

Tema 1: Máquinas de Turing. Funciones y Lenguajes Calculables (Parte 1)

Serafín Moral

Universidad de Granada

Febrero, 2025

Contenido

- Máquinas de Turing
- Lenguajes recursivamente enumerables. Lenguajes recursivos
- Técnicas de construcción de Máquinas de Turing: memoria adicional, pistas múltiples, subrutinas
- Extensiones del concepto de MT
 - Pasos estáticos
 - Máquinas multicinta
 - Máquinas no deterministas
- Limitaciones de las MT
 - Máquinas con cintas semiilimitadas
- Problemas y Programas
 - Tipos de problemas
 - Programas
 - Problemas de decisión y Lenguajes
 - Palabras y Números
- Calculabilidad
 - El lenguaje diagonal
 - El lenguaje universal
 - Problemas indecidibles
 - Teorema de Rice
 - El problema de las correspondencias de Post

Enunciado

- Entrada: un programa y unos datos de entrada
- Salida: SI (cuando el programa termina para esos datos) y NO (cuando el programa cicla de forma indefinida para esos datos)

¿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++;
    }
}
```

- El programa termina para una entrada n si y solo si existen números enteros positivos x, y, z tales que

$$x^n + y^n = z^n$$

- Ahora se sabe que no termina para $n > 2$ según el *último teorema de Fermat*, pero hace algunos años esto no se sabía.
- El teorema fue enunciado en 1637, pero no fue demostrado hasta 1995 por el matemático Andrew Wiles
- Se han necesitado más de 300 años para saber si un programa concreto termina, pero nosotros queremos un algoritmo que nos diga si cualquier programa termina!!
- Este es un problema indecidible. La teoría para demostrar esto la haremos usando Máquinas de Turing. La podríamos hacer usando programas en C, pero las demostraciones matemáticas serían más complejas.

Máquinas de Turing

Una **Máquina de Turing** (MT) es una séptupla $(Q, A, B, \delta, q_0, \#, F)$ en la que

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

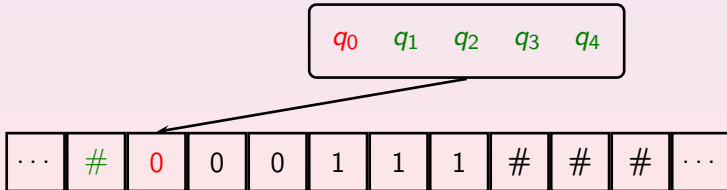
Consideremos la MT

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

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

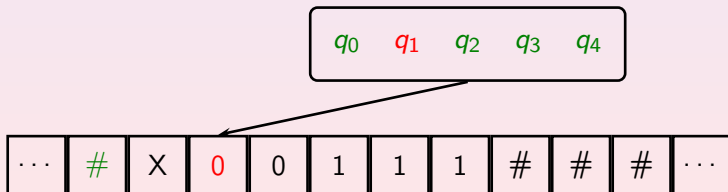
Funcionamiento

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



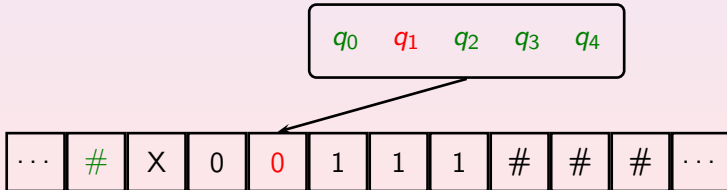
Funcionamiento

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



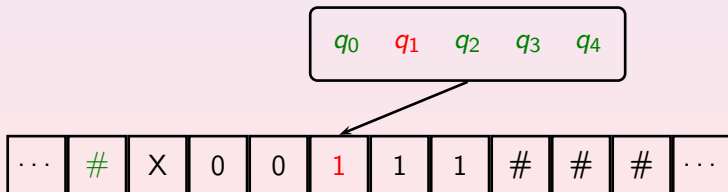
Funcionamiento

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



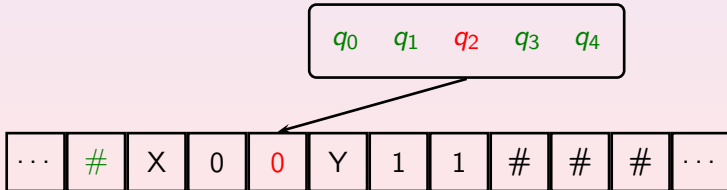
Funcionamiento

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



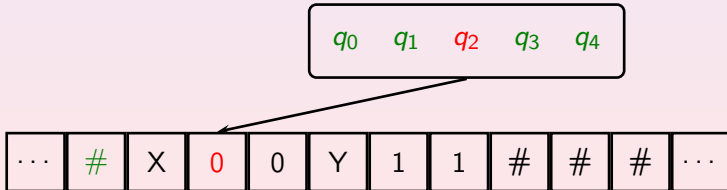
Funcionamiento

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



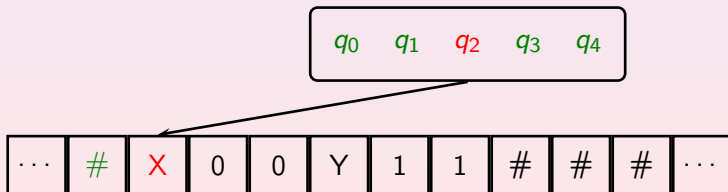
Funcionamiento

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



