delta(q_0,0) = (q_1,X,D) \quad \delta(q_0,Y) = (q_3,Y,D) \quad \delta(q_1,0) = (q_1,0,D)$$
$$\delta(q_1,1) = (q_2,Y,I) \quad \delta(q_1,Y) = (q_1,Y,D) \quad \delta(q_2,0) = (q_2,0,I)$$
$$\delta(q_2,X) = (q_0,X,D) \quad \delta(q_2,Y) = (q_2,Y,I)$$
$$\delta(q_3,Y) = (q_3,Y,D) \quad \delta(q_3,\#) = (q_4,\#,D)$$



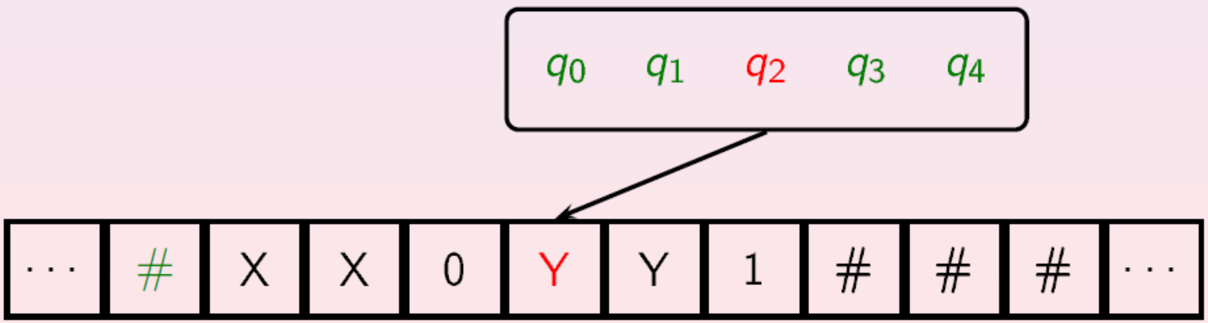
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\quad$ $\delta(q_0,Y)=(q_3,Y,D)$ $\quad$ $\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\quad$ $\delta(q_1,Y)=(q_1,Y,D)$ $\quad$ $\delta(q_2,0)=(q_2,0,I)$

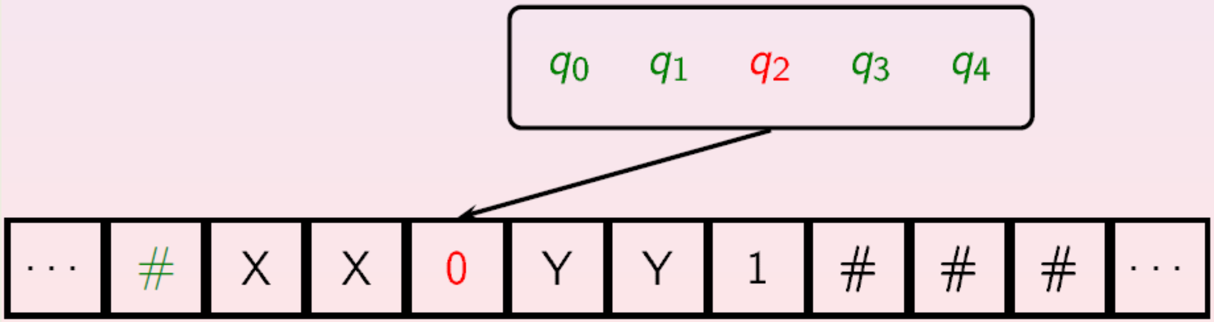
$\delta(q_2,X)=(q_0,X,D)$ $\quad$ $\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\quad$ $\delta(q_3,\#)=(q_4,\#,D)$



Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



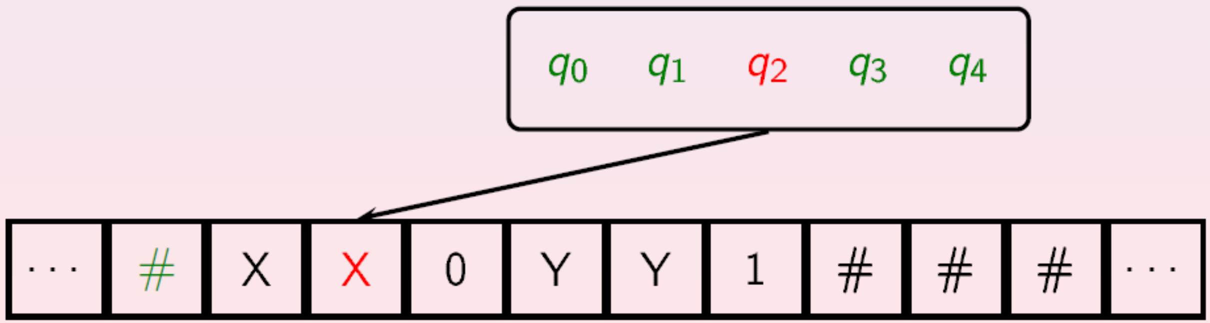
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\delta(q_0,Y)=(q_3,Y,D)$ $\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\delta(q_1,Y)=(q_1,Y,D)$ $\delta(q_2,0)=(q_2,0,I)$

$\delta(q_2,X)=(q_0,X,D)$ $\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\delta(q_3,\#)=(q_4,\#,D)$



