UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

FUNDAMENTOS TEÓRICOS DE LA INFORMÁTICA

Lic. Martin Villarreal

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Máquina de Turing:

  - En 1936 Turing definió una máquina abstracta, conocida hoy como máquina de Turing.

  - Su idea central era abstraer el conjunto de operaciones que realiza una persona cuando realiza un cálculo.

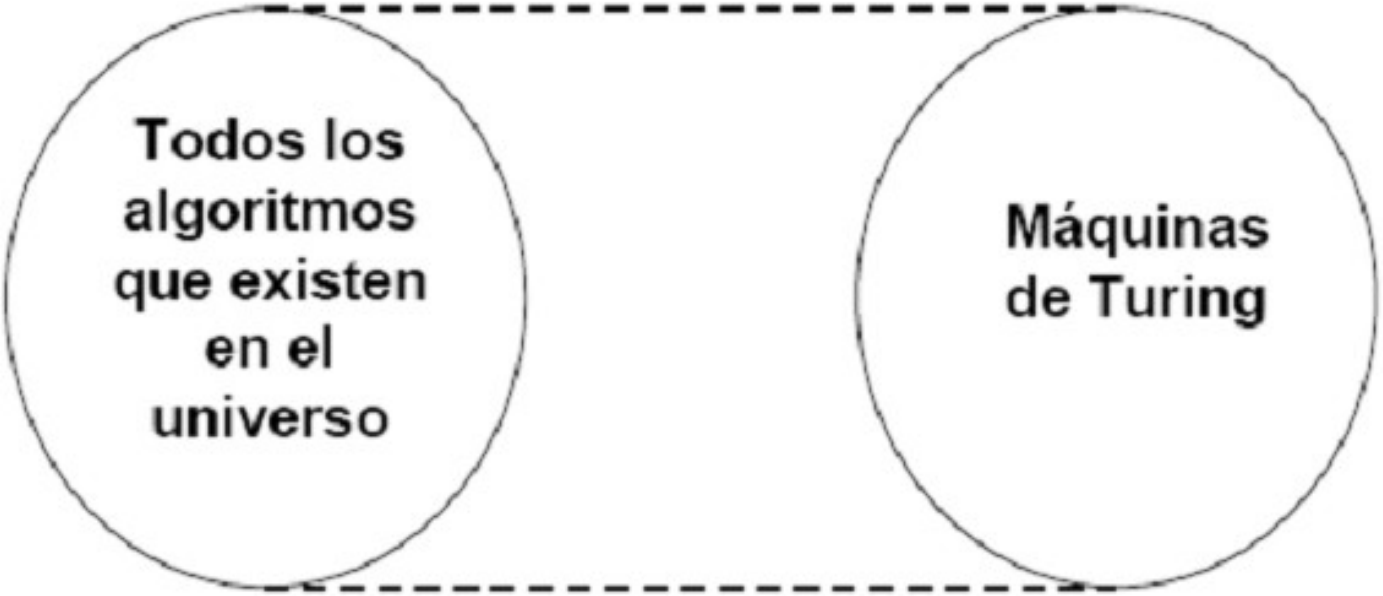
  - El gran aporte de Turing es establecer que una máquina de Turing es capaz de ejecutar cualquier procedimiento efectivo.

- **Procedimiento efectivo: es un conjunto de reglas, destinadas a resolver un problema, escritas en un determinado lenguaje, que son interpretables y ejecutables.**

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

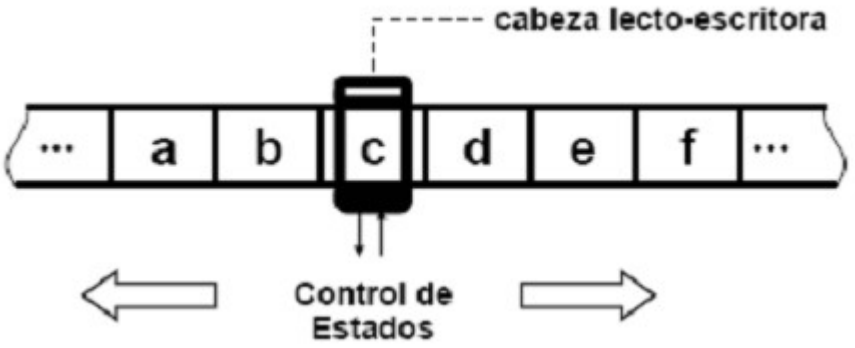
- Un resultado importante



Si existe un algoritmo para realizar cierta tarea, entonces también existe una máquina de Turing que realice la misma tarea y viceversa.

Máquinas de Turing

- Como funciona?



- La máquina posee una cabeza lecto-escritora que puede leer y escribir símbolos en una cinta (puede convertir la entrada en salida)

- La cabeza lecto-escritora puede realizar movimientos a izquierda y derecha

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

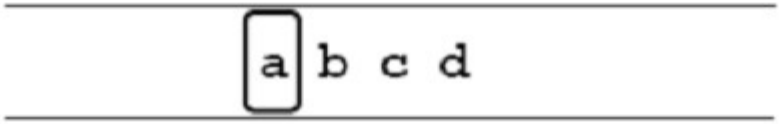
- Convenciones:

  - La cinta se extiende infinitamente en ambas direcciones.

  - En cada posición de la cinta solo se puede almacenar un símbolo.

  - Los contenidos de las celdas se pueden acceder en cualquier orden.

  - Inicialmente la cabeza lecto-escritora está ubicada en el primer caracter de la entrada, es decir el primer carácter no blanco de la cinta.



  - combinaremos operaciones de escritura y desplazmiento para abreviar algunas definiciones.

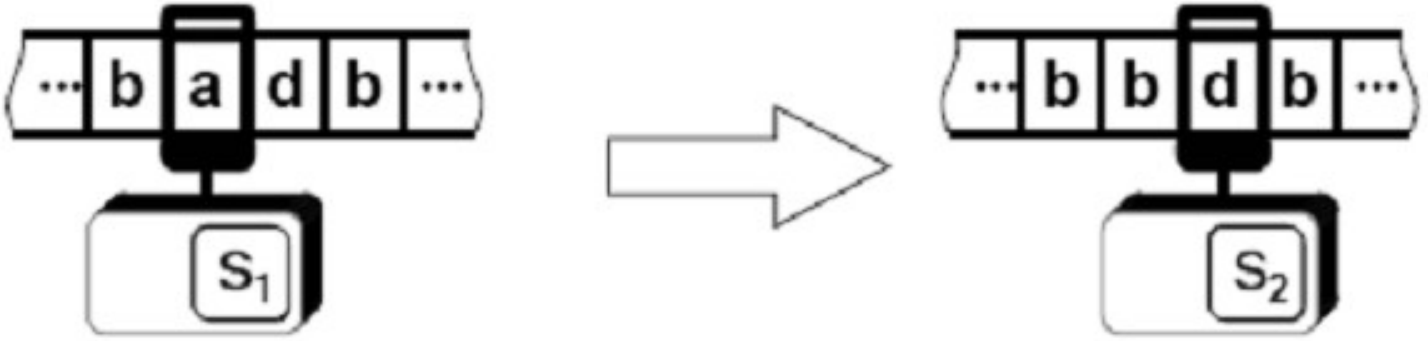
Máquinas de Turing

- δ es una función parcial, δ:QxΓ→QxΓx{I,N,D} donde I=movimiento izquierda, D=movimiento derecha y N=no mover.