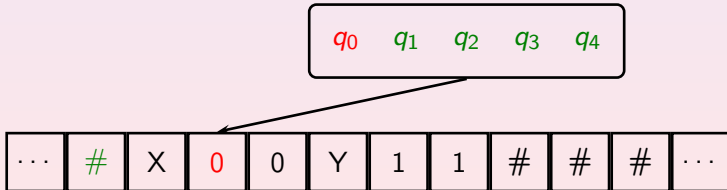
Funcionamiento

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



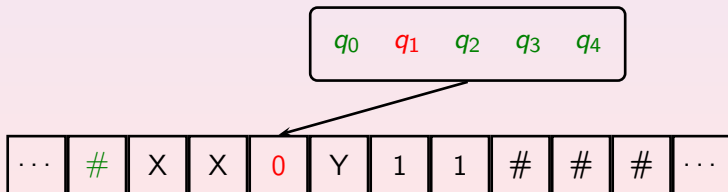
Funcionamiento

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



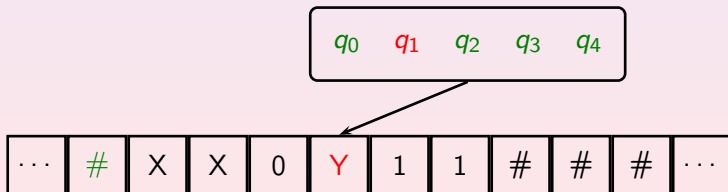
Funcionamiento

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



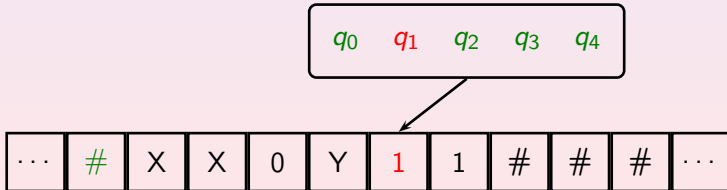
Funcionamiento

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



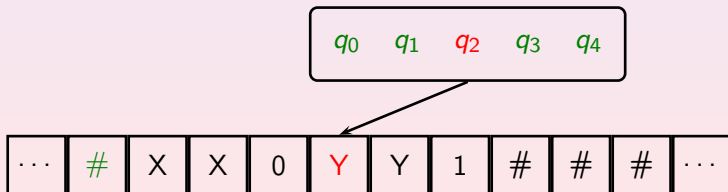
Funcionamiento

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



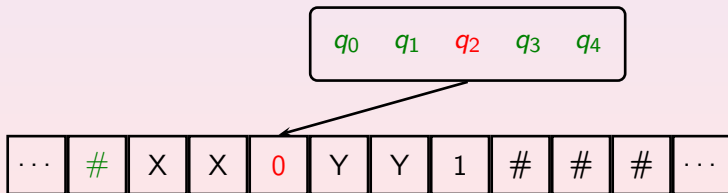
Funcionamiento

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



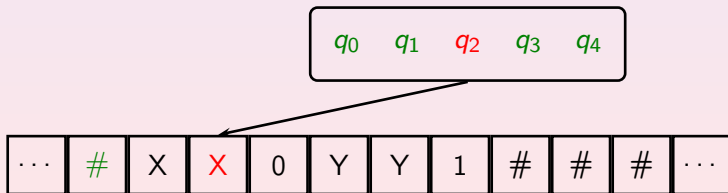
Funcionamiento

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



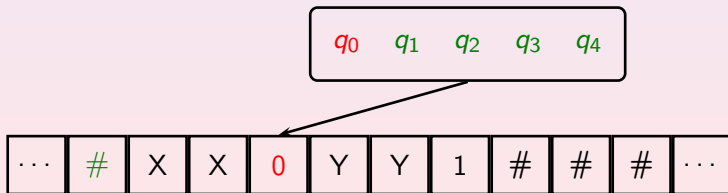
Funcionamiento

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



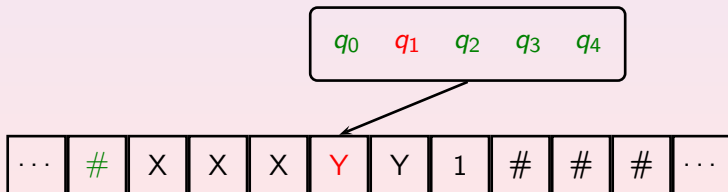
Funcionamiento

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



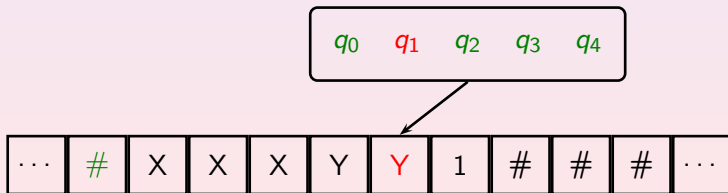
Funcionamiento

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



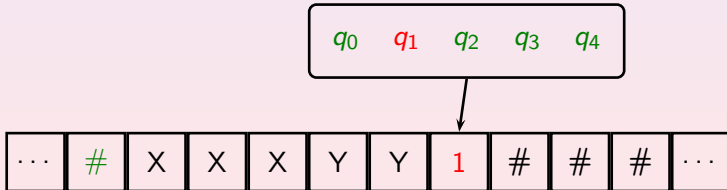
Funcionamiento

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



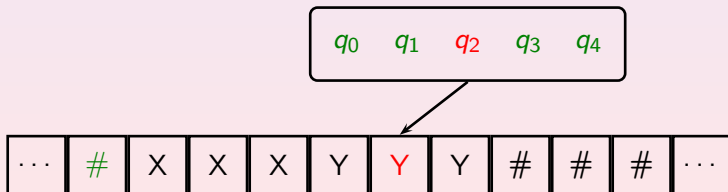
Funcionamiento

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



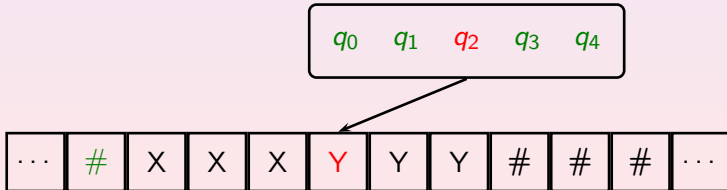
Funcionamiento

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



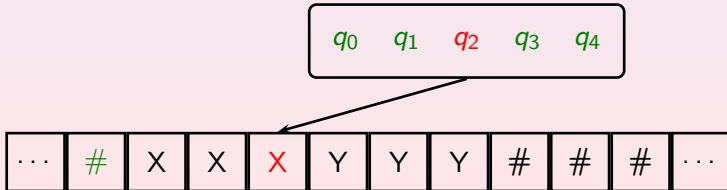
Funcionamiento

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



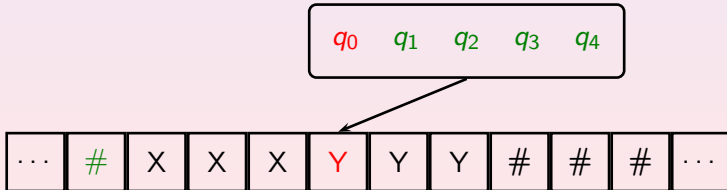
Funcionamiento

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



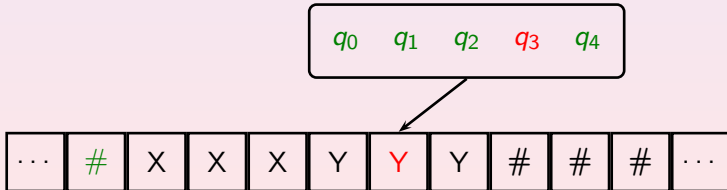
Funcionamiento

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



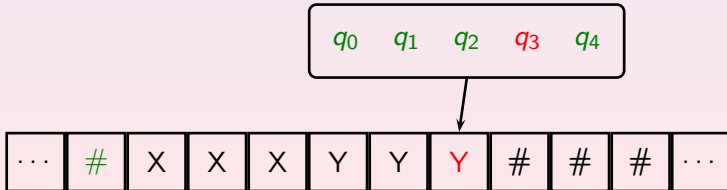
Funcionamiento

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



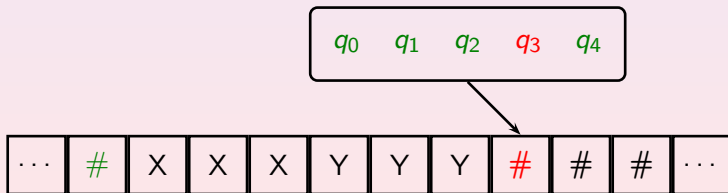
Funcionamiento

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



Funcionamiento

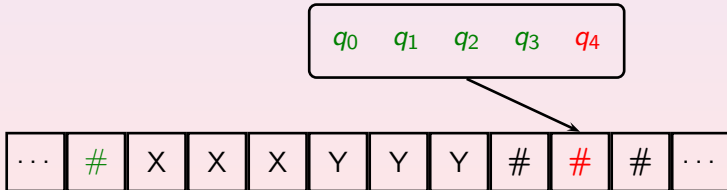
$$\begin{array}{lll} \delta(q_0, 0) = (q_1, X, D) & \delta(q_0, Y) = (q_3, Y, D) & \delta(q_1, 0) = (q_1, 0, D) \\ \delta(q_1, 1) = (q_2, Y, I) & \delta(q_1, Y) = (q_1, Y, D) & \delta(q_2, 0) = (q_2, 0, I) \\ \delta(q_2, X) = (q_0, X, D) & \delta(q_2, Y) = (q_2, Y, I) & \\ \delta(q_3, Y) = (q_3, Y, D)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



Funcionamiento

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

La palabra 000111 es aceptada



Configuración

Una **configuración** de una Máquina de Turing es una tripleta (q, w_1, w_2) donde

- q es el estado en el que se encuentra la máquina
- w_1 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.