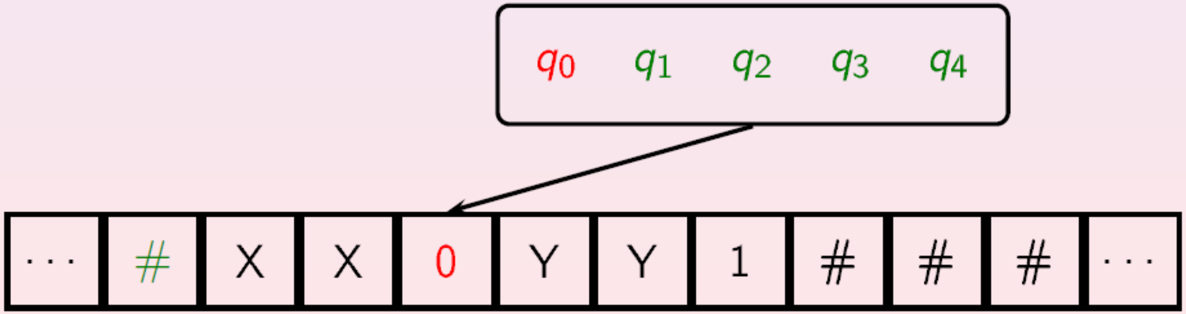
Ejemplo de funcionamiento

$\delta(q_0, 0) = (q_1, X, D)$ $\delta(q_0, Y) = (q_3, Y, D)$ $\delta(q_1, 0) = (q_1, 0, D)$

$\delta(q_1, 1) = (q_2, Y, I)$ $\delta(q_1, Y) = (q_1, Y, D)$ $\delta(q_2, 0) = (q_2, 0, I)$

$\delta(q_2, X) = (q_0, X, D)$ $\delta(q_2, Y) = (q_2, Y, I)$

$\delta(q_3, Y) = (q_3, Y, D)$ $\delta(q_3, \#) = (q_4, \#, D)$



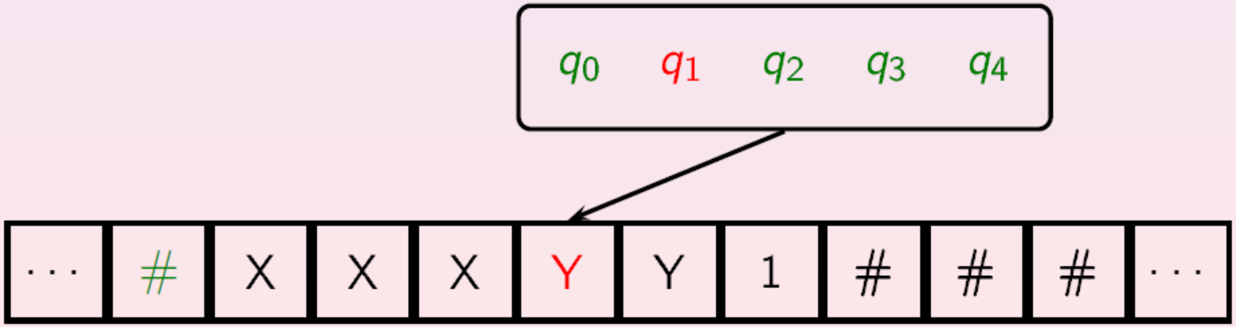
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\quad\delta(q_0,Y)=(q_3,Y,D)$ $\quad\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\quad\delta(q_1,Y)=(q_1,Y,D)$ $\quad\delta(q_2,0)=(q_2,0,I)$

$\delta(q_2,X)=(q_0,X,D)$ $\quad\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\quad\delta(q_3,\#)=(q_4,\#,D)$



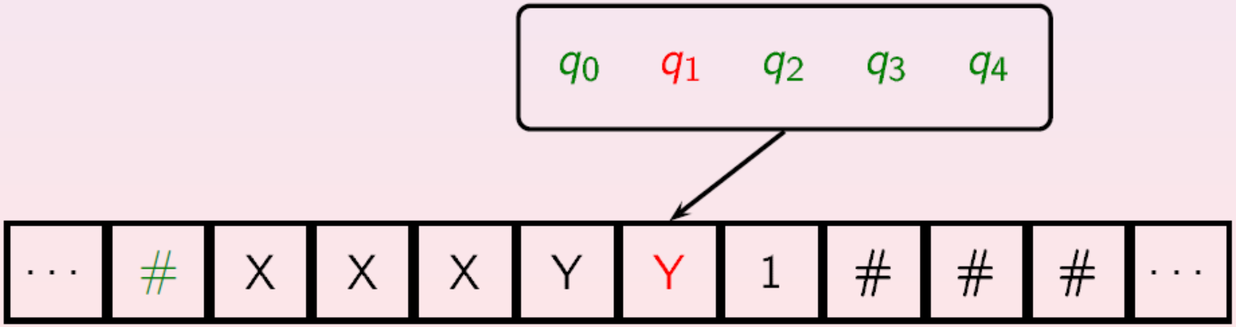
Ejemplo de funcionamiento

$\delta(q_0,0) = (q_1,X,D)$ $\delta(q_0,Y) = (q_3,Y,D)$ $\delta(q_1,0) = (q_1,0,D)$

$\delta(q_1,1) = (q_2,Y,I)$ $\delta(q_1,Y) = (q_1,Y,D)$ $\delta(q_2,0) = (q_2,0,I)$