$\delta(S_1, a)=(S_2, b, D)$

Si el estado actual es S_1, el símbolo bajo la cabeza de la MT es a, entonces

pasar al estado S_2, escribir en la cinta el símbolo b, y mover la cabeza a la derecha.



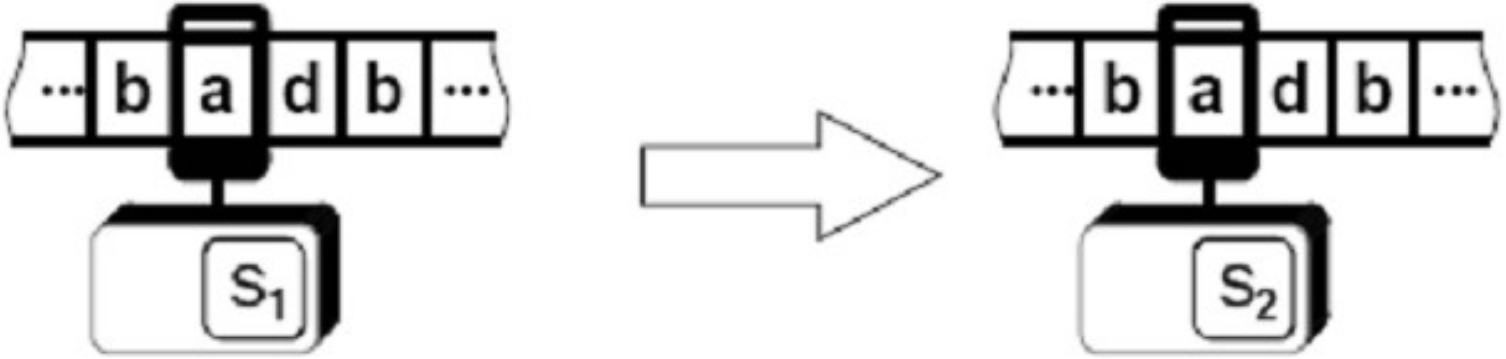
Máquinas de Turing

- δ es una función parcial, δ:QxΓ→QxΓx{I,N,D} donde I=movimiento izquierda, D=movimiento derecha y N=no mover.

$\delta(S_1, a)=(S_2, a, N)$ {

Si el estado actual es S1, el símbolo bajo la cabeza de la MT es a, entonces

pasar al estado S2, escribir en la cinta el símbolo a, y detenerse



UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Máquina de Turing: Se define como la 7-upla T=(Q, Σ, Γ, ƀ, q0, δ, F) donde**

Q es un conjunto finito de estados

Σ es el alfabeto de entrada

Γ es el alfabeto de la cinta. Σ∪{ƀ}⊆ Γ

ƀ ∈ Γ es el símbolo blanco y ƀ ∉ Σ

q0 ∈ Q es el estado inicial

δ: QxΓ→QxΓx{I,D,N} es una función parcial donde {I,N,D} son posibles movimientos de la cabeza lecto-escritora: Izquierda, Derecha y no hay movimiento respectivamente.

F ⊆ Q es el conjunto de estados finales.

Máquinas de Turing

- Ejemplo: Desarrollar una MT para reconocer cadenas de $L = \{a^{2n} \mid n \geq 0\}$

  -Sea $T=(S, \Sigma, \Gamma, ƀ, s0, \delta, F)$ una MT, donde

  $S = \{s0, s1\}$, $\Sigma = \{a\}$,$\Gamma=\{a,ƀ\}$, $F = \{s0\}$, y la función se define por:

  $(s0, a) = (s1, a, D)$

  $(s1, a) = (s0, a, D)$

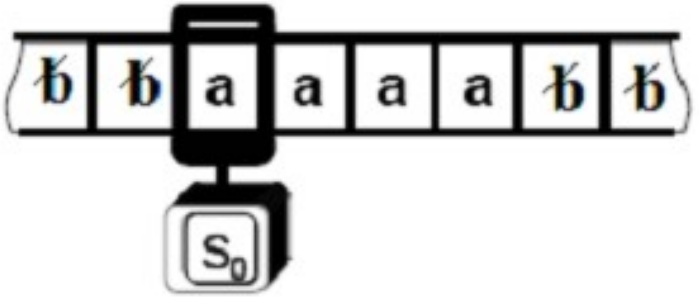
  - Esta MT resuelve el problema de reconocer L.

  - Veamos como se comporta T para la cadena aaaa

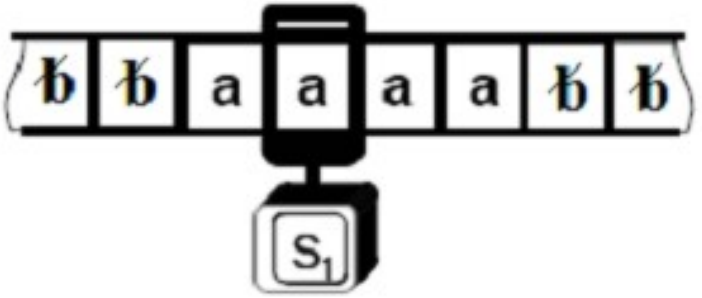
UNTDF
Universidad Nacional de Tierra del Fuego,
Antártida e Islas del Atlántico Sur

