**Tesis de Church-Turing** sostiene que todo lo computable puede ser llevado a cabo por una máquina de Turing

Una máquina de Turing es un artefacto que resuelve problemas (computacionales), en su visión más general calculando una solución. Ej: encontrar un camino del vértice v1 al vértice v2 en un grafo. Una máquina de Turing **calculadora** devuelve un camino en el grafo del vértice v1 al vértice v2, si es que existe.

Una segunda visión de máquina de Turing es la de una máquina que sólo puede resolver un problema de decisión, produciendo únicamente la respuesta sí o no ante una instancia. Volviendo al ejemplo anterior, la máquina acepta los grafos que tienen un camino del vértice v1 al vértice v2, y rechaza al resto. Esta es la visión de máquina de Turing **reconocedora** en la que la resolución de un problema consiste en reconocer el lenguaje de las cadenas de símbolos que representan las instancias positivas del problema.

Hay todavía una tercera visión de máquina de Turing, según la cual se resuelve un problema generando todas sus instancias positivas. En el ejemplo anterior devolvería todos los distintos caminos del grafo que vayan desde v1 a v2. Estas son denominadas **máquinas generadoras**.

Una **MT M** está compuesta por:   
• Una **cinta** infinita en los dos extremos, dividida en celdas. Cada celda puede almacenar un símbolo.

• Una **unidad de control**. En todo momento la unidad de control almacena el estado corriente de M.

• Un **cabezal**. En todo momento el cabezal apunta a una celda. El símbolo apuntado se denomina símbolo corriente. El cabezal puede moverse sólo de a una celda por vez, a la izquierda o a la derecha.

Los **estados** pertenecen a un conjunto **Q**, y los **símbolos** a un alfabeto **Γ**. La cinta tiene la entrada limitada a izquierda y derecha por infinitos símbolos blancos (**B**). La unidad de control almacena el estado inicial (**q0**). Si la entrada es la cadena vacía (**λ**) entonces el cabezal apunta a algún blanco. A partir de la configuración inicial, M se comporta de acuerdo a lo especificado en su función de transición **δ.**

M en cada paso lee un estado y un símbolo. Cuando **δ** no está definida para el estado corriente y el símbolo corriente, M se detiene. Si M nunca se detiene, se dice que entra en un *loop***.**

Una máquina de Turing M es una 6-tupla (Q, Ʃ, Γ, δ, q0, F), tal que:

• Q es el conjunto de estados de M.

• Ʃ es el alfabeto de las entradas de M.

• Γ es el alfabeto de las cadenas de la cinta de M. Por convención, B ∈ (Γ– Ʃ).

• δ es la función de transición de M. Se define δ: Q x Γ → Q x Γ x {L, R, S}, tal que L representa el movimiento del cabezal a la izquierda, R el movimiento a la derecha, y S indica que el cabezal no se mueve.

• q0 es el estado inicial de M.

• F es el conjunto de estados finales de M (su significado se aclara a continuación).

Considerando la visión de **MT reconocedora** de un lenguaje, si a partir de la entrada **w** la MT M se detiene en un estado **q ∈ F**, se dice que M *acepta w*. En cambio, cuando se detiene en un estado **q ∈ (Q – F)** o no se detiene, se dice que M *no acepta (o rechaza) w.* El conjunto de las cadenas aceptadas por la MT M es el *lenguaje reconocido* por M, y se denota con **L(M)**. Considerando la visión de **MT calculadora**, sólo cuando M se detiene en un estado **q ∈ F** debe tenerse en cuenta el contenido final de la cinta, es decir, la salida.

**Clase 2 - Jerarquía de la computabilidad**

Hay una jerarquía de clases de lenguajes teniendo en cuenta si son reconocidos y de qué manera por una máquina de Turing. A continuación se describe dicha jerarquía de la computabilidad, y se prueban algunas propiedades de las clases de lenguajes que la componen. Para facilitar las demostraciones desarrolladas, las máquinas de Turing utilizadas se describen con mayor nivel de abstracción que en la clase anterior.

Un lenguaje es **recursivamente numerable** si y sólo si existe una MT que lo reconoce.

Es decir, si L es el conjunto de todos los lenguajes, Ʃ), sólo los lenguajes recursivamente numerables de L son reconocibles por una MT, por esto es que a los problemas de decisión asociados se los conoce como **computables**.

La clase de los lenguajes recursivamente numerables se denomina **RE** (por recursively enumerable languages).

De esta manera, dado L ∈ RE, si M es una MT tal que L(M) = L, se cumple para toda cadena w de Ʃ\* que:

* Si w ∈ L, entonces M a partir de w se detiene en su estado qA.
* Si w ∉ L, entonces M a partir de w se detiene en su estado qR o no se detiene.

No todos los lenguajes son recursivamente numerables, y sólo algunos tienen la propiedad de que las MT que los reconocen se detienen siempre. Considerando este último caso, se define que un lenguaje es recursivo si y sólo si existe una MT M que lo reconoce y que se detiene cualquiera sea su entrada. La clase de los lenguajes recursivos se denomina **R**. A los problemas de decisión asociados se los conoce como **decidibles**, porque las MT que los resuelven pueden justamente decidir, cualquiera sea la instancia, si es positiva o negativa.

Ahora, dado L ∈ R, si M es una MT tal que L(M) = L, se cumple para toda cadena w de Ʃ\* que:

* Si w ∈ L, entonces M a partir de w se detiene en su estado qA.
* Si w ∉ L, entonces M a partir de w se detiene en su estado qR.

No todos los problemas computables son decidibles, y no todos los problemas son computables.

**Algunas propiedades de clausura de la clase R**

Considerando las operaciones de complemento, intersección, unión y concatenación de lenguajes, se cumple que la clase R es cerrada con respecto a todas ellas.

Considerando las operaciones de intersección, unión y concatenación de lenguajes, se

cumple que también la clase RE es cerrada con respecto a ellas. En cambio, a diferencia

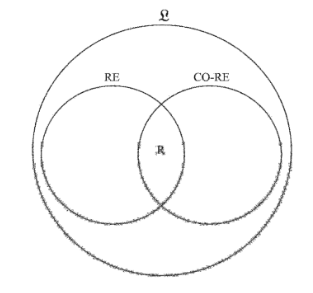
de la clase R, RE no es cerrada con respecto al complemento,

Sea CO-RE la clase de los lenguajes complemento, con respecto a Ʃ\*, de los lenguajes

recursivamente numerables. Formalmente: CO-RE = {L | L ∈ ∧ LC ∈ RE}.

Considerando CO-RE, la siguiente figura muestra una versión más detallada de la

jerarquía de la computabilidad:



De la figura se desprende que un lenguaje L es recursivo si y sólo si tanto L como LC

son recursivamente numerables,

Se distinguen cuatro categorías de lenguajes. Enumeradas de acuerdo a su dificultad creciente, son:

1. R

2. RE – R

3. CO-RE – R

4. L– (RE ⋃ CO-RE)

En este contexto, dado un par cualquiera de lenguajes L y LC,, se cumple alternativamente que:

* Tanto L como LC pertenecen a R. (teorema 2.1)
* L pertenece a RE – R, y LC pertenece a CO-RE – R.
* Tanto L como LC pertenecen a – (RE ⋃ CO-RE).

---------------------