- w_2 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 $u \in A^*$, la **configuración inicial** de la Máquina de Turing $(Q, A, C, \delta, q_0, \#, F)$ asociada a esta palabra es (q_0, ε, u) , siendo $(q_0, \varepsilon, \#)$ si $u = \varepsilon$.

Paso de Cálculo (movimiento a la izquierda)

Si $\delta(q, a) = (p, b, l)$ entonces decimos que de la configuración $(q, c_1 \dots c_n, ad_2 \dots d_m)$ llegamos en un paso de cálculo a la configuración $(p, c_1 \dots c_{n-1}, c_n b d_2 \dots d_m)$ lo que se denota como $(q, c_1 \dots c_n, ad_2 \dots d_m) \vdash (p, c_1 \dots c_{n-1}, c_n b d_3 \dots d_m)$ donde se supone:

- Si $c_1 \dots c_n = \varepsilon$, entonces $c_1 \dots c_{n-1} = \varepsilon$ y $c_n = \#$.
- Se eliminan los blancos a la derecha de la palabra $c_n b d_3 \dots d_m$ excepto el primero si toda la palabra está formada por blancos.

Paso de Cálculo (movimiento a la derecha)

Si $\delta(q, a) = (p, b, D)$ entonces decimos que de la configuración $(q, c_1 \dots c_n, a d_2 \dots d_m)$ llegamos en un paso de cálculo a la configuración $(p, c_1 \dots c_n b, d_2 d_3 \dots d_m)$ lo que se denota como $(q, c_1 \dots c_n, a d_2 \dots d_m) \vdash (p, c_1 \dots c_n b, d_2 d_3 \dots d_m)$ donde se considera:

- Si $m = 1$ entonces $d_2 d_3 \dots d_m = \#$
- Se eliminan todos los blancos a la izquierda en $c_1 \dots c_n b$.

Relación de pasos de cálculo

Si R y R' son configuraciones de una máquina de Turing $M = (Q, A, C, \delta, q_0, \#, F)$, se dice que desde R se llega en una sucesión de pasos de cálculo a R' lo que se denota como $R \vdash^* R'$ si y solo si existe una sucesión finita de configuraciones R_1, \dots, R_n tal que $R = R_1$, $R' = R_n$ y $R_i \vdash R_{i+1}, \forall i < n$.

Lenguaje aceptado por una Máquina de Turing

Si M es una máquina de Turing, entonces el lenguaje aceptado es el conjunto de palabras $L(M)$ tales que $u \in L(M)$ si y solo si existen $w_1, w_2 \in B^*$ y $q \in F$ tales que $(q_0, \varepsilon, u) \vdash^* (q, w_1, w_2)$ (es decir desde la configuración inicial asociada a u se puede llegar mediante una sucesión de pasos de cálculo a una configuración en la que estamos en un estado final).

Lenguaje Recursivamente Enumerable

Definición: Recursivamente Enumerable

Un lenguaje $L \subseteq A^*$ se dice **recursivamente enumerable (e.r.)** si y solo si existe una máquina de Turing $M = (Q, A, C, \delta, q_0, \#, F)$ 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 $q \in F$ 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.

Aceptación y Parada de MTs

- Si una MT llega a un estado de aceptación, ya acepta la palabra de entrada, con independencia de lo que haga después. Luego ese cálculo es irrelevante y la MT acepta el mismo lenguaje si no sigue calculando. Por ese motivo, nosotros supondremos que los estados de aceptación no tienen transiciones salientes y las MTs siempre terminan cuando llegan a un estado de aceptación.
- Una MT puede no aceptar una palabra de dos formas distintas:
 - Llegando a un estado no final en el que para el contenido de la cinta no hay transición definida. Termina y rechaza. En algunos casos se le exige que la MT llegue a un estado de rechazo, pero para nosotros no será necesario.
 - Ciclando de forma indefinida sin llegar a un estado de aceptación. En este caso nunca llega a aceptar la palabra, aunque técnicamente tampoco la rechaza. La palabra no es aceptada ni rechazada.

Definición

Un lenguaje se dice **recursivo** si es aceptado por una MT que siempre termina: todas las palabras son aceptadas o rechazadas.

Un lenguaje recursivo es siempre recursivamente enumerable. Los lenguajes recursivos son aquellos cuyo problema de aceptación poder 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.

Máquinas de Turing Calculadoras

Definición

Dada una MT $M = (Q, A, B, \delta, q_0, \#, F)$, la función f calculada por esta MT es una función

$$f : D \rightarrow B^*$$

tal que $D \subseteq A^*$ es el conjunto de entradas para los que la MT termina y si $u \in D$, entonces $f(u)$ es el contenido de la cinta cuando la MT termina excluyendo los símbolos en blanco.

Funciones Parcialmente Calculables

Una función f se dice que es parcialmente calculable cuando existe una MT que la calcula.

Funciones Calculables Totales

Si una función es parcialmente calculable y $D = A^*$ (la MT termina en todas las entradas) se dice que es calculable total.

Ejemplo: restar números en unario

- Vamos a diseñar una MT que resta números en unario.