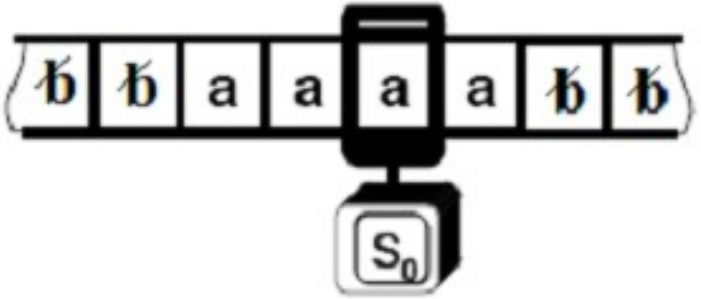
Máquinas de Turing

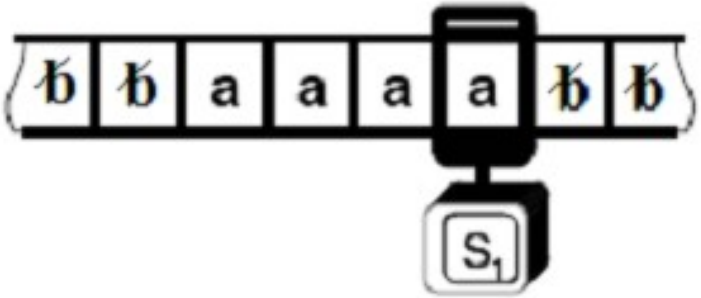
$\delta(s_0, a) = (s_1, a, D)$

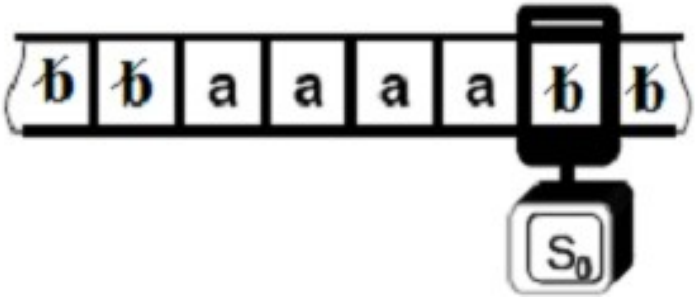


$\delta(s_1, a) = (s_0, a, D)$









UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejercicio: Considere la MT definida por

  Q = {q1, q2}

  Σ= {a, b}

  Γ= {a, b, ƀ}

  F = {q2}

  s = q1

  y δ definido por

  δ(q1, a) = (q1, a,D)

  δ(q1, b) = (q1, a,D)

  δ(q1, ƀ) = (q2, ƀ, I)

- Indique el comportamiento de la MT para la cadena aabb

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Configuración: la configuración de la MT viene dada por: el estado actual, el contenido de la cinta y la posición de la cabeza lecto-escritora.
  - Notacion 1:

  (qi,w1σw2)

  donde
    - qi es el estado actual
    - w1 es la cadena que precede la celda donde se encuentra la cabeza lecto-escritora, σ es el símbolo de la cinta donde se encuentra la cabeza lecto-escritora.
    - w2 es la cadena que aparece a continuación.
- El ejemplo anterior podría expresarse como

  (q1, aabb) ⊢ (q1, aabb) ⊢ (q1, aabb) ⊢ (q1, aaab) ⊢ (q1, aaaaƀ) ⊢ (q2, aaaa)

Máquinas de Turing

- Notacion 2 : Una notacion alternativa para (qi,w $\underline{a_k}$ u) es la siguiente

$a_1 a_2 \ldots a_{k-1}\ q_i\ a_k \ldots a_n$

donde

- qi es el estado actual.
- La cabeza lectora-escritora se encuentra posicionada sobre a_k

El ejemplo anterior se podria expresar como:

(q1aabb) ⊢ (aq1abb) ⊢ (aaq1bb) ⊢ (aaaq1b) ⊢ (aaaaq1Ь) ⊢ (aaaq2a)

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejercicios:

  - Desarrollar una MT que reconozca palabra generadas con los símbolos del alfabeto {a, b} comiencen con la letra a.

  - Desarrollar una MT que reconozca palabra generadas de la misma forma que en el ejercicio anterior pero que terminen con la letra a.

  - Verifique como se comportan las MT anteriores para las palabras abaab y bbaa.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- En relación a las configuraciones decimos que:

  - Si δ(qi, a) no esta definido y la configuración de la máquina de Turing es (qi,w1aw2), no es posible pasar a otra configuración. En este caso decimos que la máquina de Turing está parada.

  - La secuencia de todos los movimientos que conducen a una configuración de parada se llama computación.

  - Vamos a requerir que nuestra máquina de Turing pare siempre que llegue a un estado final, por lo que no definiremos ninguna transición para los estados finales.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Loop infinito:

Considere la MT definida por

Q = {q1, q2}, Σ = {a, b}, Γ= {a, b, ƀ}, F = {⊘}, s = q1

y δ definido por

δ(q1, a) = (q2, a, D)
δ(q1, b) = (q2, b, D)
δ(q1, ƀ) = (q2, ƀ, D)
δ(q2, a) = (q1, a, I)
δ(q2, b) = (q1, b, I)
δ(q2, ƀ) = (q2, ƀ, I)

- Indique el comportamiento de la MT para la cadena abab

- Esta situación de loop infinito la representaremos con

(q,w1σw2) ⊦* ∞ o w1qσw2 ⊦* ∞

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Cadena aceptada por la MT: una cadena w ∈ Σ* , es aceptada por una MT, si comienza en el estado inicial, con la cabeza lecto-escritora en el símbolo más a la izquierda y luego de leer toda la cadena w, llega a un estado qf ∈ F y para.**

- **Sea M=(Q, Σ, Γ, ƀ, q_0, δ, F) una máquina de Turing, el lenguaje aceptado por M es L(M) = {w ∈ Σ*| q_0w ⊢* w1pw2 para p ∈ F y w1, w2 ∈ Γ*}**
  (note que la MT para cuando llega al estado final)

Máquinas de Turing

- Ejemplo: Desarrollar una máquina de Turing para reconocer cadenas del lenguaje

$L = \{a^n \mid n \geq 0\}$

- Basta con definir una máquina de Turing $M=(Q, \Sigma, \Gamma, ƀ, q_1, \delta, F)$

donde:

$Q = \{q1, q2\}$, $\Sigma = \{a\}$; $\Gamma = \Sigma \cup \{ƀ\}$; $F = \{q2\}$

y δ definido por

$\delta(q1, a) = (q1, a, D)$

$\delta(q1, ƀ) = (q2, ƀ, N)$

- Indique el comportamiento de la MT para la cadena aaa

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejemplo: Desarrollar una máquina de Turing para reconocer cadenas del lenguaje

$L = \{a^n b^n \mid n \geq 1\}$

- Se utilizara $M=(Q, \Sigma, \Gamma, ƀ, q_1, \delta, F)$ donde:

$\Sigma = \{a, b\}$, $F = \{q5\}$

$Q = \{q1, q2, q3, q4, q5\}$, $\Gamma = \{a, b, c, d, ƀ\}$,

y δ definido por

$\delta(q1, a) = (q2, c, D)$ $\quad$ $\delta(q2, a) = (q2, a, D)$

$\delta(q2, d) = (q2, d, D)$ $\quad$ $\delta(q2, b) = (q3, d, I)$

$\delta(q3, d) = (q3, d, I)$ $\quad$ $\delta(q3, a) = (q3, a, I)$

$\delta(q3, c) = (q1, c, D)$

$\delta(q1, d) = (q4, d, D)$ $\quad$ $\delta(q4, d) = (q4, d, D)$

$\delta(q4, ƀ) = (q5, ƀ, I)$

- Indique el comportamiento de la MT para la cadena aaabbb

Máquinas de Turing

- Uso de máquinas de Turing para computar otro tipo de funciones

  - Las MT pueden ser consideradas como la implementación de una función de cadena f definida mediante f(w) = u cuando se cumple

  $s_0 w \vdash^* s_f u$

  donde s_0 es el estado inicial y s_f es un estado final.