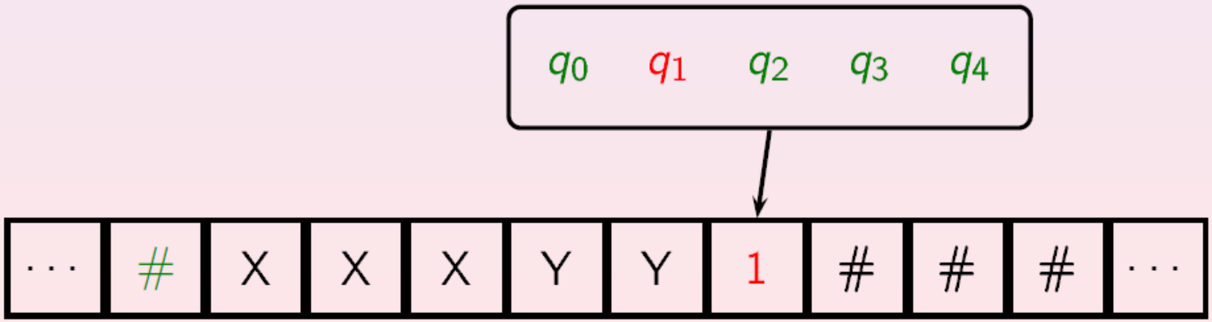
$\delta(q_2,X) = (q_0,X,D)$ $\delta(q_2,Y) = (q_2,Y,I)$

$\delta(q_3,Y) = (q_3,Y,D)$ $\delta(q_3,\#) = (q_4,\#,D)$



Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



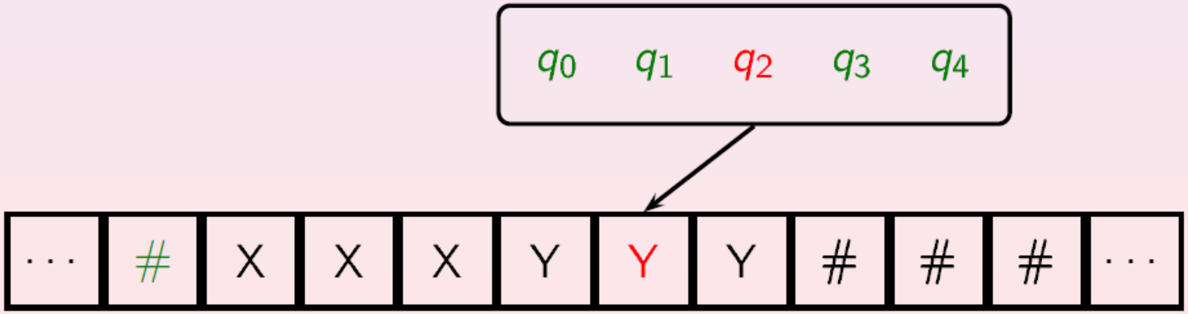
Ejemplo de funcionamiento

$\delta(q_0, 0) = (q_1, X, D)$ $\quad$ $\delta(q_0, Y) = (q_3, Y, D)$ $\quad$ $\delta(q_1, 0) = (q_1, 0, D)$

$\delta(q_1, 1) = (q_2, Y, I)$ $\quad$ $\delta(q_1, Y) = (q_1, Y, D)$ $\quad$ $\delta(q_2, 0) = (q_2, 0, I)$

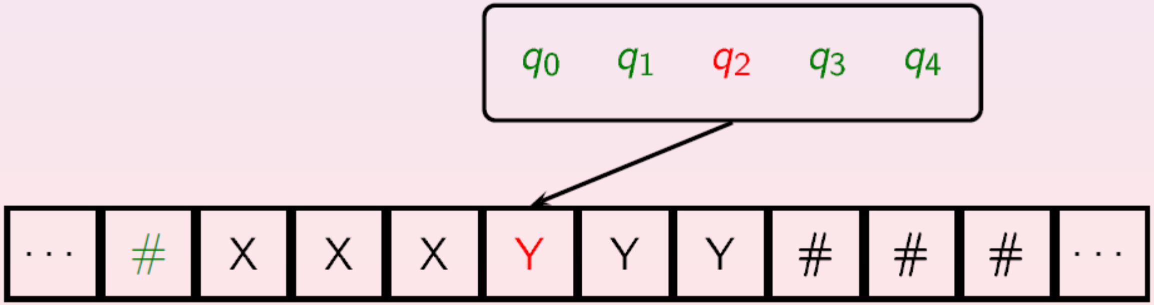
$\delta(q_2, X) = (q_0, X, D)$ $\quad$ $\delta(q_2, Y) = (q_2, Y, I)$

$\delta(q_3, Y) = (q_3, Y, D)$ $\quad$ $\delta(q_3, \#) = (q_4, \#, D)$



Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



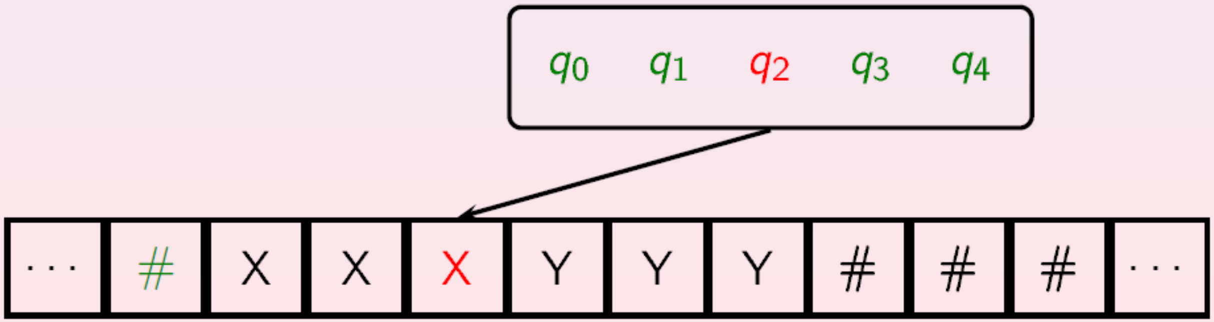
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\delta(q_0,Y)=(q_3,Y,D)$ $\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\delta(q_1,Y)=(q_1,Y,D)$ $\delta(q_2,0)=(q_2,0,I)$

