9. Máquinas de Turing.

Introducción.

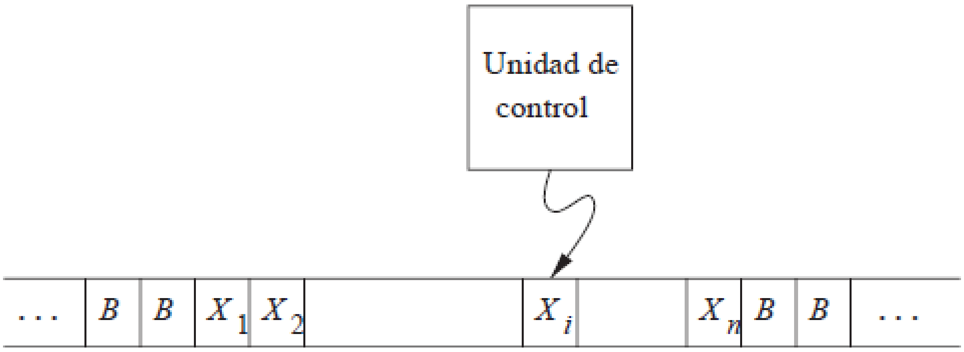


Figura 8.8. Una máquina de Turing.

Una Máquina de Turing consta de una ***unidad de control*** que puede encontrarse en un estado cualquiera de un conjunto finito de estados.

Tiene una ***cinta*** dividida en cuadrados o casillas y cada ***casilla*** puede contener un símbolo de entre un conjunto finito de símbolos.

Inicialmente, la ***entrada***, que es una cadena de símbolos de longitud finita que pertenecen a un ***alfabeto de entrada***, se colocan en la cinta. Las restantes casillas de la cinta (que se extiende infinitamente hacia la izquierda y hacia la derecha) almacenan un símbolo denominado ***espacio en blanco***.

El espacio en blanco es un símbolo de cinta, pero no un símbolo de entrada.

Existe una ***cabeza de cinta*** que a modo de ***cursor*** está situado en una de las casillas de la cinta. Se dice que la Máquina de Turing ***apunta*** a dicha casilla. Inicialmente, la cabeza de la cinta está en la casilla más a la izquierda que contiene la entrada.

Un ***movimiento*** de la Máquina de Turing depende del estado de la unidad de control y del símbolo de cinta al que señala la cabeza.

En un movimiento, la Máquina de Turing:

1. **Cambia de estado**. El siguiente estado podría ser el mismo que el estado actual.
2. **Escribe un símbolo de cinta** en la casilla que señala la cabeza, reemplazando el que estuviera anteriormente. Opcionalmente, podría ser el mismo que ya se encontraba allí.
3. **Mueve la cabeza de la cinta hacia la izquierda o hacia la derecha.**

Definición General.

Es una Tupla $MT = < Q, \Sigma, \Gamma, \delta, q_0, B, F >$ donde:
- Q , conjunto finito de *estados*
- Σ , alfabeto de *entrada* ($\Sigma \subseteq \Gamma$)
- Γ, alfabeto de la *cinta*
- δ , función de transición $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$
- $q_o \in Q$, *estado inicial*
- B es un símbolo que representa el *espacio en blanco*. ($B \in \Gamma, B \notin \Sigma$)
- $F \subseteq Q$, conjunto de *estados finales* o de *aceptación*.

Función de transición

La función de transición se define $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$

$$\delta(q, X) = (p, Y, Sentido)$$

Los argumentos de son un estado q y un símbolo de cinta X.

El resultado, si está definido, tiene:

- p es el estado al que pasa la máquina.
- Y es el símbolo de Γ, que se escribe en la cinta.
- $Sentido$ es un sentido en el que la cinta se mueve: izquierda (I) o derecha (D).

Es decir: a cada par (estado, símbolo de la cinta) le corresponde uno y sólo un estado, se escribe un símbolo en la cinta, y el cabezal se mueve en uno de dos sentidos: a izquierda (I) o a derecha (D).