  - Por conveniencia y claridad se requiere que la cabeza de lectura-escritura empiece y termine, respectivamente, sobre el símbolo de las cadenas de entrada y salida que esta situado más a la izquierda.

- **Definición: Se dice que una función f es Turing Computable si existe una máquina de Turing $M=(Q, \Sigma, \Gamma, ь, s_0, \delta, F)$ tal que**

  $s_0 w \vdash^* s_f u$

  para algún $s_f \in F$, cuando f(w) = u.

Máquinas de Turing

- Ejemplo: Utilizar una MT para implementar la función suma f(n, m) = n + m.

  - Supongamos que tenemos Σ= {a, b} y que representamos los enteros positivos mediante cadenas de a.

  - Así, el entero positivo n estara representado por a^n.

  - La función suma f(n, m) = n +m podrá ser implementada mediante la transformacion de a^nba^m en $a^{n+m}b$

Máquinas de Turing

- Usaremos M = ({s0, s1, s2, s3, s4}, {a, b}, {a, b, ƀ}, ƀ, s0, δ, {s4})

y δ definida de la siguiente forma:

δ(s0, a) = (s0, a,D)

δ(s0, b) = (s1, a,D)

δ(s1, a) = (s1, a,D)

δ(s1, ƀ) = (s2, ƀ, I)

δ(s2, a) = (s3, b, I)

δ(s3, a) = (s3, a, I)

δ(s3, ƀ) = (s4, ƀ,D)

- Verifique que la MT desplaza la b hacia el final, a la derecha de a^{n+m}

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejercicio: Utilizar una MT para implementar la función resta f(x, y) = x – y, siendo x e y numeros naturales representados en unario.

  - Ejemplo: 5 - 3 se representara como

  11111-111.

  - Estrategia a utilizar

    - Ir “descontando” cada 1 del minuendo contra otro 1 del sustraendo, reemplazando ambos por un caracter arbitrario (ej: *).

    - Cuando se termine el sustraendo se borran los caracteres inútiles de manera que queden solo los restos del minuendo.

    - Para evitar tener que recorrer el residuo, descontamos caracteres del minuendo de derecha a izquierda.

Máquinas de Turing

- **Combinación de máquinas de Turing**

  - Es posible construir una MT compuesta si se ejecuta primero una MT y luego otra.

  - Si T1 y T2 son máquinas distintas con conjuntos de estados y funciones de transición distintos. Se escribe T1T2 para denotar la MT compuesta y su conj. de estados es la unión de los dos conj.

  - Comienza en el estado inicial de T1 y ejecuta los movimientos de T1 usando la funcion δ1.

  - Cualquier movimiento que haga que T1 se detenga en el estado de aceptación, T1T2 ejecuta el mismo movimiento, excepto que se mueve al estado inicial de T2.

  - A partir de ese punto los movimientos de T1T2 son los de T2 usando δ2.

  - T1T2 acepta cuando lo hace T2 y solo en ese caso.

Máquinas de Turing

- **Definición: Sean T1 y T2 dos máquinas de Turing sobre el mismo alfabeto de entrada Σ y el mismo alfabeto de cinta Γ, donde:**

T1 = (S1, Σ, Γ,Ь, s1, δ1, F1) y T2 = (S2, Σ, Γ, Ь, s2, δ2, F2)

Se supone que S1 ∩ S2 = ∅.

- La composición de las máquinas de Turing T1 y T2 es la máquina de Turing T1T2 donde

S = S1 ∪ S2, s = s1, F = F2

$$\delta(s,\gamma)=\begin{cases}\delta 1(s,\gamma)\ si\ s\in S1\ y\ \delta 1(s,\gamma)\neq(p,t,X)\ para\ todo\ p\in F1\\ \delta 2(s,\gamma)\ si\ s\in S2\\ (s2,t,X)\ si\ s\in S1\ y\ \delta 1(s,\gamma)=(p,t,X)\ para\ algún\ p\in F1\end{cases}$$

Máquinas de Turing

- Ejemplo: Combinar las siguientes máquinas de turing

- **Máquina 1**: Sea T1 = (Q1,Σ, Γ,ƀ, s1, δ1, F1) donde

Q1 = {q0, q1} , Σ = {a}, Γ = {a, ƀ},s1=q0, F1 = {q1}

y la función de transición δ1

δ1(q0, a) = (q0, a,D)

δ1(q0, ƀ) = (q1, ƀ,N)

- **Máquina 2**: Sea T2 = (Q2,Σ, Γ,ƀ, s2, δ2, F2) donde

Q2 = {p0, p1} , s2=p0, F2 = {p1}

y la función de transición δ2

δ2(p0, ƀ) = (p1, a,D)

Máquinas de Turing

- T1 busca el primer blanco que encuentre a partir de donde ha comenzado, mientras que T2 reemplaza el ƀ por a y para.

- Al combinar estas dos máquinas de Turing obtenemos un dispositivo que primero busca hacia la derecha el primer blanco y despues escribe una a.

- Las transiciones de T1T2 estan dadas por:

δ(q0, a) = (q0, a, D)

δ(q0, ƀ) = (p0, ƀ, N)

δ(p0, ƀ) = (p1, a, D)

Máquinas de Turing

- Tambien podriamos construir T1 a partir de máquinas mas sencillas, por ejemplo si tengo una máquina R que se comporta de la siguiente manera: para todo símbolo de la cinta, escribe el mismo símbolo leído y se mueve a derecha y para.

- R para el alfabeto de cinta de T1 se define como R = (P,Σ, Γ,ƀ, p1, δ, F)

P={p1, p2}, Σ={a}, Γ={a, ƀ}, F={p2}

y δ se define como:

δ(p1,a)=(p2,a,R)

δ(p1,ƀ)=(p2,ƀ,R)

Máquinas de Turing

- Puedo definir una maquina ƀ que se comporta de la siguiente manera para todo símbolo de la cinta, escribe una ƀ y para.

- ƀ para el alfabeto de cinta T2 se define como ƀ=(R,Σ, Γ,ƀ, r1, δ, F)

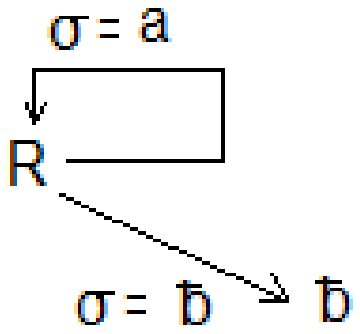
R={r1,r2}, Σ={a}, Γ={a, ƀ}, F={r2}

y δ se define como:

δ(r1,a)=(r2,ƀ,N)

δ(r1,ƀ)=(r2,ƀ,N)

- Ahora puedo representar T1 como



Me muevo a derecha, mientras lo que leo es a, me muevo a derecha.
Cuando leo ƀ escribo ƀ y paro.