$\delta(q_2,X)=(q_0,X,D)$ $\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\delta(q_3,\#)=(q_4,\#,D)$



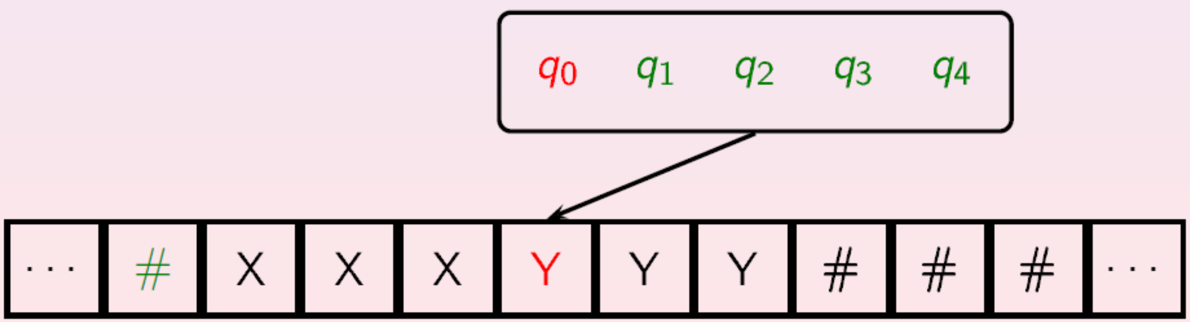
Ejemplo de funcionamiento

$\delta(q_0,0)=(q_1,X,D)$ $\delta(q_0,Y)=(q_3,Y,D)$ $\delta(q_1,0)=(q_1,0,D)$

$\delta(q_1,1)=(q_2,Y,I)$ $\delta(q_1,Y)=(q_1,Y,D)$ $\delta(q_2,0)=(q_2,0,I)$

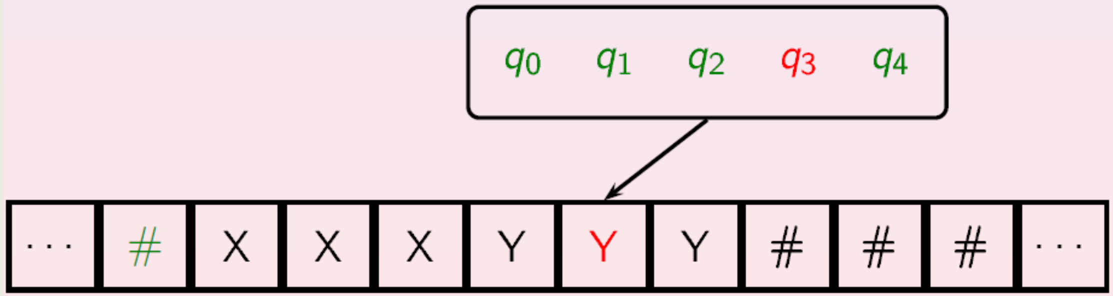
$\delta(q_2,X)=(q_0,X,D)$ $\delta(q_2,Y)=(q_2,Y,I)$

$\delta(q_3,Y)=(q_3,Y,D)$ $\delta(q_3,\#)=(q_4,\#,D)$



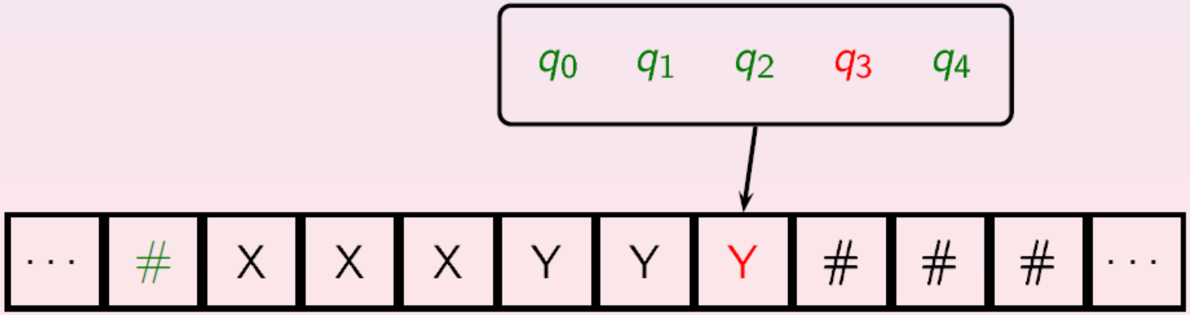
Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



