1-La Máquina de Turing.

Turing trato de modelar una computadora humana:

- Se escriben símbolos de un conjunto finito.
- Cada acción depende del símbolo que es examinado y del "estado mental" del momento.
- Los estados mentales pueden cambiar pero son finitos

Una **Máquina de Turing** es un dispositivo dotado de una cinta de memoria ilimitada, formada por celdas donde se pueden leer y escribir símbolos. En cualquier instante sólo hay un número finito de celdas en uso, quedando las demás en blanco. En dicha cinta existe un cabezal que puede estar en diferentes estados (representan los estados mentales). Es decir, está compuesta por:

- **Cinta infinita** en los dos extremos, dividida en celdas que almacenan símbolos de un alfabeto.
- **Unidad de control**: almacena el estado de la máquina.
- **Cabezal**: apunta a una celda y solo se puede mover en un paso, a derecha o izquierda. El símbolo apuntado se llama símbolo corriente.

Funcionamiento:

- En cada paso, el cabezal lee la posición en la que se encuentra en la cinta, y dependiendo del estado, escribe; posteriormente avanza o retrocede a la celda adyacente.
- Puede cambiar de estado.
- Repetimos hasta que no tenga definida una tarea

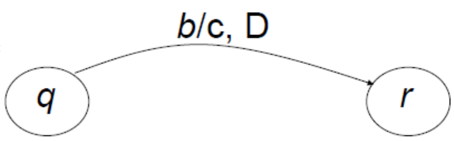
Diferentes visiones:

- Una máquina de Turing resuelve problemas calculando una solución.
- Se puede ver como una máquina que sólo puede resolver problemas de decisión, produciendo la respuesta sí o no ante una instancia. (**MT reconocedora**). Se trata de reconocer el lenguaje de las cadenas de símbolos que representan las instancias positivas del problema
- Resuelve el problema generando todas las instancias positivas del problema.

Caracterización formal:

Un MT es una séptupla (Q, A, B, δ, q0, #, F):

- Q: Conjunto finito de estados.
- A: alfabeto de entrada
- B: alfabeto de símbolos de la cinta (incluye a A).
- δ: función de transición que asigna a cada estado q ∈ Q y símbolo b ∈ B, el valor δ(q, b) que puede ser vacío (no definido) o una tripleta (r, c, M) donde r ∈ Q, c ∈ B, M ∈ {I (izd), D(der)}. Notación gráfica:



- q0: estado inicial.
- #: símbolo blanco de B\A.
- F: estados finales (está contenido en Q).

Existen representaciones alternativas: cintas limitadas a izquierda, posibilidad de no mover el cabezal en una transición…

Configuración (tripleta $(q, w1, w2)$):

- q: estado en el que se encuentra la máquina.
- w1: representación de la parte de la palabra que hay a la izquierda del cabezal.(puede ser vacío)
- w2: representación de la parte de la palabra que se obtiene empezando en el cabezal hacia la derecha. (no puede ser vacío)

Ejemplo:

Si $\delta(q, a) = (p, b, D)$, el paso de cálculo sería:

$(q, c1 \ldots cn , ad2 \ldots dm) \vdash (p, c1 \ldots cn\ b, d2d3 \ldots dm)$

Si $\delta(q, a) = (p, b, I)$, el paso de cálculo sería:

$(q, c1 \ldots cn , ad2 \ldots dm) \vdash (p, c1 \ldots cn{-}1, cn\ b\ d2d3 \ldots dm)$

Desde una configuración R se llega a otra R' en una **sucesión de pasos de cálculo** $(R \vdash^* R')$ $\leftrightarrow$ existe una sucesión finita de configuraciones Ri tales que $R=R1$, $R'=Rn$ y $Ri \vdash Ri+1$, $i=1,\ldots,n$

Una **MT para** cuando en el estado actual y símbolo de la cinta no hay ninguna transición definida o se alcanza un estado final.

El **lenguaje aceptado por una MT** M es el subconjunto L(M) de A^* tal que $u \in L(M) \leftrightarrow$ existen $w1,w2 \in B^*$ y $q \in F$ tales que

$(q0, \varepsilon, u) \vdash^* (q, w1, w2)$

2-Lenguajes Recursivamente Enumerables.

L $\subseteq$ A* se dice **LRE** $\leftrightarrow$ existe una MT M tal que L(M)=L, es decir, si existe una MT M=(Q, A, B, δ, q0, #, F) que alcanza un estado final desde (q0, ε, u) mediante una sucesión de pasos de cálculo para toda u$\in$L.