Máquinas de Turing

- Podriamos llamar a la maquina obtenida “$R_ь$” y definir una máquina “a” que se comporte de manera similar a “ь” pero que escriba a (como T2). Entonces podriamos representar T1T2 como

$R_ь a$

La Máquina se mueve a derecha hasta encontrar el primer blanco y escribe a.

Máquinas de Turing

- **Máquina de Turing multipista.**

- Una modificación sencilla a la MT básica es aquella en la que cada celda de la cinta se divide en subceldas.

- Cada subcelda es capaz de contener un símbolo de la cinta.

- Se dice que esta cinta tiene multiples pistas ya que cada celda contiene varios caracteres.

- Su contenido puede ser representado mediante n-uplas ordenadas.

- Los movimientos que realice la máquina dependeran de su estado actual y de la n-upla que represente el contenido de la celda actual.

- Esto no incrementa la potencia de cálculo sino que facilita la resolución de algunos problemas.

Máquinas de Turing

- Ejemplo:

- Construir una MT que sume 2 números binarios

- Si se busca sumar 101 y 10, la cinta debería contener:

$\not b$	1	0	1	$\not b$
$\not b$	0	1	0	$\not b$
$\not b$	$\not b$	$\not b$	$\not b$	$\not b$

La MT realizará la suma en la tercer pista.

- El alfabeto de la cinta estará formado por las ternas.

$(\not b, \not b, \not b)$	$(1, 1, \not b)$	$(1, 1, 0)$	$(1, 1, 1)$	$(0, 0, \not b)$
$(0, 0, 0)$	$(0, 0, 1)$	$(\not b, \not b, 1)$	$(0, 1, \not b)$	$(0, 1, 0)$
$(0, 1, 1)$	$(1, 0, \not b)$	$(1, 0, 0)$	$(1, 0, 1)$	

Máquinas de Turing

$\not b$	1	0	1	$\not b$
$\not b$	0	1	0	$\not b$
$\not b$	$\not b$	$\not b$	$\not b$	$\not b$

La función δ será

$$\delta(s_0,(\not b,\not b,\not b)) = (s_1,(\not b,\not b,\not b)),I) \qquad \delta(s_0,(1,1,\not b)) = (s_0,(1,1,\not b),D)$$
$$\delta(s_0,(0,1,\not b)) = (s_0,(0,1,\not b),D) \qquad \delta(s_0,(1,0,\not b)) = (s_0,(1,0,\not b),D)$$
$$\delta(s_0,(0,0,\not b)) = (s_0,(0,0,\not b),D)$$

$$\delta(s_1,(0,0,\not b)) = (s_1,(0,0,0),I) \qquad \delta(s_1,(0,1,\not b)) = (s_1,(0,1,1),I)$$
$$\delta(s_1,(1,0,\not b)) = (s_1,(1,0,1),I) \qquad \delta(s_1,(1,1,\not b)) = (s_2,(1,1,0),I)$$
$$\delta(s_1,(\not b,\not b,\not b)) = (s_3,(\not b,\not b,0),N)$$

$$\delta(s_2,(0,0,\not b)) = (s_2,(0,0,1),I) \qquad \delta(s_2,(0,1,\not b)) = (s_2,(0,1,0),I)$$
$$\delta(s_2,(1,0,\not b)) = (s_2,(1,0,0),I) \qquad \delta(s_2,(1,1,\not b)) = (s_2,(1,1,1),I)$$
$$\delta(s_2,(\not b,\not b,\not b)) = (s_3,(\not b,\not b,1),N)$$

Máquinas de Turing

- **Máquina de Turing multicinta**

  - Esta MT tiene varias cintas, cada una de las cuales tiene su propia cabeza de lectura escritura que se controla de forma independiente.

  - En un solo movimiento, esta MT:

    1- Cambia de estado dependiendo del estado actual y del contenido de las celdas de todas las cintas, que están analizando las cabezas de lectura-escritura.

    2- Escribe un nuevo símbolo en cada una de las celdas barridas por sus cabezas de lectura-escritura.

    3- Mueve cada una de sus cabezas hacia la izquierda o hacia la derecha de forma independiente.

  - La función $\delta: Q x \Gamma^n \rightarrow Q x \Gamma^n x \{I,D,N\}^n$ donde una transición de la forma $\delta(s, (\gamma 1, \gamma 2,...,\gamma n)) = (p,(\varphi 1,\varphi 2,...,\varphi n), (X1,X2,...,Xn))$ significa que cambia del estado s al p, reemplaza γi por φi en la cinta i y mueve la cabeza de la cinta i en la dirección Xi.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejemplo: Construir una MT multicinta que reconozca las palabras de $L = \{a^n b^n \mid n \geq 1\}$

  - Supongamos que inicialmente colocamos la cadena a analizar en la cinta 1 y que s0 es el estado inicial.

  La funcion δ es la siguiente:

  δ(s0, (a, ƀ)) = (s0, (a, a), (D,D))
  δ(s0, (b, ƀ)) = (s1, (b, ƀ), (N, I))
  δ(s1, (b, a)) = (s1, (b, a), (D, I))
  δ(s1, (ƀ; ƀ)) = (s2, (ƀ, ƀ), (I,D))

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Máquinas de Turing Universales**

  - La MT Universal es una MT que, a partir de una descripción adecuada de una MT T y una cadena de entrada w, simula el comportamiento de T sobre la cadena w.

  - Por lo tanto, se requiere determinar una forma de describir una MT arbitraria T=(S,Σ,Γ,ƀ, s1,δ, F)

    1- Se requiere que T tenga un único estado de aceptación.

    2- Se supone que S={s1, s2,..., sn} donde s1 es el estado inicial y s2 el único estado final de T.

    3- Suponemos Γ={γ1,γ2,...,γn} donde γ1 es el blanco.

  - Partiendo de estas suposiciones, T estará completamente determinada por medio de su función de transición.

Máquinas de Turing

- Para codicar δ

  1- Se representara s1 por 1, s2 por 11 y así sucesivamente.

  2- De forma similar, se representará yi por una cadena de i unos.

  3- Finalmente representamos las directivas para la cabeza de lectura-escritura de forma que I sera 1 y D sera 11.

  4- Se utilizara al 0 como separador.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Dada una MT de la forma
  T=(S={s1,s2,...,sn},Σ, Γ= {γ1,γ2,...,γn},γ1, s1,δ, {s2})
  su codificación solo requiere representar las transiciones δ.
  - Ejemplo: La transición
  δ(s3,γ1) = (s4,γ3, I)
  se puede representar por 01110101111010111010.
  - De esto se deduce que T tiene una codificación representada por una cadena finita de ceros y unos.
  - Ademas, dada una codificación, podemos decodificarla correctamente.