La función puede estar indefinida para algunos argumentos.

Configuración instantánea

La ***configuración instantánea*** es una descripción de la Máquina de Turing en un momento dado.

Se denota con $\alpha_1 q \alpha_2$, donde q es el estado actual de la máquina de Turing y $\alpha_1 \in \Gamma^*$ y $\alpha_2 \in \Gamma^*$

$\alpha_1$es el contenido de la cinta a la izquierda del cursor (sin incluir la celda sobre el cursor)

α_2 es el contenido de la cinta a la derecha del cursor (incluyendo la celda sobre el cursor)

Secuencia de configuraciones:

El símbolo $\mapsto_M$ o directamente $\mapsto$ indica el movimiento válido entre dos configuraciones.

Movimiento a izquierda

Suponiendo que $\delta(q, X_i) = (p, Y, I)$:

$$X_1 X_2 \dots X_{i-1} \boldsymbol{q} X_i X_{i+1} \dots X_n \mapsto_M X_1 X_2 \dots X_{i-2} \boldsymbol{p} X_{i-1} \boldsymbol{Y} X_{i+1} \dots X_n$$

La cabeza ahora apunta a la casilla $i - 1$

Excepciones:

1. Si $i = 1$, entonces M se mueve al espacio en blanco que se encuentra a la izquierda de X_1. En ese caso:
$$\boldsymbol{q} X_1 X_2 \dots X_n \mapsto_M \boldsymbol{pBY} X_2 \dots X_n$$
2. Si $i = n$ e $Y = B$, entonces el símbolo B, escrito sobre X_n se añada a la secuencia infinita de los espacios en blanco que hay después de la cadena de entrada y no aparecerá en la siguiente configuración.
$$X_1 X_2 \dots X_{n-1} q X_n \mapsto X_1 X_2 \dots X_{n-2} \boldsymbol{p} X_n$$

Movimiento a derecha

Suponiendo que $\delta(q, X_i) = (p, Y, D)$: (movimiento a derecha)

$$X_1 X_2 \dots X_{i-1} \boldsymbol{q} X_i X_{i+1} \dots X_n \mapsto_M X_1 X_2 \dots X_{i-1} \boldsymbol{Y} \boldsymbol{p} X_{i+1} \dots X_n$$

La cabeza ahora apunta a la casilla $i + 1$

1. Si $i = n$, entonces la casilla $i + 1$ almacena un espacio en blanco, por lo que dicha casilla no formaba parte de la configuración anterior.
$$X_1 X_2 \dots X_{n-1} \boldsymbol{q} X_n \mapsto_M X_1 X_2 \dots X_{n-1} \boldsymbol{Y} \boldsymbol{p} B$$
2. Si $i = 1$ e $Y = B$, entonces el símbolo B escrito X_1 se añade a la secuencia infinita de los espacios en blanco anteriores a la cadena de entrada y no aparecerá en la siguiente configuración.
$$\boldsymbol{q} X_1 X_2 \dots X_n \mapsto \boldsymbol{p} X_2 \dots X_{n-1}$$

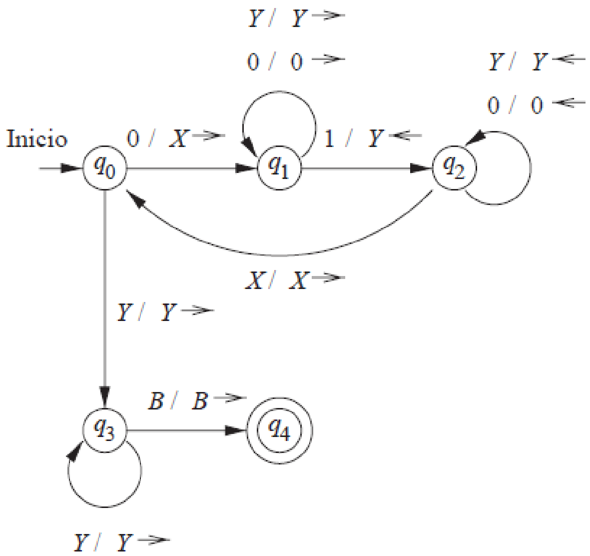
Lenguaje aceptado por una Máquina de Turing