- Para dos números naturales $n, m \in \mathbb{N}$ calcula $f(n, m)$ que es igual a $n - m$ si $n \geq m$ y 0 si $n < m$.
- La entrada debe de ser $0^n 10^m$ y la salida debe de ser una confiración en la que en la cinta esté $0^{f(n, m)}$ rodeado de blancos. No nos preocupa cual es la salida si la entrada no es correcta (no corresponde a dos series de ceros separadas por un 1).
- La MT será
$$M = (\{\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \{0, 1, \#\}, \delta, q_0, \#, q_6\})$$

Restar números en unario: función de transición

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	—	—	—

El primer 0 se convierte en blanco y se mueve a la derecha en q_1 hasta que encuentre un 1 y cambia a q_2 .

Restar números en unario: función de transición

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	—	—	—

En q_2 se mueve a la derecha saltando 1's hasta que encuentra un 0 y entonces empieza a moverse a la izquierda hasta encontrar un blanco, entonces se mueve a la derecha y empieza de nuevo en q_0 .

Restar números en unario: función de transición

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	—	—	—

El proceso se repite: si al volver a q_0 lo que encuentra es un 1, entonces es que $m \geq n$ y el resultado es la cadena vacía que representa a 0, para ello pasa a q_5 donde borra todo.

Restar números en unario: función de transición

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	—	—	—

Si en q_2 no encuentra 0, es que $n > m$ y hemos borrado un 0 de más, se pasa a q_4 en el nos movemos a la izquierda borrando los 1's y añadiendo un 0 (cuando se llega a #) y se pasa al estado final.

Diseñar máquinas de Turing para los siguientes lenguajes:

- Palabras sobre el alfabeto $\{0,1\}$ con el mismo número de ceros que de unos.
- $L = \{a^n b^n c^n \mid n \geq 1\}$
- $\{ww^{-1} \mid w \in \{0,1\}^*\}$
- $\{wcw \mid w \in \{0,1\}^*\}$

Programación de Máquinas de Turing: 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 $b \in B$.
- A menudo queremos que se recuerde un símbolo en cualquier estado y entonces el conjunto de estados sería el conjunto de las parejas $Q' \times B$ formadas por un elemento $q \in Q'$ y un símbolo $b \in B$.
- Podemos considerar que un estado es $[q', b]$ donde q' es el estado básico y b el símbolo recordado.
- Escribir los estados de esta forma ayuda a comprender el significado de los mismos y se usa para describir las MT sin llegar al detalle de las transiciones.

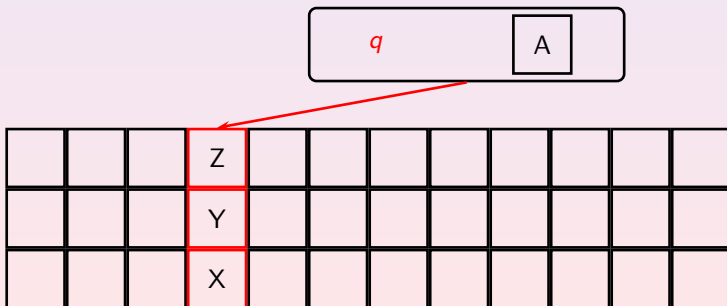
Vamos a hacer una máquina de Turing que reconozca el lenguaje $01^* + 10^*$: la Máquina tiene que recordar el primer símbolo leído y después comprobar que nunca más aparezca.

La Máquina es $M = (Q, \{0, 1\}, \{0, 1, \#\}, \delta, [q_0, \#], \{[q_1, \#]\})$ donde

- $Q = \{q_0, q_1\} \times \{0, 1, \#\}$
- Las posibles transiciones de δ son:
 - 1 $\delta([q_0, \#], a) = ([q_1, a], a, D)$ para $a = 0$ o $a = 1$.
 - 2 $\delta([q_1, a], \bar{a}) = ([q_1, a], \bar{a}, D)$, donde \bar{a} es el complementario de a (esto es, $\bar{a} = 1$ si $a = 0$ y $\bar{a} = 0$ si $a = 1$).
 - 3 $\delta([q_1, a], \#) = ([q_1, \#], \#, D)$

Programación de Máquinas de Turing: 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 $B \times B$ y tener k cintas a suponer que el alfabeto de trabajo es B^k . Se supone que un símbolo $a \in A$ se identifica con $(a, \#, \dots, \#)$.



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 $L = \{wcw \mid w \in \{0,1\}^+\}$.

La MT tiene los siguientes elementos

$M = (Q, A, B, \delta, [q_1, \#], [\#, \#], \{[q_9, \#]\})$ donde

- $Q = \{q_1, q_2, \dots, q_9\} \times \{0, 1\}$ (podemos recordar 0, 1.
- $B = \{0, 1, c, \#\} \times \{\#, *\}$
- $A = \{0, 1, c\}$. 0 se identifica con $[0, \#]$ y 1 se identifica con $[1, \#]$
- La función de transición δ se especifica en la siguiente pantalla.

Ejemplo: Función de Transición

a y b pueden ser 0,1.

$$\begin{aligned}\delta([q_1, \#], [a, \#]) &= ([q_2, a], [a, *], D) & \delta([q_2, a], [b, \#]) &= ([q_2, a], [b, \#], D) \\ \delta([q_2, a], [c, \#]) &= ([q_3, a], [c, \#], D) & \delta([q_3, a], [b, *]) &= ([q_3, a], [b, *], D) \\ \delta([q_3, a], [a, \#]) &= ([q_4, \#], [a, *], I) & \delta([q_4, \#], [a, *]) &= ([q_4, \#], [a, *], I) \\ \delta([q_4, \#], [c, \#]) &= ([q_5, \#], [c, \#], I) & \delta([q_5, \#], [a, \#]) &= ([q_6, \#], [a, \#], I) \\ \delta([q_6, \#], [a, \#]) &= ([q_6, \#], [a, \#], I) & \delta([q_6, \#], [a, *]) &= ([q_1, \#], [a, *], D) \\ \delta([q_5, \#], [a, *]) &= ([q_7, \#], [a, *], D) & \delta([q_7, \#], [c, \#]) &= ([q_8, \#], [c, \#], D) \\ \delta([q_8, \#], [a, *]) &= ([q_8, \#], [a, *], D) & \delta([q_8, \#], [\#, \#]) &= ([q_9, \#], [\#, \#], D)\end{aligned}$$

Leemos el primer símbolo, lo recordamos en el estado y lo marcamos como leído, pasando a q_2

Ejemplo: Función de Transición

a y b pueden ser 0,1.