Una **MT para** cuando en el estado actual y símbolo de la cinta no hay ninguna transición definida o se alcanza un estado final.

Una palabra u$\in$A* es aceptada por M=(Q, A, B, δ, q0, #, F) cuando M para a partir de la configuración (q0, ε, u).

3-Lenguajes Recursivos.

L $\subseteq$ A* se dice **LR** $\leftrightarrow$ existe una MT M tal que L(M)=L y para sobre cualquier entrada de A*.

Un LR es siempre LRE.

Hay dos tipos de estados finales: de aceptación y de rechazo.

Un problema de decisión cuyo lenguaje es recursivo se dice decidible. En otro caso, el problema es indecidible.
Indecidible: no existe un algoritmo que tome como entrada una instancia de un problema y determine si la respuesta es SÍ o NO.

(Añadir tabla de propiedades de cierre)

4-Máquinas de Turing Calculadoras.

Sea $D \subseteq A^*$, $H \subseteq B\setminus\{\#\}$ y $f : D \rightarrow H^*$ una función.
La función **f se dice calculable parcial** si existe una MT,
$M = (Q, A, B, \delta, q0, \#, F)$ tal que si $u \in D$ entonces desde la configuración inicial $(q0, \varepsilon, u)$ se llega a una configuración $(q, u1, u2)$ en la que $u1u2 = f(u)$, $q \in F$ y la máquina para. Es decir, llegamos a una configuración en la que estamos en un estado final, la máquina termina y el contenido de la cinta es f(u)
Calculable total: si la función es calculable parcial y $D=A^*$.
Ejemplo: una MT que calcula m-n (en notación unaria)

5-Técnicas de construcción de MT .

Una MT puede irse ampliando hasta conseguir la MT Universal capaz de procesar otras MT.

Memoria adicional.

A veces queremos que se recuerde un símbolo en cualquier estado, por lo que el conjunto de estados será el conjunto de las parejas QxB donde $q \in Q$ es un estado y $b \in B$ es un símbolo.

Pistas múltiples.

Disponer de la MT con una cinta con varias pintas:

- Disponemos de varias casillas en cada posición donde poder escribir un símbolo.
- Existe un único cabezal.
- Dos pistas equivale a suponer que el alfabeto de trabajo (B) $= B \times B$ y tener k cintas a suponer que es B**k .

Subrutinas.

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

- En dicho conjunto, hay un estado inicial y otro que sirve como estado de retorno.
- La llamada a una subrutina se produce cuando existe una transición a un estado inicial.
- 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 a posicionarse.
- El estado se puede determinar haciendo varias copias del último estado de la subrutina, una para cada estado al que haya que volver.
- Si hay que llamar a una subrutina desde varios estados se hace una copia de la subrutina tantas veces como haga falta.

Ejemplo: multiplicación de números en base 1.

6-Variaciones de la MT .

Movimientos estáticos:

MT que se puede quedar en el mismo sitio.

δ(q,b) puede ser vacío o una tripleta (p, c, M) con p∈Q, c∈B y M∈{I,D,S} indicando S que el cabezal no se mueve.
No supone ninguna potencia adicional, pues se puede similar con una MT tradicional:
Supón que quieres δ(q,b) = (p, c, S). Hacemos
δ(q,b)= (rp, d, D) y desde todos los estados rp solo podemos hacer δ(rp,d)= (p, d, I)

Máquina multicinta:

Tiene k cintas ilimitadas y una cabeza de lectura para cada una.

¡No es lo mismo múltiples cintas que múltiples pistas!:

- Múltiples pistas tan solo nos hace visualizar la cinta de una MT tradicional en la que B=BxB (cada símbolo está formado por una n-upla de símbolos básicos)
- Múltiples cintas si que deja de ser una MT tradicional. El cabezal puede estar en una posición distinta en cada cinta. Hay que especificar qué movimiento Mi hay que realizar en cada cinta i, además de lo que se ve en cada cinta, bi, y lo que se escribe en cada una, ci.
  δ puede asignar a cada (q,b1,…,bk) un vector (p, c1,..,ck, M1,…,Mk) con q,p∈Q, bi,ci∈B y Mi∈{I,D,S}.

CONFIGURACIÓN:
Vector (q, u1, w1, u2, w2, . . . , uk , wk) donde
q: Estado en el que está la MT.
ui: es la parte de la palabra que hay a la izquierda del cabezal de lectura de la cinta i.
wi: la parte de la palabra que hay en la cinta i desde el cabezal de lectura de esa cinta hacia la derecha.