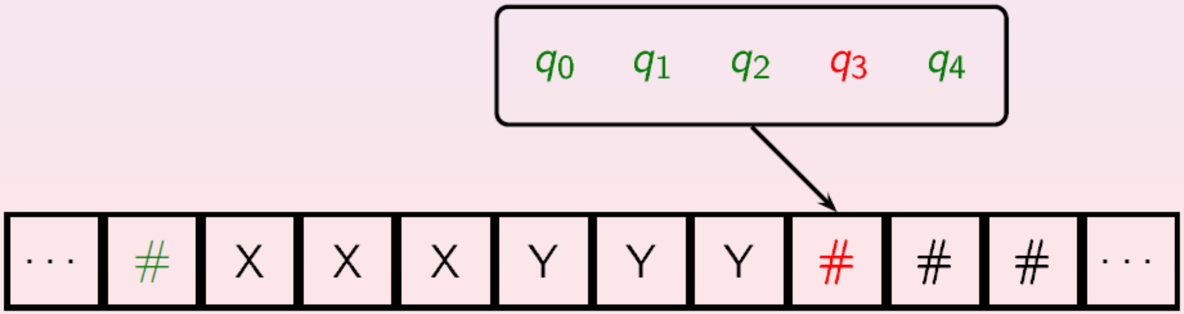
Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



Ejemplo de funcionamiento

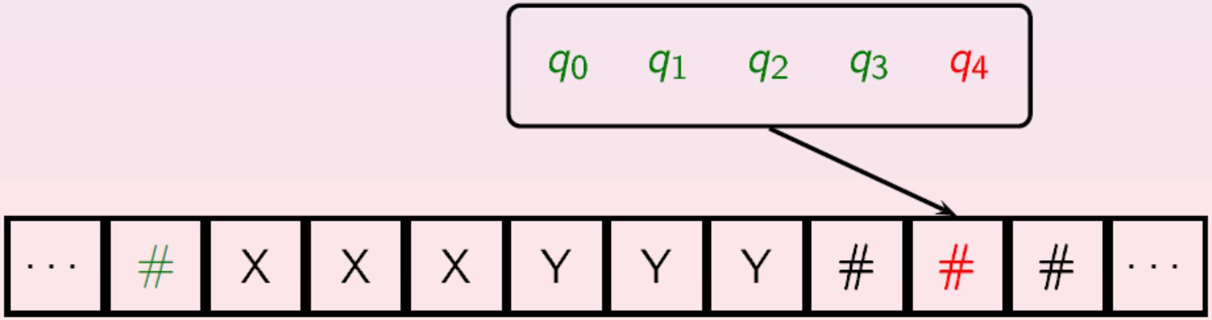
$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$



Ejemplo de funcionamiento

$$\delta(q_0,0)=(q_1,X,D) \quad \delta(q_0,Y)=(q_3,Y,D) \quad \delta(q_1,0)=(q_1,0,D)$$
$$\delta(q_1,1)=(q_2,Y,I) \quad \delta(q_1,Y)=(q_1,Y,D) \quad \delta(q_2,0)=(q_2,0,I)$$
$$\delta(q_2,X)=(q_0,X,D) \quad \delta(q_2,Y)=(q_2,Y,I)$$
$$\delta(q_3,Y)=(q_3,Y,D) \quad \delta(q_3,\#)=(q_4,\#,D)$$

La palabra 000111 es aceptada



Configuración de MT

Una **configuración** de una MT es una tripleta (q, w1, w2) donde:

- q es el estado en el que se encuentra la máquina.
- w1 es la representación de la parte de la palabra que hay a la izquierda de la posición del cabezal de lectura (puede ser vacío). Esta representación se obtiene eliminando la sucesión infinita de blancos a la izquierda de las casillas que son distinto de blanco.
- w2 es la representación de la parte de la palabra que se obtiene empezando en el cabezal de lectura hacia la derecha. No puede ser vacío. Esta representación se obtiene eliminando la sucesión infinita de blancos a la derecha de las casillas que son distinto de blanco.

Configuración inicial

Si u está en A*, la **configuración inicial** de la Máquina de Turing $(Q,A,C,\delta,q_0,\#,F)$ asociada a esta palabra es (q_0,ε,u).

Paso de cálculo: Movimiento a la izquierda

Si $\delta(q,a) = (p,b,I)$ entonces decimos que de la configuración $(q,c_1\ldots c_n,ad_2\ldots d_m)$ llegamos en un paso de cálculo a la configuración $(p,c_1\ldots c_{n-1}, c_nbd_3\ldots d_m)$, lo que se denota como

$$(q, c_1 \ldots c_n, ad_2 \ldots d_m) \vdash (p, c_1 \ldots c_{n-1}, c_nbd_3 \ldots d_m)$$

con dos excepciones:

- Si $n=0$ (partimos de $(q,\varepsilon, ad_2\ldots d_m)$), se llega a $(p, \varepsilon, \#bd_2\ldots d_m)$
- Si $m=1$ y $b=\#$, entonces se llega a $(p,c_1\ldots c_{n-1}, \#)$

Las dos excepciones se pueden dar de forma simultánea (si $b=\#$) y entonces de (q,ε,a) se llega a $(p, \varepsilon, \#)$

Paso de cálculo: Movimiento a la derecha

Si δ(q,a) = (p,b,D) entonces decimos que de la configuración $(q,c_1\ldots c_n,ad_2\ldots d_m)$ llegamos en un paso de cálculo a la configuración $(p,c_1\ldots c_nb, d_2\ldots d_m)$, lo que se denota como

$$(q, c_1 \ldots c_n, ad_2 \ldots d_m) \vdash (p, c_1 \ldots c_n b, d_2 \ldots d_m)$$

con dos excepciones:

- Si n=0 y b=#, desde $(q\ \varepsilon, ad_2\ldots d_m)$ se llega a $(p, \varepsilon, d_2\ldots d_m)$
- Si m=1 (partimos de $(q,c_1\ldots c_n, a)$), entonces se llega a $(p,c_1\ldots c_nb, \#)$

Las dos excepciones se pueden dar de forma simultánea (si b=#) y entonces de (q,ε,a) se llega a (p, ε, #)

Relación de paso de cálculo

Si R y R' son dos configuraciones de una MT, se dice que desde R se llega a R' si existe una sucesión de pasos de cálculo consecutivos comenzando en R y terminando en R'.