$$\delta([q_1, \#], [a, \#]) = ([q_2, a], [a, *], D)$$

$$\delta([q_2, a], [c, \#]) = ([q_3, a], [c, \#], D)$$

$$\delta([q_3, a], [a, \#]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_4, \#], [c, \#]) = ([q_5, \#], [c, \#], I)$$

$$\delta([q_6, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_5, \#], [a, *]) = ([q_7, \#], [a, *], D)$$

$$\delta([q_8, \#], [a, *]) = ([q_8, \#], [a, *], D)$$

$$\delta([q_2, a], [b, \#]) = ([q_2, a], [b, \#], D)$$

$$\delta([q_3, a], [b, *]) = ([q_3, a], [b, *], D)$$

$$\delta([q_4, \#], [a, *]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_5, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_6, \#], [a, *]) = ([q_1, \#], [a, *], D)$$

$$\delta([q_7, \#], [c, \#]) = ([q_8, \#], [c, \#], D)$$

$$\delta([q_8, \#], [\#, \#]) = ([q_9, \#], [\#, \#], D)$$

En q_2 nos movemos hacia la derecha hasta que encontremos la c ,
entonces cambiamos a q_3

Ejemplo: Función de Transición

a y b pueden ser 0,1.

$$\delta([q_1, \#], [a, \#]) = ([q_2, a], [a, *], D)$$

$$\delta([q_2, a], [c, \#]) = ([q_3, a], [c, \#], D)$$

$$\delta([q_3, a], [a, \#]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_4, \#], [c, \#]) = ([q_5, \#], [c, \#], I)$$

$$\delta([q_6, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_5, \#], [a, *]) = ([q_7, \#], [a, *], D)$$

$$\delta([q_8, \#], [a, *]) = ([q_8, \#], [a, *], D)$$

$$\delta([q_2, a], [b, \#]) = ([q_2, a], [b, \#], D)$$

$$\delta([q_3, a], [b, *]) = ([q_3, a], [b, *], D)$$

$$\delta([q_4, \#], [a, *]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_5, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_6, \#], [a, *]) = ([q_1, \#], [a, *], D)$$

$$\delta([q_7, \#], [c, \#]) = ([q_8, \#], [c, \#], D)$$

$$\delta([q_8, \#], [\#, \#]) = ([q_9, \#], [\#, \#], D)$$

En q_3 saltamos todo lo marcado hasta que encontremos un símbolo no marcado. En ese momento si es el mismo que el recordado, lo marcamos y pasamos a q_4 .

Ejemplo: Función de Transición

a y b pueden ser 0,1.

$$\delta([q_1, \#], [a, \#]) = ([q_2, a], [a, *], D)$$

$$\delta([q_2, a], [c, \#]) = ([q_3, a], [c, \#], D)$$

$$\delta([q_3, a], [a, \#]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_4, \#], [c, \#]) = ([q_5, \#], [c, \#], I)$$

$$\delta([q_6, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_5, \#], [a, *]) = ([q_7, \#], [a, *], D)$$

$$\delta([q_8, \#], [a, *]) = ([q_8, \#], [a, *], D)$$

$$\delta([q_2, a], [b, \#]) = ([q_2, a], [b, \#], D)$$

$$\delta([q_3, a], [b, *]) = ([q_3, a], [b, *], D)$$

$$\delta([q_4, \#], [a, *]) = ([q_4, \#], [a, *], I)$$

$$\delta([q_5, \#], [a, \#]) = ([q_6, \#], [a, \#], I)$$

$$\delta([q_6, \#], [a, *]) = ([q_1, \#], [a, *], D)$$

$$\delta([q_7, \#], [c, \#]) = ([q_8, \#], [c, \#], D)$$

$$\delta([q_8, \#], [\#, \#]) = ([q_9, \#], [\#, \#], D)$$

En q_4 nos movemos a la izquierda pasando a q_5 cuando pasemos por la c y a q_6 cuando pasemos por un símbolo no marcado hasta encontrar un símbolo marcado, cuando pasamos a q_1 y repetimos.

Ejemplo: Función de Transición

a y b pueden ser 0,1.

$$\begin{array}{ll}\delta([q_1, \#], [a, \#]) = ([q_2, a], [a, *], D) & \delta([q_2, a], [b, \#]) = ([q_2, a], [b, \#], D) \\ \delta([q_2, a], [c, \#]) = ([q_3, a], [c, \#], D) & \delta([q_3, a], [b, *]) = ([q_3, a], [b, *], D) \\ \delta([q_3, a], [a, \#]) = ([q_4, \#], [a, *], I) & \delta([q_4, \#], [a, *]) = ([q_4, \#], [a, *], I) \\ \delta([q_4, \#], [c, \#]) = ([q_5, \#], [c, \#], I) & \delta([q_5, \#], [a, \#]) = ([q_6, \#], [a, \#], I) \\ \delta([q_6, \#], [a, \#]) = ([q_6, \#], [a, \#], I) & \delta([q_6, \#], [a, *]) = ([q_1, \#], [a, *], D) \\ \delta([q_5, \#], [a, *]) = ([q_7, \#], [a, *], D) & \delta([q_7, \#], [c, \#]) = ([q_8, \#], [c, \#], D) \\ \delta([q_8, \#], [a, *]) = ([q_8, \#], [a, *], D) & \delta([q_8, \#], [\#, \#]) = ([q_9, \#], [\#, \#], D)\end{array}$$

Si en q_5 encontramos un símbolo marcado es que hemos terminado de analizar w , ahora nos queda comprobar que no quedan símbolos no marcados a la derecha (con estados q_7 y q_8)

Subrutinas

- 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.
- 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

- Vamos a diseñar una MT que multiplique números en binario escritos en base 1: la MT comenzará con una cadena de la forma $0^m 10^n$ en la cinta, y terminará con 0^{mn} al final. No nos preocupamos si 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 $0^i 10^n 10^{kn}$ donde $i + k = m$ para valores de $k = 1, \dots, m$.
- 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 $0^i 10^n 10^{kn}$ a $0^{i-1} 10^n 10^{(k+1)n}$.
- Finalmente la subcadena $10^n 1$ del principio se sustituye por blancos.

Ejemplo: Subrutina

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:

$$\delta(q_1, 0) = (q_2, X, D)$$

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

$$\delta(q_2, \#) = (q_3, 0, I)$$

$$\delta(q_3, 0) = (q_3, 0, I) \quad \delta(q_3, 1) = (q_3, 1, I)$$

$$\delta(q_3, X) = (q_1, X, D) \quad \delta(q_1, 1) = (q_4, 1, I)$$

$$\delta(q_4, X) = (q_4, 0, I) \quad \delta(q_4, 1) = (q_5, 1, D)$$

Estructura del Programa

- Pasamos de la configuración en la que hay $0^m 10^n$ en la cinta y estamos colocados al principio de esta palabra a la configuración en la que en la cinta hay $0^m 10^n 1$ y estamos colocados al principio de la palabra.