LENGUAJE ACEPTADO:
Conjunto de palabras u del lenguaje de entrada 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 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:
Función f es calculada por una MT si para cada palabra u para la que está definida f(u) se cumple:
Que la MT comience con u en la primera cinta y el resto de las cintas vacías, el cabezal de la primera cinta en el primer símbolo de u y se llegue a una configuración en la que estamos en un estado final y el contenido de la última cinta es f(u).

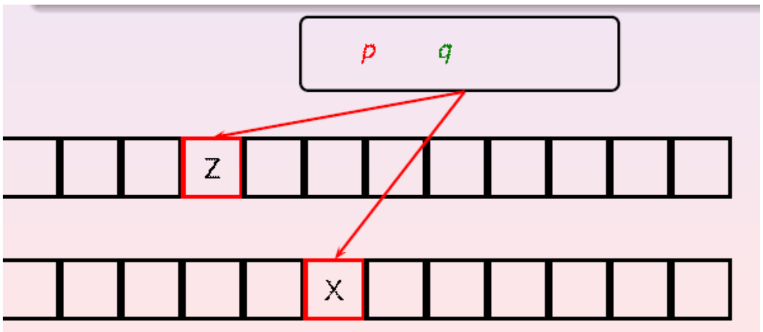
VARIAS CINTAS vs UNA CINTA:
Todo lenguaje aceptado por una MT con varias cintas es RE.
Si M es una MT con k cintas, su funcionamiento se puede simular con una MT, N, con una sola cinta.

COMPLEJIDAD TEMPORAL.

Tiempo de ejecución de la MT M para una entra u: número de pasos que M realiza antes de pararse. Si no para, se dice que el tiempo de ejecución es infinito.

Complejidad Temporal de la MT M: función T(n) que da para cada n el tiempo máximo de ejecución de todas las entradas de longitud n. Nos interesan las MT que siempre paran.

El tiempo empleado por la MT N en simular n pasos de la MT M de k cintas es $O(n^2)$



Máquinas no deteministas:

Desde una configuración hay distintas transiciones posibles.

LENGUAJE ACEPTADO:

Conjunto de todas las palabras aceptadas.

Se 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 si hay otros pasos de cálculo posibles que no llegan a un estado de aceptación.

EQUIVALENCIA MTND Y MT.

Toda MT es una MTND.

Si un lenguaje es aceptado por una MTND → es RE.

Máquinas con cintas semiilimitadas:

La cinta es ilimitada sólo a la derecha, en la izquierda tiene un tope.

Tampoco existe el símbolo espacio en blanco #.

Supongamos MT M1 que nunca escribe # y cuyo cabezal nunca se mueve hacia la izquierda → Todo lenguaje aceptado por una MT es también aceptado por M1.

Supondremos tenemos el símbolo ⊲ 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.

7-Calculabilidad .

Codificación de cadenas y de MTs.

CODIFICACIÓN DE CADENAS:

- Podemos establecer una biyección entre las palabras de un

alfabeto A={a1,…,an} y los números naturales:

$Z(\varepsilon)=0$, $Z(ai)=i$

Para w=aik,…,ai1, $z(w)= Z(aik).n^{k-1}+ Z(aik-1).n^{k-2}+....+Z(ai1).n^{0}$

Ejemplo

A = {0, 1, 2},

Z (Ɛ) = 0, Z (0) = 1, Z (1) = 2, Z (2) = 3,

$Z (202) = Z(2).n^2+ Z(0).n^1+Z(2).n^0 = 3 \times 3^2 +1 \times 3 +3$

Dado un número natural m, se puede encontrar una cadena C(m) en el alfabeto A cuya codificación sea m.

- Podemos establecer una biyección entre alfabetos A y B:

Llamo ZB a la aplicación de las palabras de B a los naturales.

Llamo ZA a la aplicación de los naturales a A*.

ZAoZB es la biyección de B* a A*.

Toda función calculable sobre un alfabeto es equivalente a otra función calculable sobre otro alfabeto. Basta con trabajar sobre un solo alfabeto (usualmente el binario).

CODIFICACIÓN DE MTs.

A cada MT podemos asignarle una cadena y un natural sobre el alfabeto binario. Veámoslo:

Sean los estados de la MT {q1(inicial), q2(final), q3,…, qk}. Sea B={a1=0, a2=1, a3=#, a4,…, am}. Al movimiento izquierda le asigno un 1 y al derecha un 2. Este número será u(M).