Lenguaje aceptado por una MT

Si **M** es una máquina de Turing, entonces el lenguaje L(M) aceptado es el conjunto de palabras u que verifican que, desde el estado inicial, existe una sucesión de pasos de cálculo que llegue a un estado final con u como entrada.

1. Introducción
2. Máquinas de Turing
3. Lenguajes recursivamente enumerables
4. Construcción de máquinas de Turing
5. Extensiones del modelo de MT

Lenguaje recursivamente enumerable

Sea L un lenguaje. Se dice que **L es recursivamente enumerable** si existe una MT **M** tal que **L(M)=L**.

Parada en las máquinas de Turing

Una máquina **para** cuando en el estado actual y símbolo de la cinta no hay ninguna transición definida.

Cuando se llega a un estado final **q** podemos suponer que la Máquina de Turing para, es decir no hay ninguna transición definida.

Existe otro criterio de aceptación: una palabra es aceptada cuando la MT para. **La clase de lenguajes aceptada por este criterio es también la clase de los lenguajes recursivamente enumerables**.

Lenguaje recursivo

Un lenguaje se dice **recursivo** si es aceptado por una MT que siempre para.

Lenguajes recursivos y recursivamente enumerables

Un lenguaje recursivo es siempre recursivamente enumerable. Los lenguajes recursivos son aquellos cuyo problema de aceptación puede ser resuelto mediante un algoritmo.

En el caso de lenguajes recursivos, podemos suponer que hay dos tipos de estados finales: de aceptación y de rechazo. La máquina acepta cuando se llega a un estado de aceptación y rechaza cuando llega a un estado de rechazo.

Lenguaje recursivo

Un lenguaje se dice **recursivo** si es aceptado por una MT que siempre para.

Lenguajes recursivos y recursivamente enumerables

Un lenguaje recursivo es siempre recursivamente enumerable. Los lenguajes recursivos son aquellos cuyo problema de aceptación puede ser resuelto mediante un algoritmo.

En el caso de lenguajes recursivos, podemos suponer que hay dos tipos de estados finales: de aceptación y de rechazo. La máquina acepta cuando se llega a un estado de aceptación y rechaza cuando llega a un estado de rechazo.

1. Introducción
2. Máquinas de Turing
3. Lenguajes recursivamente enumerables
4. Construcción de máquinas de Turing
5. Extensiones del modelo de MT

Funciones parcialmente calculables

La función **f** se dice parcialmente calculable si existe una MT $M=(Q,A,C,\delta ,q_0 , \#,F)$ tal que para cualquier palabra **u** del dominio de **f** entonces desde la configuración (q_0,ε,u) (configuración inicial) se llega a una configuración (q,u_1,u_2) en la que $u_1u_2=f(u)$, **q** es final y la máquina para (llegamos a una configuración en la que estamos en un estado final, la máquina termina y el contenido de la cinta es **f(u)**.

Si **u** no es del dominio de **f, entonces la máquina no para.**

Funciones calculables totales

Si una función es calculable parcial y su dominio es A* (está siempre definida) se dice que es calculable total.

Ejemplo: Restador unario

Para dos números naturales **n,m**, calcular f(n,m)=n-m si n>m, y f(n,m)= 0 si n<m.

La entrada debe de ser 0^n10^m y la salida debe de ser una configuración en la que en la cinta esté $0^{f(n,m)}$ rodeado de blancos. No nos preocupa cuál es la salida si la entrada no es correcta (no corresponde a dos series de ceros separadas por un 1).

La MT será $M=(\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0,1\}, \{0,1,\#\}, \delta, q_0, \#, q_6)$

Ejemplo: Restador unario

δ viene dada por la siguiente tabla:

Estado	0	1	#
q_0	$(q_1, \#, D)$	$(q_5, \#, D)$	$-$
q_1	$(q_1, 0, D)$	$(q_2, 1, D)$	$-$
q_2	$(q_3, 1, I)$	$(q_2, 1, D)$	$(q_4, \#, I)$
q_3	$(q_3, 0, I)$	$(q_3, 1, I)$	$(q_0, \#, D)$
q_4	$(q_4, 0, I)$	$(q_4, \#, I)$	$(q_6, 0, D)$
q_5	$(q_5, \#, D)$	$(q_5, \#, D)$	$(q_6, \#, D)$
q_6	$-$	$-$	$-$

Ejercicios:

Diseñar máquinas de Turing para los siguientes lenguajes:

- Palabras sobre el alfabeto {0,1} con el mismo número de ceros que de unos.
- $L=\{a^nb^nc^n | n>= 1\}$
- $L= \{ ww^{-1} | w \text{ en } \{0,1\}^*\}$
- $L=\{ wcw | w \text{ en } \{0,1\}^* \}$

1. Introducción
2. Máquinas de Turing
3. Lenguajes recursivamente enumerables
4. Construcción de máquinas de Turing
5. Extensiones del modelo de MT

Recordando símbolos

Una MT puede diseñarse para que recuerde un símbolo del alfabeto de trabajo (o del alfabeto de entrada).