El lenguaje de una Máquina de Turing es: $L = \{\omega \in \Sigma^* \wedge q_0\omega \mapsto^* \alpha p\beta | p \in F\}$

Ejemplo: $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1, X, Y, B\}, \delta, q_0, B, \{q_4\})$

Cuya función de transición se puede ver en la tabla o en el diagrama:

	Símbolo				
Estado	0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–



La Máquina de Turing acepta $L = \{0^n1^n | n > 0\}$
La cadena **0011** es aceptada:

$$q_00011 \mapsto Xq_1011 \mapsto X0q_111 \mapsto Xq_20Y1 \mapsto q_2X0Y1 \mapsto Xq_00Y1 \mapsto XXq_1Y1 \mapsto XXYq_11 \mapsto XXq_2YY \mapsto Xq_2XYY \mapsto XXq_0YY \mapsto XXYq_3Y \mapsto XXYYq_3B \mapsto XXYYBq_4B$$

Pero no acepta **0010**:

$$q_00010 \mapsto Xq_1010 \mapsto X0q_110 \mapsto Xq_20Y0 \mapsto q_2X0Y0 \mapsto Xq_00Y0 \mapsto XXq_1Y0 \mapsto XXYq_10 \mapsto XXY0q_1B$$

Ya que en ese estado no está definido $\delta(q_1, B)$, y por lo tanto la máquina se detiene.

Aceptación por "parada"

Existe otro concepto de aceptación que se emplea en las Máquinas de Turing: aceptación por parada.

Una MT se detiene al aceptar un string, si no tiene un movimiento. Es decir, en un estado q, apuntando a un símbolo de cinta X, $\delta(q, X)$ no está definida.

Suponemos que una Máquina de Turing siempre se detiene cuando está en un estado de aceptación.

Entonces, para decidir si una cadena pertenece o no a un lenguaje:

Si M se detiene en un estado final de aceptación $\Rightarrow \omega \in L(M)$

Si M se detiene en un estado no final $\Rightarrow \omega \notin L(M)$

Si M no se detiene $\Rightarrow \omega \notin L(M)$

Al conjunto de lenguajes aceptados por una Máquina de Turing se los denomina lenguajes ***recusivamente enumerables***.

Por otra parte:

Si L es un **lenguaje del tipo 0 recursivo**, entonces existe una MT tal que:

Si $\omega \in L(M) \Rightarrow$ se detiene en un estado de aceptación.

Si $\omega \notin L(M) \Rightarrow$ se detiene en un estado no final.

El término ***recursivo*** se toma como sinónimo de ***decidible***. Previo a la existencia de las computadoras, los cálculos basados en la recursión se empleaban habitualmente como un concepto de computación, y no la iteración o los bucles (como ocurre en los lenguajes de programación funcional). Por lo que decir que un problema era recursivo, significaba que era lo suficientemente sencillo como para poder escribir una función recursiva para resolverlo y además se asegura que siempre terminará.

El término ***enumerable*** deriva del hecho que son precisamente estos lenguajes cuyos strings pueden ser enumerados (listados) por una máquina de Turing. Es decir, se podría enumerar todos los elementos del lenguaje en un cierto orden.

Extensiones de la Máquina de Turing.