Máquinas de Turing

- Implementación

- Una MT Universal T_u se puede implementar como una MT de tres cintas cuyo alfabeto de entrada contenga ceros y unos.

-La primera cinta contiene la codificación de T con su cabeza situada sobre el cero inicial de la cadena de ceros y unos.

- La segunda cinta contiene la codificación del contenido de la cinta de T con su cabeza situada sobre el uno que pertenece a la codificación del símbolo actual.

- La tercera cinta se usa para guardar el estado actual de T. Inicialmente contiene Ь1Ь.

Máquinas de Turing

- Funcionamiento

  - T_u analiza y compara el contenido de la segunda (cadena de entrada) y tercera cinta (estado actual) con el de la primera (funcion de transición codificada), hasta que encuentra una transición para la configuración codificada o hasta que agota todas las posibilidades.

  - Si no se encuentra una transición correspondiente a la configuración, T_u para, como debera hacerlo T. En otro caso, T_u se comporta como lo haría T.

  - La cadena final que quede en la segunda cinta de T_u será la codificación de la cadena que hubiera quedado en T.

Máquinas de Turing

- Ejemplo: Indicar una codificación completa de la MT universal que reconozca

  $L = \{a^n | n \geq 0\}$

  - La función de transicion es:

  $\delta(s1, a) = (s1, a, D)$

  $\delta(s1, ь) = (s2, ь, I)$

  - Observe que tiene un único estado final s2 y que la numeración de los estados es la adecuada.

  - La codificación será 0101101011011010101101010

Máquinas de Turing

- Ejercicio:

  Obtener una codificación completa de la MT universal dada por F={s2}

  δ(s0, ƀ) = (s0, ƀ,D)

  δ(s0, a) = (s1, ƀ, I)

  δ(s1, ƀ) = (s2, a, I)

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Autómatas linealmente acotados**

  - No es posible aumentar la capacidad de computación de una Máquina de Turing estandar aunque se complique la estructura de la cinta.

  - Se puede probar que es posible limitar su potencia si se restringe la manera en que se usa la cinta.

  - Una restricción posible es limitar el número de celdas de la cinta que pueden ser procesadas por la máquina de Turing, basandose en la longitud de la cadena de entrada.

  - Es decir, que habrá mas espacio disponible para la computación de las computaciones de cadenas largas que para las cortas.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Autómata Linealmente acotado (ALA): es una máquina de Turing no determinista T= (S,Σ, Γ,Ь, s1, δ, F) en la cual Γ contiene dos símbolos especiales < y >.**

La máquina comienza con la configuración (s1,<w>). No se permite que T reemplace los símbolos < y > ni que desplace su cabeza lectora-escritora a la izquierda de < o a la derecha de >.

El ALA tiene que realizar su computación en las celdas de la cinta ocupadas por la cadena de entrada.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejemplo: Construir un ALA sobre {a, b} que invierta los símbolos de la cadena de entrada.

Q = {qo, q1}, Σ= {a, b}, F = {q1}, Γ= {a, b,<,>}

La función δ esta definida como:

δ(q0,<) = (q0,<,D)

δ(q0, a) = (q0, b,D)

δ(q0, b) = (q0, a,D)

δ(q0,>) = (q1,>,N)

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Ejercicio: Obtener un ALA que determine si sus cadenas de entrada tienen longitud impar.

Suponer que las cadenas de entradas pertenecen a {a, b}*.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Lenguajes aceptados por una MT**

  - Una cadena w sobre algún alfabeto es aceptada por una MT T, cuando una computación que empezó con w en la cinta de T y con T en su estado inicial, termina con T en su estado final.

  - w sera rechazada si T no termina en un estado de aceptación o si T nunca para.

  - Los métodos de rechazo no son equivalentes

  - Analicemos dos MT distintas que resuelven un mismo problema.

Máquinas de Turing

- Ejemplo: Construir una MT que acepte el lenguaje a* sobre {a, b}

  - Caso 1

  Las cadenas no aceptadas hacen que la máquina pare en un estado no aceptador.

  Q = {q0, q1}, Σ= {a, b}, F = {q1}, Γ= {a, b, ƀ}

  δ(q0, a) = (q0, a,D)

  δ(q0, ƀ) = (q1, ƀ,D)

  (Esta MT para con todas las cadenas de entrada).

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Caso 2

Las cadenas no aceptadas hacen que la máquina entre en un bucle infinito.

Q = {q0, q1, q2}, Σ= {a, b}, F = {q2}, Γ= {a, b, ƀ}

δ(q0, a) = (q0, a,D)

δ(q0, b) = (q1, b,D)

δ(q0, ƀ) = (q2, ƀ,D)

δ(q1, a) = (q1, a,D)

δ(q1, b) = (q1, b,D)

δ(q1, ƀ) = (q1, ƀ,D)

(Esta MT NO para con todas las cadenas de entrada)

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Un lenguaje L sobre un alfabeto se dice *recursivamente enumerable* si es aceptado por alguna máquina de Turing. Más formalmente,

- **Un lenguaje L es recursivamente enumerable si para alguna máquina de Turing M ocurre que**

$L = \{w \in \Sigma^* | qw \vdash^* upv$ para $p \in F$ y $u, v \in \Sigma^*\}$

(donde q es el estado inicial de M y F es el conj.de estados finales de M).

- **Un lenguaje L es un lenguaje recursivo si L es recursivamente enumerable y hay alguna máquina de Turing que para sobre todas las entradas que acepta L.**

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Relación entre MT y lenguajes regulares

  - Veremos que es posible convertir un autómata finito determinista en una maquina de Turing.

  - Luego, puesto que todo lenguaje regular puede ser aceptado por un AFD, tenemos el siguiente teorema:

- **Teorema: Si L es un lenguaje regular entonces L es también un lenguaje recursivo.**

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

• Sea M = (Q,Σ, q, F, δ) un AFD. Se puede construir una MT M'= (Q',Σ', Γ,q, ƀ, F', δ') para la cual L(M) = L(M') por medio de :

Q'= Q ∪ {q'} donde q' es un nuevo estado que no esta en Q

Σ'=Σ

Γ= Σ ∪ {ƀ}

F'= {q'}

δ'(q, σ) = (δ(q, σ), σ,D) para q ∈ Q y σ ∈ Σ'

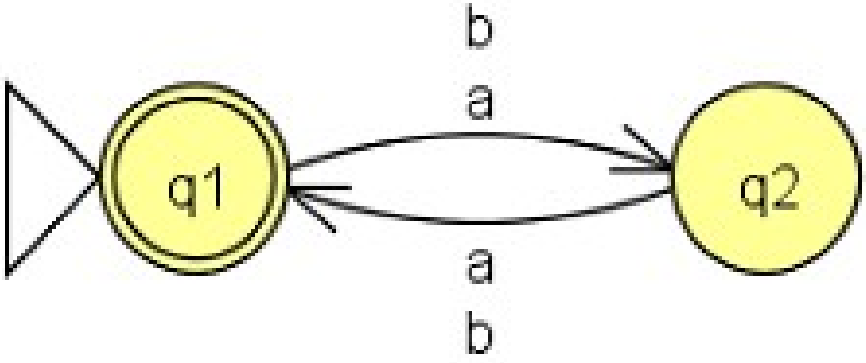
δ'(q, ƀ) = (q', ƀ,N) para todo q ∈ F

- La MT resultante cambia su estado con cada símbolo de entrada para reflejar el correspondiente cambio de estado en el AF. Cuando se encuentra un ƀ en el extremo derecho de la cadena pasa a un estado de aceptación solo si el AF acepta la cadena.