Por ejemplo, si queremos que se recuerde un símbolo de B cuando está en el estado q, entonces basta con cambiar el estado q por las parejas de estados [q,b] donde b está en B.

A menudo queremos que se recuerde un símbolo en cualquier estado y entonces el conjunto de estados sería el conjunto de las parejas Q’xB formadas por un elemento q de Q' y un símbolo b de B.

Escribir los estados de esta forma ayuda a comprender el significado de los mismos.

Ejemplo

Diseñar una máquina de Turing para el lenguaje 01*+10* (recordar el primer símbolo y comprobar que nunca más aparezca).

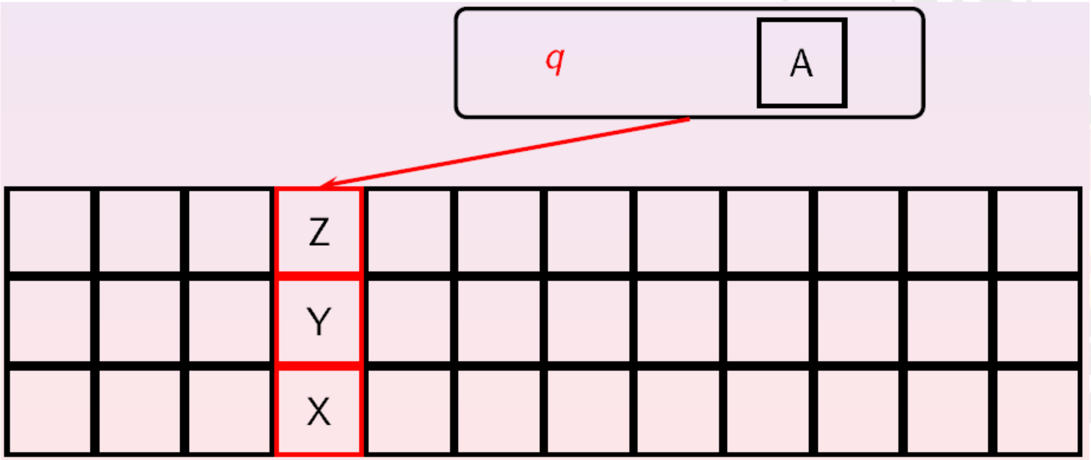
- La Máquina es M=(Q, {0,1},{0,1,#}, δ, [q0,#], {[q1,#]}) donde:
  - Q={q0,q1}x{0,1,#}
  - Las posibles transiciones serían:

$\delta([q_0, B], a) = ([q_1, a], a, D)$ para $a = 0$ o $a = 1$.
$\delta([q_1, a], \overline{a}) = ([q_1, a], \overline{a}, D)$, donde $\overline{a}$ es el complementario de a (esto es, $\overline{a} = 1$ si $a = 0$ y $\overline{a} = 0$ si $a = 1$).
$\delta([q_1, a], \#) = ([q_1, \#], \#, D)$

Pistas múltiples

A menudo es útil pensar que la MT tiene una cinta con varias pistas: en lugar de tener una sola casilla, disponemos de varias casillas en cada posición donde poder escribir un símbolo.

Tener dos pistas equivale a suponer que el alfabeto de trabajo está formado por los elementos de BxB y tener k cintas a suponer que el alfabeto de trabajo es B^k.



Ejemplo

Una forma de utilizar las pistas múltiples es imaginar que una pista se usa para los datos y otra para poner una marca.

Vamos a diseñar una MT que acepte el lenguaje $L = \{ wcw| w en \{0,1\}^{+}\}$.

La MT tiene los siguientes elementos $M=(Q,A,B,\delta,[q_1,\#],[\#,\#],\{[q_9,\#]\backslash\})$ donde

- $Q= \{q_1,q_2,\ldots,q_9\}x [0,1]$ (podemos recordar 0,1)
- $B= \{0,1,c, \#\}x \{\#,*\}$
- $A= \{0,1,c\}$. 0 se identifica con [0, #] y 1 se identifica con [1,#]

La función de transición δ se especifica en la siguiente pantalla.

Ejemplo

a y b pueden ser $0, 1$.

$\delta([q_1, \#], [a, \#]) = ([q_2, a], [a, *], D)$ $\quad$ $\delta([q_2, a], [b, \#]) = ([q_2, a], [b, \#], D)$
$\delta([q_2, a], [c, \#]) = ([q_3, a], [c, \#], D)$ $\quad$ $\delta([q_3, a], [b, *]) = ([q_3, a], [b, *], D)$
$\delta([q_3, a], [a, \#]) = ([q_4, \#], [a, *], I)$ $\quad$ $\delta([q_4, \#], [a, *]) = ([q_4, \#], [a, *], I)$
$\delta([q_4, \#], [c, \#]) = ([q_5, \#], [c, \#], I)$ $\quad$ $\delta([q_5, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$
$\delta([q_6, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$ $\quad$ $\delta([q_6, \#], [a, *]) = ([q_1, \#], [a, *], D)$
$\delta([q_5, \#], [a, *]) = ([q_7, \#], [a, *], D)$ $\quad$ $\delta([q_7, \#], [c, \#]) = ([q_8, \#], [c, \#], D)$
$\delta([q_8, \#], [a, *]) = ([q_8, \#], [a, *], D)$ $\quad$ $\delta([q_8, \#], [\#, \#]) = ([q_9, \#], [\#, \#], D)$

Subrutinas (I)

Una subrutina en una MT es un conjunto de estados que realiza una acción concreta.