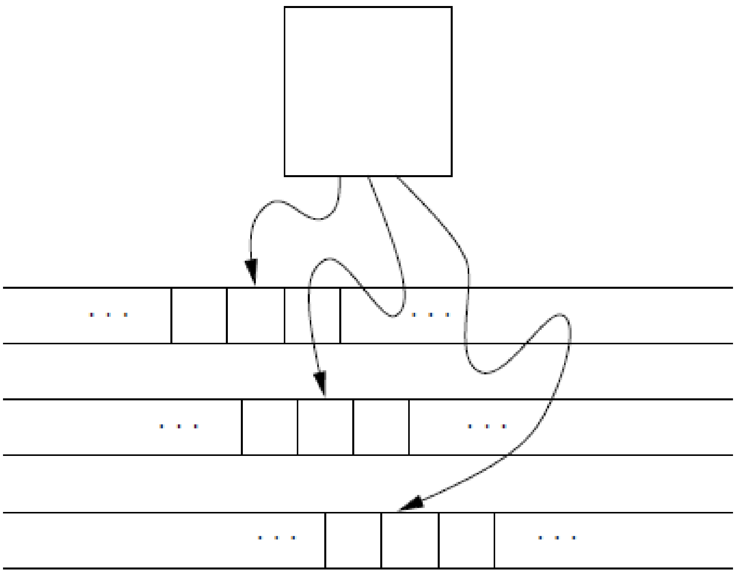
Máquina de Turing con Cinta Infinita en una Sola Dirección.

Se denota con la Tupla $MT =< Q, \Sigma, \Gamma, \delta, q_0, B, F >$ pero al ser la cinta infinita en una sola dirección, cambia la forma de denotar las descripciones instantáneas y los movimientos o secuencias entre configuraciones.

L es reconocido por una MT con ***cinta infinita en ambas direcciones*** si y sólo si es reconocido por una MT con ***cinta infinita en una sola dirección***.

Máquina de Turing con Varias Cintas.

Se denota con la Tupla $MT =< Q, \Sigma, \Gamma, \delta, q_0, B, F >$ pero el dispositivo tiene un número finito de cintas, cada una de las cuales está dividida en casillas, por lo que cambia el funcionamiento.



Inicialmente:

1. La entrada se coloca en la primera cinta.
2. Todas las casillas de las demás cintas contienen espacios en blanco.
3. La unidad de control se encuentra en el estado inicial.
4. La cabeza de la primera cinta apunta al extremo izquierdo de la entrada.
5. Las cabezas de las restantes cintas apuntan a una casilla arbitraria.

Luego, un movimiento de la MT depende del estado y del símbolo señalado por las cabezas de cada una de las cintas. Entonces, la máquina:

1. Cambia de estado (que puede volver a ser el mismo).
2. Escribir un nuevo símbolo en cada celda de cada cinta (o el mismo símbolo).
3. Mover la cabeza de las cintas a derecha o izquierda o no moverse (cada cinta en forma independiente)

Es decir que la función de transición se define $\delta: Q \times \Gamma^k \rightarrow Q \times (\Gamma \times \{I, D, S\})^k$ siendo k el número de cintas y S el movimiento "estacionario".

L es reconocido por una MT con ***varias cintas*** si y sólo si es reconocido por una MT con ***una sola cinta***.

Máquina de Turing No determinista

Se denota con la Tupla $MT =< Q, \Sigma, \Gamma, \delta, q_0, B, F >$ pero la función de transición δ es tal que para el estado q y símbolo X se obtiene un conjunto de tuplas.

Si L es reconocido por una MT ***no determinista*** entonces L es reconocido por una MT ***determinista***.

Autómatas Acotados Linealmente

Un Autómata Acotado Linealmente es una Máquina de Turing no determinista que usa una cinta de **longitud finita** que **es función lineal de la longitud de la cadena de entrada**.

Es una Tupla $AAL =< Q, \Sigma, \Gamma, \delta, q_0, B, F >$ donde:

- Q , conjunto finito de *estados*

- Σ , alfabeto de *entrada* ($\Sigma \subseteq \Gamma$). Contiene dos símbolos especiales: # y $ que son los marcadores de izquierda y derecha respectivamente. Estos símbolos están inicialmente en los extremos de la entrada y su función es evitar que la cabeza de cinta abandone la zona donde aparece la entrada.