$$\begin{aligned}\delta(q_0, 0) &= (q_0, 0, D) & \delta(q_0, 1) &= (q_0, 1, D) \\ \delta(q_0, \#) &= (q_{13}, 1, I) & \delta(q_{13}, 0) &= (q_{13}, 0, I) \\ \delta(q_{13}, 1) &= (q_{13}, 1, I) & \delta(q_{13}, \#) &= (q_{14}, \#, D)\end{aligned}$$

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

$$\begin{aligned}\delta(q_{14}, 0) &= (q_6, \#, D) \\ \delta(q_6, 0) &= (q_6, 0, D) & \delta(q_6, 1) &= (q_1, 1, D)\end{aligned}$$

Estructura del Programa (II)

- 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):

$$\delta(q_5, 0) = (q_7, 0, I)$$

$$\delta(q_7, 1) = (q_8, 1, I)$$

$$\delta(q_8, 0) = (q_9, 0, I)$$

$$\delta(q_8, \#) = (q_{10}, \#, D)$$

$$\delta(q_9, 0) = (q_9, 0, I)$$

$$\delta(q_9, \#) = (q_{14}, \#, D)$$

$$\delta(q_{10}, 1) = (q_{11}, \#, D)$$

$$\delta(q_{11}, 0) = (q_{11}, \#, D) \quad \delta(q_{11}, 1) = (q_{12}, \#, D)$$

Descripción de Máquinas de Turing

- Para describir MTs se pueden usar diagramas de transición similares a los autómatas finitos, pero no los vamos a usar en esta asignatura.
- Normas para describir una MT
 - Se puede pedir descripción detallada o algorítmica.
 - En el primer caso:
 - 1 Dar una idea global del funcionamiento de la MT
 - 2 Se dan las instrucciones concretas de la MT, pero agrupando las instrucciones por grupos, e ´indicando que acción realiza cada grupo.
 - En el caso caso dar una descripción global como un algoritmo, describiendo acciones que se sabe que se pueden traducir en transiciones. Por ejemplo: 'Moverse a la derecha hasta encontrar un blanco'

Variaciones de la MT básica

- Extensiones:

- **MT que se pueden quedar en la misma posición en un paso:** no es necesario moverse a la izquierda o derecha y pueden quedarse en el mismo sitio (S).
- **MT con múltiples cintas:** hay distintas cintas en las que se puede leer o escribir y una cabeza de lectura para cada una de ellas.
- **MT no deterministas:** hay distintas transiciones que puede realizar una MT en una configuración dada.

- Limitaciones:

- **MT con cintas semiilimitadas:** la cinta de la MT es ilimitada sólo por la derecha y en la izquierda tiene un tope.

Ninguna de estas modificaciones cambiará la potencialidad de las MT.

MT que se pueden quedar en la misma posición en un paso

- En estas MT se supondrá que $\delta(q, b)$ puede ser vacío (no definido) o una tripleta (p, c, M) donde $p \in Q, c \in B, M \in \{I, D, S\}$. El símbolo S indica que el cabezal de lectura no se mueve en ninguna dirección y permanece en el mismo sitio.
- Esto no supone ninguna potencialidad adicional, ya que si $\delta(q, b) = (p, c, S)$, esto lo podemos simular con un nuevo estado r_p por cada estado p de la MT con estos movimientos haciendo, $\delta(q, b) = (r_p, c, D)$ y desde todos los estados r_p lo único que se puede hacer es movernos a la izquierda: $\delta(r_p, d) = (p, d, I)$ para todo estado r_p y todo símbolo d de B .

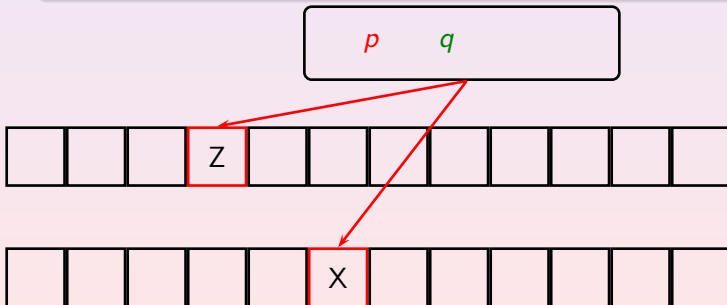
MT con Múltiples Cintas

- En estas MTs suponemos que existen k cintas ilimitadas en las que leer y escribir.
- Las diferencias entre múltiples pistas y múltiples cintas son:
 - Múltiples pistas no supone una modificación de la definición de MT, simplemente es una forma de visualizar la cinta de una MT en la que el alfabeto de trabajo es un producto cartesiano $B = B' \times B'$, es decir cada símbolo está formado por una pareja de símbolos básicos. El tener múltiples cintas si implicará una modificación de la definición. Ahora a cada (q, b_1, \dots, b_k) , δ le podrá asignar un vector $(p, c_1, \dots, c_k, M_1, \dots, M_k)$.
 - Cuando hay múltiples cintas, el cabezal de lectura podrá estar en una posición distinta en cada cinta. Por eso hay que especificar qué movimiento M_i hay que realizar en cada cinta i , además de lo que se ve en cada cinta b_i y lo que se escribe en cada una c_i . Suponemos que el movimiento de cada cinta puede ser $\{I, D, S\}$.

MT con Múltiples Cintas

Configuración

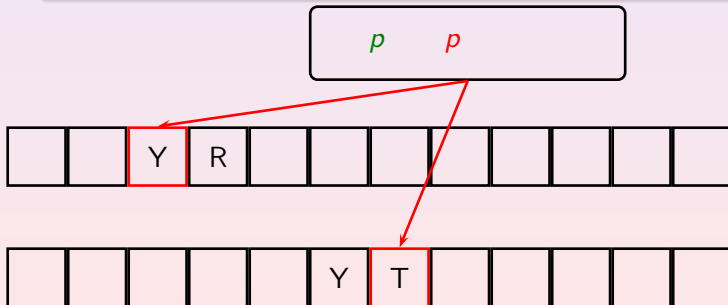
Será un vector $(q, u_1, w_1, u_2, w_2, \dots, u_k, w_k)$, donde q es el estado en el que está la MT, u_i es la parte de la palabra que hay a la izquierda del cabezal de lectura de la cinta i y w_i la parte de la palabra que hay en la cinta i a partir del cabezal de lectura de esa cinta hacia la derecha (incluyendo el símbolo que se lee en ese momento).



MT con Múltiples Cintas

Configuración

Será un vector $(q, u_1, w_1, u_2, w_2, \dots, u_k, w_k)$, donde q es el estado en el que está la MT, u_i es la parte de la palabra que hay a la izquierda del cabezal de lectura de la cinta i y w_i la parte de la palabra que hay en la cinta i a partir del cabezal de lectura de esa cinta hacia la derecha (incluyendo el símbolo que se lee en ese momento).



Lenguaje Aceptado y Función Calculada

Lenguaje Aceptado

El aceptado por una MT con múltiples cintas es el conjunto de palabras u tales que empezando en una configuración en la que en la primera cinta está la palabra u y el resto de las cintas son vacías y el cabezal de lectura de la primera cinta está en el primer símbolo de u y en cualquier casilla de las otras cintas termina en un estado de aceptación.