- En este conjunto de estados habrá un estado inicial y otro estado que sirve como estado de retorno.
- No se añade ninguna funcionalidad nueva, sólo es una forma de organizar los estados de una MT agrupando aquellos que realizan una tarea concreta y suponiendo que siempre podemos movernos a ese conjunto de estados.
- La MT no tiene un sistema de llamadas que permita saber a qué posición y en qué estado hay que volver.
- La posición se puede recordar con una pista adicional y un símbolo extra que indique la casilla en la que se tiene que posicionar.

Subrutinas (II)

- El estado se puede determinar haciendo varias copias del último estado de la subrutina, una para cada estado al que haya que volver. El número de copias es finito.
- Siempre que hagamos un conjunto de estados para una tarea determinada, por ejemplo, desplazar el contenido de todas las casillas a partir de la posición actual un lugar a la derecha, supondremos que esta tarea siempre la podemos hacer en una MT sin necesidad de especificar los estados.

Ejemplo: Multiplicación de números en base 1

La MT comenzará con una cadena de la forma $0^m10^n\$$ en la cinta, y terminará con 0^{mn} al final. No nos preocupamos se la MT tienen una entrada mal escrita. Finalizará con algo en la cinta que no tendrá sentido, en general.

En etapas sucesivas, la cinta va a contener cadenas de la forma $\$0^i10^n10^{kn}$ donde i+k=m para valores de k=1…m.

En un paso básico se cambia el primer cero del primer grupo por un blanco y se añaden n ceros al último grupo: se pasa de $\$0^i10^n10^{kn}$ a $0^{i-1}10^n10^{(k+1)n}$.

Finalmente la subcadena 10^n1 se sustituye por blancos.

Ejemplo: Multiplicación de números en base 1

Vamos a diseñar un conjunto de estados que copia n ceros al final de la cinta cuando está situada justo al principio de la serie de n ceros. Termina en la misma posición de la cinta en el estado q_5.

Tiene los siguientes estados y estructura:

$$\delta(q_1,0) = (q_2, X, D)$$
$$\delta(q_2,0) = (q_2,0,D) \qquad \delta(q_2,1) = (q_2,1,D)$$
$$\delta(q_2,\#) = (q_3,0,I)$$
$$\delta(q_3,0) = (q_3,0,I) \qquad \delta(q_3,1) = (q_3,1,I)$$
$$\delta(q_3,X) = (q_1,X,D) \qquad \delta(q_1,1) = (q_4,1,I)$$
$$\delta(q_4,X) = (q_4,0,I) \qquad \delta(q_4,1) = (q_5,1,D)$$

Ejemplo: Multiplicación de números en base 1

Estructura del programa:

Pasamos de la configuración en la que hay 0^m10^n en la cinta y estamos colocados al principio de esta palabra a la configuración en la que en la cinta hay 0^m10^n1 y estamos colocados al principio de la palabra.

$$\delta(q_0,0)=(q_0,0,D) \quad \delta(q_0,1)=(q_0,1,D)$$
$$\delta(q_0,\#)=(q_{13},1,I) \quad \delta(q_{13},0)=(q_{13},0,I)$$
$$\delta(q_{13},1)=(q_{13},1,I) \quad \delta(q_{13},\#)=(q_{14},\#,D)$$

Miramos si hay un 0 al principio, lo sustituimos por un blanco, #, y nos ponemos en situación para copiar n ceros al final.

$$\delta(q_{14},0)=(q_6,\#,D)$$
$$\delta(q_6,0)=(q_6,0,D) \quad \delta(q_6,1)=(q_1,1,D)$$

Ejemplo: Multiplicación de números en base 1

Después de hacer la copia, estamos en q_5 y desde este estado, volvemos a la posición inicial si hay más 0 en la primera serie, o finalizamos si ya no hay más ceros (q_{12} es el estado final de parada):

$$
\begin{array}{ll}
\delta(q_5,0)=(q_7,0,I) & \delta(q_7,1)=(q_8,1,I) \\
\delta(q_8,0)=(q_9,0,I) & \delta(q_8,\#)=(q_{10},\#,D) \\
\delta(q_9,0)=(q_9,0,I) & \delta(q_9,\#)=(q_{14},\#,D) \\
\delta(q_{10},1)=(q_{11},\#,D) & \\
\delta(q_{11},0)=(q_{11},\#,D) & \delta(q_{11},1)=(q_{12},\#,D)
\end{array}
$$

Otras extensiones

- MT con múltiples cintas
- MT con múltiples cintas y pistas
- MT no deterministas
- … etc

Ninguna extensión a la MT sirve para aumentar su potencia de cálculo. Únicamente para *simplificar* la construcción de una máquina que acepte un lenguaje u otro.

UNIVERSIDAD DE GRANADA

DECSAI

Modelos de Computación

Grado en Ingeniería Informática

Tema 7 – Máquinas de Turing

Manuel Pegalajar Cuéllar
manupc@ugr.es

Departamento de Ciencias de la Computación e Inteligencia Artificial
http://decsai.ugr.es