δ(qi, aj)= (qk, al, M) se codifica como $0^i\ 10^j\ 10^k\ 10^l\ 10^{u(M)}$ y todas las codificaciones se van separando por 11. Da lugar a una cadena que llamo R(M)=w. Calculamos su número Z(w)=Z(M) con el procedimiento para asignar números a palabras.

(M,u) con M MT y u palabra del alfabeto {0,1} se codifica como:

Ponemos M como hemos visto, después 111 y después la codificación de u. A esta cadena le llamo R(M,u). A esta cadena le puede asignar un número Z(M,u).

Cada número natural n corresponderá a una MT o a una cadena sin sentido. Sea T(n) dicha máquina (o nula). Llamo T(w) a la MT cuyo código es w.

{0,1}* es numerable y se puede ordenar. Se puede hablar de la i-esima MT, Mi=wi, y de la j-esima palabra wj (siendo ambas cadenas de {0,1}*). Se puede hablar de la máquina wi como procesadora de la palabra wj (diremos wi procesa wj). Si wi no corresponde a una MT, la asociaremos con una MT de un solo estado y ninguna transición.

	W1	W2	W3	..	..
W1≡M1	1	0	0	..	..
W2≡M2	0	1	1	..	..
W3≡M3	0	0	1	..	..
..	..	..	..	..	..
..	..	..	..	..	..

T_{ij}=1 sii wi acepta wj

T_{ij}=0 en otro caso

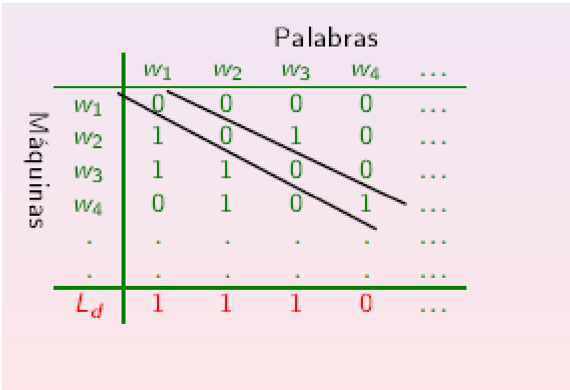
El lenguaje de diagonalización.

Vamos a construir un Ld sobre {0,1} que no es RE.

Uso la idea de diagonalización de Cantor.

$w \in \{0,1\}^*$, $w \in Ld \leftrightarrow$ la MT cuya codificación es w no acepta w.

Ld se construye a partir de la diagonal y cambiando 0/1 por 1/0.



TEOREMA: Ld no es RE.

Por reducción al absurdo sea MT aceptando Ld.

Dicha MT estará colocada en la lista de MT en algún lugar, supongamos i-ésimo lugar, Mi=wi.

Si $wi \in Ld \rightarrow$ Mi acepta wi, pero por la definición de Ld, wi no pertenece a Ld.

Si wi no pertenece a Ld $\rightarrow$ Mi no acepta wi, pero por la definición de Ld, $wi \in Ld$

El complemento de Ld es LRE, pero no LR.

El lenguaje universal.

El lenguaje universal, Lu, es el conjunto de todas las cadenas del alfabeto $v \in \{0,1\}^*$ que codifican parejas (M,w) (es decir, v=R(M,w)) siendo M una MT con alfabeto de entrada binario, w definida sobre el mismo alfabeto y tales que M acepta w.

Es decir, Lu está formado por el todas las cadenas que representan una MT y una entrada aceptada por dicha MT.

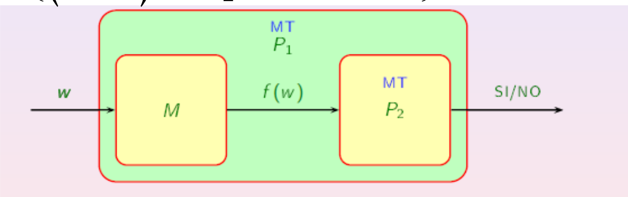
Existe una MT MU (MT universal) tal que Lu=L(MU)

TEOREMA: Lu es LRE, no LR. (demo diapo 88-93)

Problemas indecidibles.

TEOREMA: El problema de la parada es indecidible.

El lenguaje asociado Lh={(M,w)/ M para con w} es LRE, no LR.



REDUCCIONES.

Problema **P1 se reduce a P2** si existe una MT M que siempre para y calcula una función f tal que para toda entrara w a P1, tenemos que P2 produce la misma respuesta para la entrada f(w).

L1 se reduce a L2 → si L1 es indecidible, también lo es L2

→ si L1 no es RE, tampoco lo es L2.

MTs QUE ACEPTAN EL LENGUAJE VACÍO.