- Γ, alfabeto de la *cinta*

- δ , función de transición $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D\}$

- $q_o \in Q$, *estado inicial*

- B es un símbolo que representa el *espacio en blanco*. ($B \in \Gamma, B \notin \Sigma$)

- $F \subseteq Q$, conjunto de *estados finales* o de *aceptación*.

Lenguaje aceptado por un AAL

Si G es una gramática de tipo 1, $G =< V, \Sigma, P, S >$, entonces existe un AAL que acepta $L(G)$

Funciones Recursivas Parciales.

La máquina de Turing puede verse como un computador de funciones de $\mathbb{N} \rightarrow \mathbb{N}$.

La forma tradicional es representar los enteros en unario, es decir $i \geq 0$ se representa por la secuencia 1^i. Si una función tiene k argumentos, $i_1, i_2, \dots i_k$, estos enteros se ponen inicialmente en la cinta separados por 0 u otro símbolo.

Si la MT se detiene con una cinta que consiste de 1^M, entonces se dice que $f(i_1, i_2, \dots i_k) = M$. Es decir, M es el resultado para la función de k argumentos.

Ejemplo:

Si $f(x, y) = x + y$, entonces $f(2,4)$ en la cinta se representa inicialmente como 1101111 o con otro separador $11 * 1111$.

Cuando se detenga, debería apuntar a 111111, ya que $f(2,4) = 6$

Si $f(i_1, i_2, \ldots, i_k)$ está definida para toda tupla $(i_1, i_2, \ldots i_k)$ entonces se dice que f es una ***función recursiva total***.

Una función $f(i_1, i_2, \ldots i_k)$computada por una máquina de Turing es llamada una ***función recursiva parcial***.

Todas las funciones aritméticas comunes en enteros, tales como la multiplicación, $n!$, 2^n, son funciones recursivas totales.

Máquina de Turing Universal

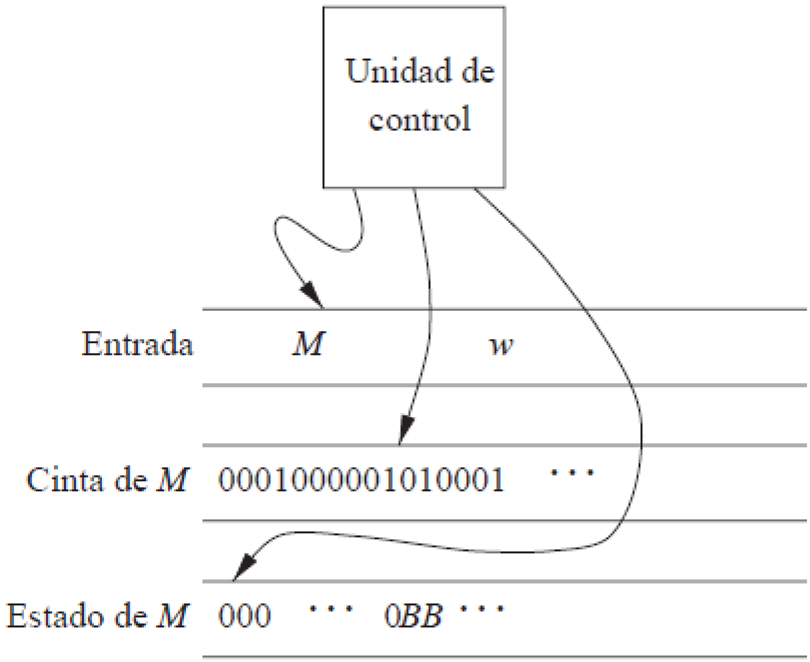
Es una Máquina de Turing capaz de simular el procesamiento de cualquier otra Máquina de Turing.

Una Máquina de Turing Universal puede determinar, para otra MT M, si una cadena es aceptada por esa MT M.

Entonces la MTU tiene como entrada una máquina de Turing válida codificada en binario, y una entrada también codificada en binario.