Función Calculada

La función parcial f calculada por una MT es la función definida en todas las entradas en las que la MT termina. Si $u \in A^*$ es una entrada para la que la MT termina, entonces lo hace con $f(u)$ como contenido de la última cinta excluyendo los blancos.

Está claro que si un lenguaje es aceptado por una MT con una cinta puede hacerlo por una MT con varias cintas. El recíproco también es cierto.

Teorema

Todo lenguaje aceptado por una MT con varias cintas es también aceptado por una MT de una cinta.

Supongamos que M es la MT con k cintas, vamos a simular su funcionamiento con una MT N con una sola cinta.

Está claro que si un lenguaje es aceptado por una MT con una cinta puede hacerlo por una MT con varias cintas. El recíproco también es cierto.

Teorema

Todo lenguaje aceptado por una MT con varias cintas es también aceptado por una MT de una cinta.

Supongamos que M es la MT con k cintas, vamos a simular su funcionamiento con una MT N con una sola cinta.

La MT N tendrá una cinta con $2k$ pistas. En cada par de pistas se simula una cinta de M . En una de las pistas se coloca un símbolo especial $*$ en el lugar en el que se encuentre el cabezal de la MT de k pistas, en la otra pista se coloca el contenido de la cinta de M .

Equivalencia (Cont.)

La MT N almacena en la unidad de control los k símbolos que contiene M . Para ello comienza a revisar la cinta de izquierda a derecha y cada vez que encuentra un símbolo lo almacena en su correspondiente lugar de la unidad de control. Lleva un contador donde empieza en 0 y se aumenta en 1 cada vez que encuentra un símbolo hasta llegar a k (el valor de k es fijo y se puede almacenar).

Una vez hecho esto, tiene todos los elementos para realizar la transición de la MT de k cintas, para ello se va colocando en cada una de las posiciones señaladas de cada una de las cintas y realiza la transición correspondiente, escribiendo el símbolo que corresponda y moviendo la señal de posición del cabezal de lectura.

Equivalencia (Cont.)

La MT N almacena en la unidad de control los k símbolos que contiene M . Para ello comienza a revisar la cinta de izquierda a derecha y cada vez que encuentra un símbolo lo almacena en su correspondiente lugar de la unidad de control. Lleva un contador donde empieza en 0 y se aumenta en 1 cada vez que encuentra un símbolo hasta llegar a k (el valor de k es fijo y se puede almacenar).

Una vez hecho esto, tiene todos los elementos para realizar la transición de la MT de k cintas, para ello se va colocando en cada una de las posiciones señaladas de cada una de las cintas y realiza la transición correspondiente, escribiendo el símbolo que corresponda y moviendo la señal de posición del cabezal de lectura.

Los estados de aceptación de N son aquellos estados que corresponden a los estados de aceptación de M .

Simulación: complejidad en tiempo

Teorema

Si la MT M del teorema anterior emplea un número de pasos inferior o igual a $t(n)$ para una entrada de longitud n , entonces la MT N de una cinta emplean un número de pasos de orden $O(t^2(n))$.

Para la demostración, basta tener en cuenta:

- Después de $t(n)$ movimientos de la MT M la diferencia entre las posiciones de los distintos cabezales de lectura es, a lo más, $2t(n)$. Al principio podemos suponer que todos están en la misma posición y, en cada paso, se alejan a lo más en dos posiciones (si dos cabezales se mueven en distintas direcciones).
- Para ver los contenidos de las distintas casillas en las cintas en la máquina N nos hacen falta $2t(n)$ pasos como máximo.
- Ahora nos movemos de derecha a izquierda y cada vez que encontramos un marcador de posicionamiento, realizamos la transición correspondiente. Esta transición, a lo más, necesita 2 movimientos (para llevarlo a cabo y volver a la posición en la que estábamos). Esto implica $2t(n) + 2k$ movimientos.
- Si le sumamos a los $2t(n)$ iniciales, hacen $4t(n) + 2k$ por cada movimiento, así que los $t(n)$ movimientos implicarán como máximo $t(n)(4t(n) + 2k)$ que es de orden $O(t^2(n))$, teniendo en cuenta que k es constante.

Máquinas de Turing No Deterministas

Definición

Una **Máquinas de Turing No Determinista (MTND)** tiene la misma definición que una MT con la única diferencia que ahora $\delta(q, a)$ puede ser un conjunto finito de tripletas

$$\{(q_1, b_1, M_1), \dots, (q_k, b_k, M_k)\}$$

Cálculo

El cálculo asociado a una MTND se define de forma similar a una MT. Ahora en una configuración en la que está en el estado q y ve a en la cinta puede evolucionar con cualquiera de las tripletas (q_i, b_i, M_i) : puede ir a q_i escribir b_i y hacer el movimiento M_i para $i = 1, \dots, k$.

Lenguaje Aceptado

Es el conjunto de todas las palabras aceptadas.

Acepta una palabra cuando para la configuración inicial asociada a la palabra, existe una sucesión de movimientos posibles que permiten llegar a un estado de aceptación y parar. No importa que haya otras computaciones posibles que no lleguen a un estado de aceptación.

Toda MT es una MTND por lo que todo lenguaje aceptado por una MT determinista es también aceptado por una MTND. También se da el inverso.

Teorema

Si un lenguaje L es aceptado por una MTND, entonces es recursivamente enumerable.

Suponemos m el número máximo de opciones en la MTND.

- Utilizamos una MT con dos cintas.
- En la primera tenemos una sucesión de configuraciones (q, u, v) separadas por un símbolo \square , también existe una marca $*$ en la *configuración activa*.
- Inicialmente hay sólo una configuración: la configuración inicial.

Equivalencia (Cont.)

- En cada momento se coge la configuración activa, se copia en la cinta auxiliar, nos vamos al final de la primera cinta y se realizan todas las transiciones posibles sobre la configuración de la cinta auxiliar colocándolas al final de la primera cinta.
- Se busca la configuración marcada y se pasa a procesar la siguiente.
- Si en algún momento sale una configuración con un estado de aceptación terminamos.
- El procedimiento hace una búsqueda en anchura exhaustiva de todos los posibles cálculos de la MTND. Si en uno se acepta, lo encontrará. Si en un nivel, todos los cálculos han terminado sin aceptar, rechaza.

Ejemplo de MT para saber si un número binario es compuesto (no es primo)

- Supongamos que u es la entrada.
- la MTND elige de forma no determinista un número en binario v de longitud menor o igual a u
- Realiza la división de u entre v
- Si la división es exacta acepta que es compuesto. En caso contrario rechaza.

Observemos como si el número **es** compuesto, **AL MENOS** una de las opciones posibles acaba en **aceptación** y si **no** lo es **TODAS** las opciones acaban en **rechazo**.