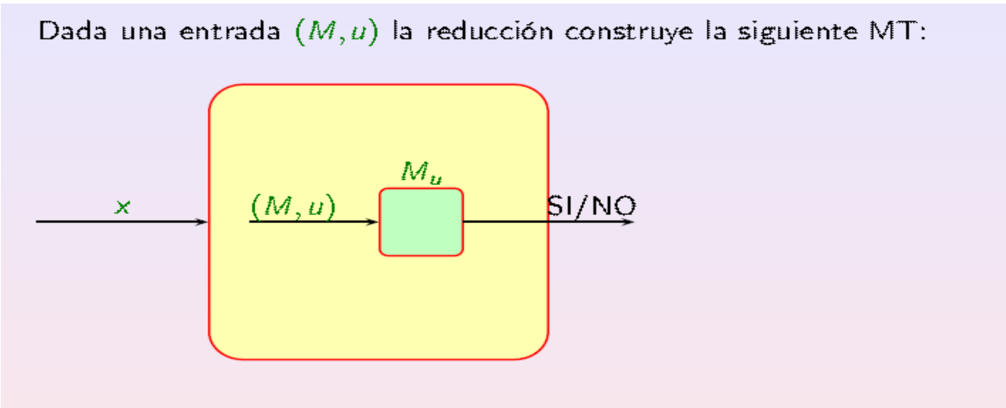
Defino los lenguajes sobre el alfabeto binario:

Le: palabras w tales que a MT M cuya codificación es w no acepta ninguna palabra.

Lne: palabras w tales que la MT M cuya codificación es w acepta alguna palabra. Podemos interpretar las cadenas binarias como la MT a la que codifica. →

Le={M/L(M) = Ø} y Lne={M/L(M) ≠ Ø} (son complementarios)

Lne es LRE, no LR.



Teorema de Rice.

PROPIEDADES DE LRE.

Propiedad LRE ↔ Propiedad de los lenguajes de las MTs.

Una propiedad P de los LRE es un subconjunto de los LRE (P⊂RE).

Un lenguaje L tiene la propiedad P si L∈P. (por ejemplo, la propiedad de estar formado por 0`s y 1`s)

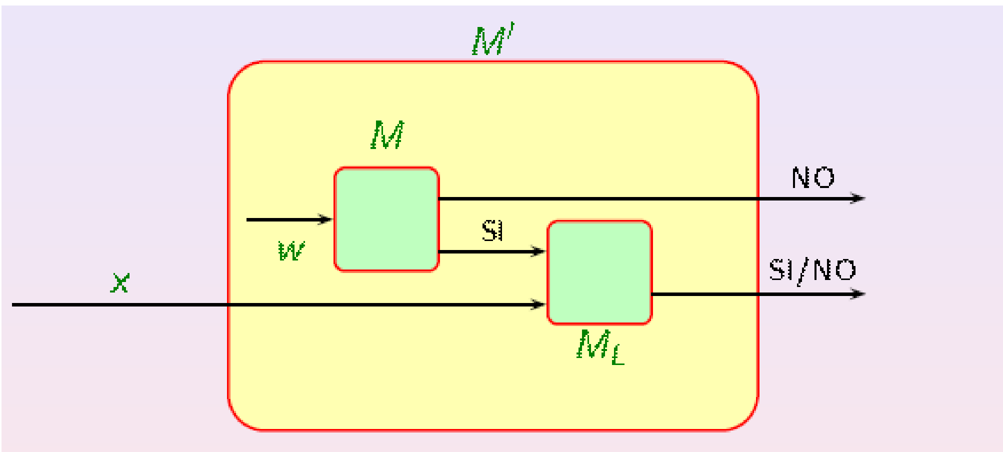
Una **propiedad es trivial** si es un conjunto vacío o el conjunto de todos los LRE, en caso contrario se dice que la propiedad es no trivial.

Una **propiedad se dice trivial** si ningún lenguaje aceptado por alguna MT no verifica la propiedad o si cualquier lenguaje aceptado por alguna MT verifica la propiedad.

TEOREMA DE RICE.

Toda propiedad no trivial sobre los LRE es indecidible.

Demo:



CONSECUENCIAS DEL TEOREMA:

- Todos los problemas que afectan al lenguaje de una MT son

indecidibles, por ejemplo son indecidibles:

Si L(M) es vacio, si L(M) es regular, si L(M) es independiente del contexto, siendo M una MT.

- No indica que cualquier problema relativo a MT's sea

indecidible, por ejemplo podría ser decidibles:

Preguntas acerca de los estados o de los movimientos de una MT.