Constan de 3 cintas:

1. Programa de la MT M, y la cadena de entrada. Luego, Contendrá la salida de la máquina.
2. Área de trabajo.
3. Estado actual de la máquina simulada.



La MTU:

- Copia la cadena de entrada de la cinta 1 a la cinta 2.
- Graba el código del estado inicial en la cinta 3.
- Busca una transición aplicable en la máquina codificada de la cinta 1. Cuando la encuentra:
    - Realiza la transición en la cinta 2.
    - Escribe el nuevo estado en la cinta 3.
- Continua el proceso hasta que se llega al estado de parada de la máquina simulada. Entonces, copia la cinta 2 en la cinta 1, coloca la cabeza de la cinta 1 donde se encontraba la de la cinta 2, y se detiene.

Hipótesis de Church-Turing

A finales del siglo XIX, el matemático David Hilbert se preguntó si era posible encontrar un algoritmo, compuesto por una cantidad finita de pasos, que permita determinar la verdad o falsedad de cualquier proposición matemática.

En 1931, Kurt Göedel publicó su teorema de la incompletitud, mediante el cual probaba que todo sistema axiomático es incompleto y que existen sentencias cuya verdad o falsedad no se pueden probar. Para demostrarlas era necesario ampliar la base axiomática, pero entonces nuevas sentencias aparecerían que no podrían ser probadas.

En 1936, Alan Turing, en el artículo On Computable Numbers, construyó un modelo formal de computador, que es el que conocemos como la Máquina de Turing, primer modelo teórico de lo que luego sería un computador programable.

La ***hipótesis de Church o Tesis de Church-Turing*** sostiene que una función es **computable** si y sólo si puede ser computada en una máquina de Turing.

La hipótesis básica de la teoría de algoritmos es que cualquier algoritmo puede ser presentado por medio de un esquema funcional de Turing y realizado en la correspondiente máquina de Turing.

La Tesis de Church – Turing **no ha podido ser demostrada todavía,** pero se asume que lo es, ya que con otros formalismos como el cálculo - λ , sistemas de Post y funciones recursivas generales se ha demostrado que definen la misma clase de funciones que las máquinas de Turing, es decir, todos definen las funciones recursivas parciales.

Uno de esos formalismos es el RAM (Random Access Machine), que consiste en un número infinito de palabras de memoria, numeradas desde 0, cada una de la cuales puede almacenar un número entero, y un número finito de registros aritméticos capaces de almacenar números enteros. Los enteros pueden ser decodificados como instrucciones en la forma usual de los computadores. Si se elige un conjunto adecuado de instrucciones, la RAM puede simular cualquier computador existente.

Se puede demostrar que una MT puede simular una RAM, siempre que las instrucciones de la RAM puedan ser simuladas por una MT.

Simulación de una máquina de Turing mediante una computadora real.

En principio, es posible simular una máquina de Turing mediante una computadora real si aceptamos que existe un suministro potencialmente infinito de un dispositivo de almacenamiento extraible (disco), para simular la parte en que no hay espacios en blanco de la cinta de la MT. Dado que los recursos físicos con los que se construyen los discos no son infinitos, este argumento es cuetionable. Sin embargo, la suposición de que existen recursos infinitos, como en una cinta de una MT, es realista en la práctica y en general se acepta.

Simulación de una computadora mediante una máquina de Turing.

Una máquina de Turing puede simular el almacenamiento y la unidad de control de una computadora real empleando una cinta para almacenar todas las posiciones y los contenidos correspondientes a los registros, la memoria principal, los discos y otros dispositivos de almacenamiento. Por tanto, podemos estar seguros de que algo que una máquina de Turing no pueda hacer tampoco lo podrá hacer una computadora real.