Máquinas de Turing

- Ejercicio: El autómata finito determinista M = (Q,Σ, s, F, δ) acepta las cadenas de longitud par sobre {a, b}*, donde

Q = {q1, q2}
Σ= {a, b}
s = q1
F = {q1}
δ(q1, a) = q2
δ(q1, b) = q2
δ(q2, a) = q1
δ(q2; b) = q1



- Construir una MT a partir de este autómata que acepte L(M)

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **MT y Lenguajes libres de contexto**

 - De manera similar veremos como construir una MT a partir de un Autómata a Pila.

 - Esto nos permitirá afirmar que todo lenguaje libre de contexto es un lenguaje recursivo.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- Sea M un APND. Podemos construir una MT que emule el comportamiento de M en lo que se refiere a la aceptación o rechazo de cadenas.

- Para esto hay que tener en cuenta los cambios de estado de M y los cambios que se producen en la pila de M cuando esta reconociendo una cadena.

- Por simplicidad, usaremos una MT con dos cintas. La cinta 1 almacenará la cadena de entrada y la cinta 2 el contenido de la pila.

-Un apilamiento corresponde a un movimiento a derecha seguido por una actualización de la cinta 2. Por lo tanto, la cabeza lectora-escritora de la cinta 2 siempre analizará el símbolo de la cima de la pila.

- Los cambios de estado de la MT reflejan los cambios de estado de M.

Máquinas de Turing

Consideremos el APND que acepta el LLC $\{a^n b^n \mid n \geq 0\}$ dado por

$S = \{s1,s2,s3\}$, $\Sigma = \{a,b\}$, $\Gamma = \{a,b,z\}$, $s = \{s1\}$, $F = \{s3\}$

$\delta(s1, \lambda, z) = \{(s3, \lambda)\}$ $\quad$ $\delta(s1, a, a) = \{(s1, aa)\}$

$\delta(s2, b, a) = \{(s2, \lambda)\}$ $\quad$ $\delta(s1, a, z) = \{(s1, az)\}$

$\delta(s1, b, a) = \{(s2, \lambda)\}$ $\quad$ $\delta(s2, \lambda, z) = \{(s3, \lambda)\}$

- La MT que construiremos empieza en el estado s'. La cabeza de lectura-escritura de la cinta 1 comienza sobre el símbolo del extremo izquierdo de la cadena de entrada y la cinta 2 esta vacía (todos los símbolos son blancos).

- Primero, marcaremos el fondo de la pila en la cinta 2 con las transiciones:

$\delta(s', (a, ƀ)) = ((s1, (a, z), (N,N))$

$\delta(s', (ƀ, ƀ)) = ((s3, (ƀ, z), (N,N))$

Máquinas de Turing

- Los dos casos en que el APND apila se ejecutan mediante un movimiento a la derecha de la cabeza de la cinta 2 y la escritura de un nuevo caracter sobre ella. En este punto, el caracter de entrada es consumido mediante el movimiento de la cabeza de la cinta 1 a la derecha. Las transiciones son:

$\delta(s1, (a, z)) = (p1, (a, z), (N,D))$

$\delta(s1, (a, a)) = (p1, (a, a), (N,D))$

$\delta(p1, (a, Ь)) = (s1, (a, a), (D,N))$

- Cuando se desapila, la cabeza de la cinta 2 se mueve hacia la izquierda:

$\delta(s1,(b, a)) = (s2, (b, a), (D, I))$

$\delta(s2, (b, a)) = (s2, (b, a), (D, I))$

Máquinas de Turing

- Finalmente la cadena se acepta cuando ha sido totalmente consumida y se ha vaciado la pila. La transición del APND

$\delta(s2, \lambda, z) = \{(s3, \lambda)\}$
corresponde al siguiente movimiento de la MT

$\delta(s2, (ƀ, z)) = (s3, (ƀ, z), (N, I))$

- Puesto que todo lenguaje libre de contexto puede ser aceptado por un AP, tenemos el siguiente teorema

Teorema: Si L es un lenguaje libre de contexto, entonces L es también un lenguaje recursivo.

Máquinas de Turing

- Los lenguajes recursivos son cerrados bajo unión, intersección y complemento.

- **Teorema: Si L1 y L2 son lenguajes recursivos, entonces L1 ∩ L2 también lo es.**

- Para probarlo, supongamos que M1 y M2 son dos MT que paran sobre toda cadena. Entonces, si $w \in L(M_i)$, sabemos que M_i parará en un estado de aceptación, y si $w \notin L(M_i)$, M_i parará en un estado que no es de aceptación para cada i = 1, 2. La operación de intersección puede ser realizada por una MT de dos cintas (la cinta 1 para emular a M1 y la 2 para M2).Por lo tanto, L(M) = L(M1) ∩ L(M2). M se detiene para toda entrada, ya que M1 y M2 también lo hacen. Es decir, que L(M) es un lenguaje recursivo.

- Analizar las operaciones restantes.

Máquinas de Turing

- **Lema: Si L es un lenguaje recursivo, entonces Σ*-L también lo es.**

- Para probarlo, supongamos que L es un leng. recursivo sobre el alfabeto Σ. Entonces hay una MT M = (S,Σ,Γ, s, ƀ, F,δ) que para sobre toda entrada y acepta L.

- Consideremos la MT M'= (S,Σ,Γ, s, ƀ, S-F,δ). Es claro que cualquier cadena de L hace que M' pare en algún estado de F, por lo que M' rechaza todas las cadenas de L.

- Por otro lado, si w ∉ L, M parará en algún estado que no es de F. Es decir, M para en algún estado de S-F. Pero entonces, M' también parará, y por lo tanto, si w ∉ L , entonces w ∈ L(M').

- Así, L(M') = Σ*-L. Es más, ya que M para sobre todas las cadenas, M' también lo hace.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- La propiedad que vimos anteriormente de los lenguajes recursivos, en general no se cumple para los lenguajes recursivamente enumerables.
- **Teorema: Hay un lenguaje recursivamente enumerable L para el cual Σ*-L no es recursivamente enumerable.**

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Algoritmos**

  - Las MT pueden considerarse como modelos de computación mecánica.

  - Una MT que realiza una tarea modela un proceso.