Ejemplo de MT para saber si hay un camino entre dos nodos del grafo.

- Inicialmente la MT tiene en la cinta codificado un grafo y un par de nodos.
- la MTND escribe de forma no determinista una lista de nodos que comienza en el primer nodo del par y termina en el último nodo del par. Podemos suponer que no repite nodos.
- Comprueba si hay un enlace entre cada nodo de la lista y el siguiente
- Si el resultado es positivo para todos los nodos de la lista acepta. En caso contrario rechaza.

Observemos como si **existe** un camino, **AL MENOS** una de las opciones posibles acaba en **aceptación** y si **no** existe **TODAS** las opciones acaban en **rechazo**.

Número de Pasos (tiempo) en MT

- En una MT el número de pasos (tiempo) para una entrada u es el número de pasos de cálculo entre la configuración de entrada y la última configuración.
- Una MT tiene complejidad $t(n)$ en tiempo si para toda entrada de longitud n la MT termina en $t(n)$ o menos pasos.

Número de Pasos (tiempo) en MTND

- En una MTND el número de pasos (tiempo) para una entrada u es el número de pasos para el cálculo más largo posible para esa entrada. Si hay una secuencia de cálculos que no termina, entonces el tiempo es infinito.
- Si decimos que una MTND tiene complejidad $t(n)$ en tiempo, quiere decir que todos los posibles cálculos de la MTND terminan en $t(n)$ o menos pasos donde n es la longitud de la entrada.
- Si una MTND tiene complejidad $t(n)$, veremos (estudio de la complejidad algorítmica) que la MT que la simula tiene complejidad $O(d^{t(n)})$ donde d es una constante mayor que uno.

Ventaja de las MTNDs

- Son útiles para resolver problemas donde tenemos que **buscar** un elemento en una población **que cumpla una condición**.
- La MTND **elige** de forma no-determinista un elemento de la población y **comprueba** si cumple condición.
- Si la cumple acepta y en otro caso rechaza.
- Observemos como son problemas que llevan un proceso de búsqueda asociado. La MTND **no tiene que describir el procedimiento de búsqueda**.
- La simulación de una MTND por una determinista es una forma de **añadir una búsqueda** que es válida en todos los casos y que da lugar a una MT determinista.

Cintas semiilimitadas

Podemos suponer que las cintas son ilimitadas sólo por la derecha y que la palabra de entrada se escribe a partir de la primera casilla. Para ello, vamos a demostrar que todo lenguaje se puede aceptar sin escribir a la izquierda de la primera casilla que ocupa la palabra de entrada, lo que equivale a suponer que no existe cinta a la izquierda de la palabra de entrada.

Teorema

Todo lenguaje aceptado por una MT M_2 es también aceptado por una MT M_1 con las siguientes restricciones:

- 1 M_1 nunca escribe el espacio en blanco $\#$
- 2 La cabeza de M_1 nunca se mueve hacia la izquierda de su posición inicial.

La primera condición es muy fácil de conseguir añadiendo un nuevo símbolo a M_1 que es otro espacio en blanco $\#'$.

Si M_2 escribe un espacio en blanco $\delta_2(q, a) = (p, \#, M)$, entonces M_1 escribe el nuevo blanco: $\delta_1(q, a) = (p, \#', M)$. Después, cada transición $\delta_1(q, \#')$ se hace idéntica a $\delta_2(q, \#)$.

Cintas semiilimitadas

Para la segunda condición, una cinta ilimitada por ambos extremos se simula con una cinta ilimitada sólo por la derecha con dos pistas y la siguiente estructura:

X_0	X_1	X_2								
\triangleright	X_{-1}	X_{-2}								

Si $M_2 = (Q_2, A, B_2, \delta_2, q_2, \#, F_2)$ entonces

$M_1 = (Q_1, A \times \{\#\}, B_1, \delta_1, q_0, [\#, \#], F_1)$ donde

- Los estados de M_1 son $\{q_0, q_1\} \cup (Q_2 \times \{S, I\})$. Los estados q_0 y q_1 sirven para preparar la cinta de entrada (por ejemplo, poner el tope \triangleright en la pista inferior). En los otros estados tenemos que especificar el valor S (pista superior) o I (pista inferior) además del estado.
- Los símbolos de trabajo de M_1 son $B_2 \times B_2$, es decir todas las parejas de símbolos de trabajo de M_2 . Cada símbolo $a \in A_2$ de M_2 se identifica con el símbolo $[a, \#]$ de M_1 . Además en B_1 están todas las parejas $[b, \triangleright]$ donde $b \in B_2$. Este símbolo \triangleright se usa como tope para saber que estamos en el extremo izquierdo de la cinta.

Cintas semiilimitadas

- $\delta_1(q_0, [a, \#]) = (q_1, [a, \triangleright], D)$ para cualquier $a \in B_2$
- $\delta_1(q_1, [a, \#]) = ([q_2, S], [a, \#], I)$ (nos movemos a la izquierda y decimos que estamos arriba)
- Si $\delta_2(q, a) = (p, b, M)$, entonces para todo $c \in B_2$
 - 1 $\delta_1([q, S], [a, c]) = ([p, S], [b, c], M)$
 - 2 $\delta_1([q, I], [c, a]) = ([p, I], [c, b], \overline{M})$, donde \overline{M} es el movimiento de sentido opuesto a M .

Cintas semiilimitadas

- Si $\delta_2(q, a) = (p, b, D)$ entonces

$$\delta_1([q, I], [a, \triangleright]) = \delta_1([q, S], [a, \triangleright]) = ([p, S], [b, \triangleright], D)$$

Si estamos al principio de la cinta, siempre se supone que el símbolo activo es el superior. Allí se escribe y si nos movemos a la derecha el superior es el activo.

- Si $\delta_2(q, a) = (p, b, I)$ entonces

$$\delta_1([q, I], [a, \triangleright]) = \delta_1([q, S], [a, \triangleright]) = ([p, I], [b, \triangleright], D)$$

Si estamos al principio de la cinta, siempre se supone que el símbolo activo es el superior. Allí se escribe y si nos movemos a la izquierda el inferior es el activo.

Los estados de aceptación F_1 de M_1 es el conjunto de estados $F_2 \times \{S, I\}$.

- Cuando usamos cintas semiilimitadas, supondremos desde el principio que tenemos el símbolo \triangleright a la izquierda de cada una de las cintas que se usen y que si se llega a ese símbolo inmediatamente nos vamos hacia la derecha en el próximo movimiento.
- Esto no supone ninguna limitación, ya que toda MT se puede simular con una con estas restricciones: siempre podemos incluir ese símbolo al principio de la palabra de entrada, desplazando todos los símbolos una casilla a la derecha y después como hemos visto siempre se puede simular la MT con una de cintas semiilimitadas en las que nunca se pasa a la izquierda de la palabra de entrada.