Tipos de Problemas

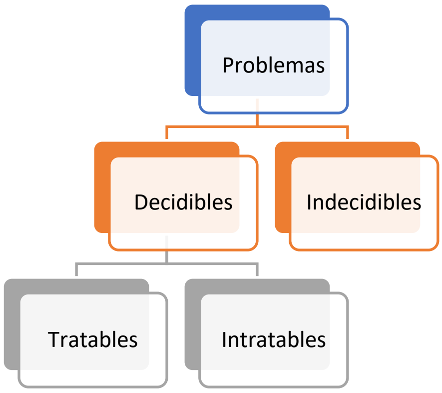
Problemas Decidibles

Son aquellos que pueden resolverse mediante una Máquina de Turing.

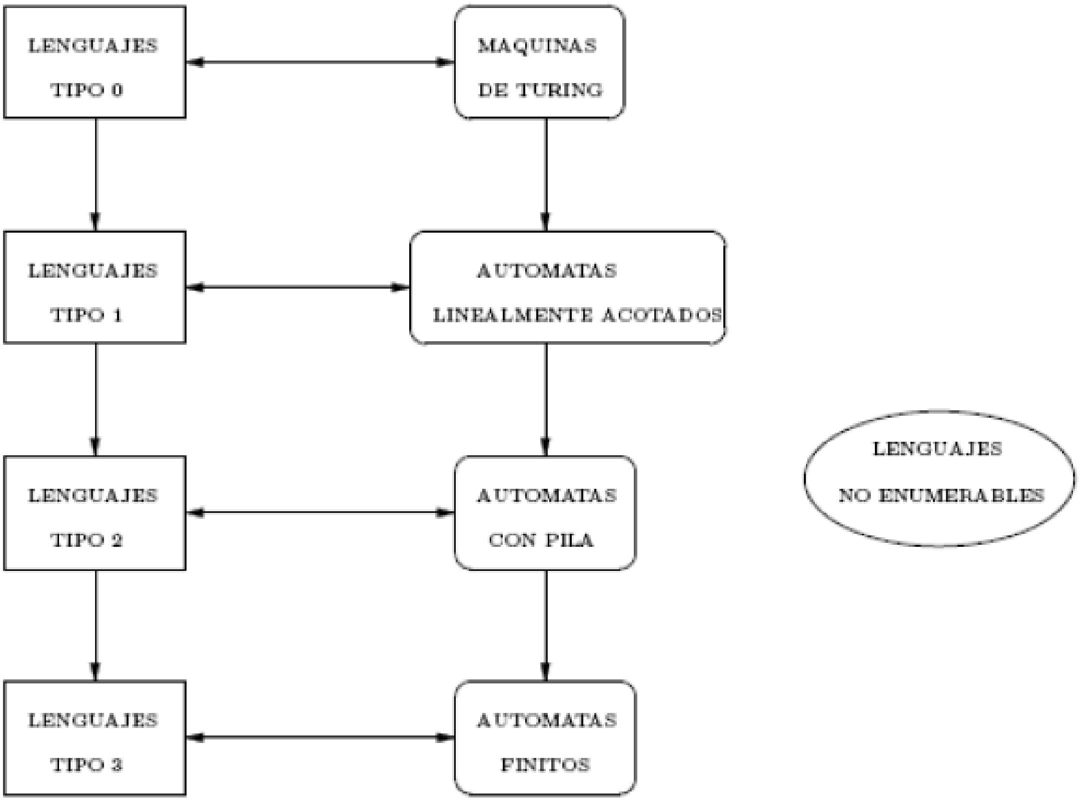
Problema de la parada es indecidible: Existen máquinas de Turing para las que no es posible decidir si se van a detener ante una cadena de entrada.

Problemas Tratables

Son aquellos para los que existe al menos un algortimo capaz de resolverlo en un tiempo razonable.



Lenguajes, gramáticas y máquinas



Lenguaje	Máquina	Gramática
Regular (tipo 3)	Autómata Finito	Regular
Libre deContexto (tipo 2)	Autómata de Pila	Libre de Contexto
Sensible al Contexto (tipo 1)	Autómata Linealmente Acotado	Sensible al Contexto
Recursivamente Enumerables (tipo 0)	Máquina de Turing	Tipo 0