  - Los procesos que siempre terminan se llaman algoritmos, y por lo tanto una MT que para sobre cualquier cadena es un modelo de algoritmo.

  - Esto significa que, para los lenguajes recursivos, hay un algoritmo que determina si una cadena está en un lenguaje L.

  - Este algoritmo será modelado por una MT que para sobre cualquier entrada y acepta L.

Máquinas de Turing

- **Problemas irresolubles**

- Por otro lado, si L es recursivamente enumerable, pero no recursivo, no hay una MT que pare sobre todas las entradas y acepte L.

- De esto se deduce que no hay ningún modelo de algoritmo que determine si $w \in L$, para una cadena arbitraria w.

- El problema de la pertenencia (¿esta w en L?), para los lenguajes recursivamente enumerables que no son recursivos, es un ejemplo del problema de irresolubilidad, que es un problema para el cual no hay un algoritmo que pueda dar respuesta.

Máquinas de Turing

- **Gramática no restringida: Una gramática no restringida (o estructurada por frases) es una 4-tupla**
  G = (N,Σ,S,P) donde
  N es un alfabeto de símbolos no terminales.
  Σ es un alfabeto de símbolos terminales con N∩Σ= ∅.
  S ∈ N es el símbolo inicial.
  P es un conjunto finito de producciones de la forma α→β donde α ∈ (N ∪ Σ)$^+$ y β ∈ (N ∪ Σ)*

- Cualquier GR o GLC es una gramática no restringida. Las gramáticas no restringidas tienen una potencia generativa mayor, debido a que hay menos restricciones con respecto a la formación de las producciones.

Máquinas de Turing

- Ejemplo: Esta gramática no restringida genera cadenas de la forma: $a^n b^n c^n$

$S \rightarrow aSBC|aBC$ $\quad CB \rightarrow BC$

$aB \rightarrow ab$ $\quad bB \rightarrow bb$

$bC \rightarrow bc$ $\quad cC \rightarrow cc$

- Si n = 1 esta gramática puede derivar abc por medio de

$S \Rightarrow aBC \Rightarrow abC \Rightarrow abc$

- Si n>1 entonces la cadena $a^n b^n c^n$ se deriva por medio de

$$S \underset{\Rightarrow}{n-1} a^{n-1} S (BC)^{n-1} \Rightarrow a^n (BC)^n \overset{*}{\underset{\Rightarrow}{}} a^n B^n C^n \underset{\Rightarrow}{n} a^n b^n C^n \underset{\Rightarrow}{n} a^n b^n c^n$$

- Vemos que $L(G) = \{a^n b^n c^n \mid n \geq 1\}$. Es decir, con una gramática no restringida podemos generar lenguajes que no son LLC.

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Gramáticas no restringidas y lenguajes**

-Teorema: Si G es una gramática no restringida, entonces L(G) es un lenguaje recursivamente enumerable.

- También se cumple que todo lenguaje recursivamente enumerable es generado por una gramática no restringida

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Gramáticas Sensibles al contexto**

- Entre las gramáticas no restringidas y la forma de las gramáticas independientes del contexto se pueden definir varias gramáticas con diferentes niveles de restricción.

- No todas producen una clase de lenguajes interesante, pero unas que si lo hacen son las que forman el conjunto de gramáticas sensibles al contexto.

- **Denifición: Una gramática G = (N, Σ, S, P) es sensible al contexto si todas las producciones son de la forma $\alpha \rightarrow \beta$ donde $\alpha, \beta \in (N \cup \Sigma)^+$ y $|\alpha| \leq |\beta|$**

Máquinas de Turing

- **Ejemplo: Se busca definir una GSC que genere L = $\{a^n b^n c^n \mid n \geq 1\}$**

G = (Vn,Vt,X_0, P)

Vn = {X_0,X_1,X_2} donde X_1 y X_2 son símbolos viajeros

Vt = {a, b, c}

P = { $X_0 \rightarrow abc$ (1)

$X_0 \rightarrow aX_1bc$, (2)

$X_1b \rightarrow bX_1$, (3)

$X_1c \rightarrow X_2bcc$, (4)

$bX_2 \rightarrow X_2b$, (5)

$aX_2 \rightarrow aaX_1$, (6)

$aX_2 \rightarrow aa\}$ (7)

- Analice una derivación D de la palabra $a^iX_1b^ic^i$, con i≥1

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

P = { $X_0 \rightarrow abc$ (1) $X_0 \rightarrow aX_1bc$, (2)

$X_1b \rightarrow bX_1$, (3) $X_1c \rightarrow X_2bcc$, (4)

$bX_2 \rightarrow X_2b$, (5) $aX_2 \rightarrow aaX_1$, (6)

$aX_2 \rightarrow aa$} (7)

- Analice una derivación D de la palabra $a^iX1b^ic^i$, con $i \geq 1$

$$a^i X_1 b^i c^i \underset{\Rightarrow}{i\ veces(3)} a^i b^i X_1 c^i \underset{\Rightarrow}{(4)} a^i b^i X_2 bc^{i+1}$$

- Ahora la única posibilidad es

$$a^i b^i X_2 bc^{i+1} \underset{\Rightarrow}{i\ veces(5)} a^i X_2 b^{i+1} c^{i+1}$$

- Esto permite derivar directamente $a^{(i+1)}X_1b^{(i+1)}c^{(i+1)}$ o $a^{(i+1)}b^{(i+1)}c^{(i+1)}$

Máquinas de Turing

- Ejemplo: Indique la manera de generar la cadena "aabbcc"

$P = \{ X_0 \rightarrow abc \quad (1)$

$X_0 \rightarrow aX_1bc, \quad (2)$

$X_1b \rightarrow bX_1, \quad (3)$

$X_1c \rightarrow X_2bcc, \quad (4)$

$bX_2 \rightarrow X_2b, \quad (5)$

$aX_2 \rightarrow aaX_1, \quad (6)$

$aX_2 \rightarrow aa\} \quad (7)$

$$X_0 \underset{\Rightarrow}{(2)} aX_1bc \underset{\Rightarrow}{(3)} abX_1c \underset{\Rightarrow}{(4)} abX_2bcc \underset{\Rightarrow}{(5)} aX_2bbcc \underset{\Rightarrow}{(7)} aabbcc$$

UNTDF
Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur

Máquinas de Turing

- **Gramáticas Sensibles al contexto:**

- Lema: El conjunto de los lenguajes sensibles al contexto contiene el conjunto de los lenguajes independientes del contexto.

- Lema: Sea G = (N, Σ, S, P) una gramática sensible al contexto. Entonces existe una máquina de Turing T que para sobre toda entrada y acepta L(G).

- Teorema: Si L es un lenguaje sensible al contexto, entonces L es recursivo.

- Lema: Hay lenguajes recursivos que no son sensibles al contexto. Es decir, la inversa del teorema anterior no se cumple.