

# LABORATORIO LENGUAJES FORMALES Y DE PROGRAMACIÓN

**Clase 3**

# AGENDA

- Aclaración de preguntas o dudas.
- Autómatas Finitos
- Diagramas de estados finitos
- Tabla de transiciones
- Autómata Finito Determinista y no determinista.
- Ejemplo

AUTOMATA FINITO

# AUTOMATA FINITO

Es una máquina que acepta las palabras validas para el lenguaje y rechaza las incorrectas. Un FA permite reconocer los componentes léxicos de un lenguaje.

---

# COMPONENTES DE UN FA

- Conjunto de estados ( $S$ )
- Conjunto de terminales o alfabeto ( $\Sigma$ )
- Conjunto de transiciones ( $T$ )
- Estado inicial ( $S_0$ )
- Estado de finalización o aceptación ( $F$ )

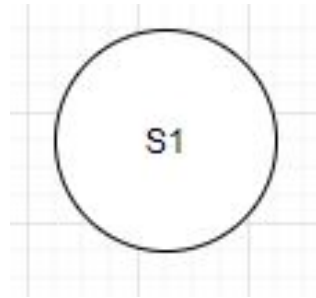
# CARACTERÍSTICAS

- Solo posee un único estado de inicio.
- Los estados finales pueden ser más de uno.
- Puede poseer varios estados de transición.
- Puede darse el caso en que el estado inicial corresponde al estado final.

# DIAGRAMAS DE ESTADOS FINITOS

# DIAGRAMA DE ESTADOS FINITOS

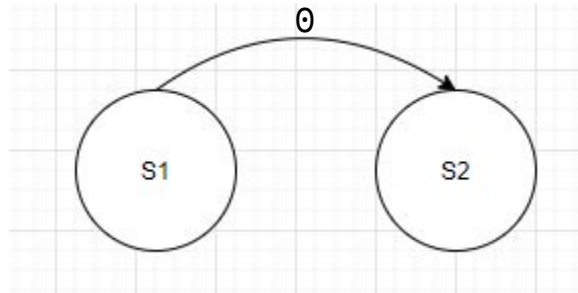
Los estados se representa como vértices, etiquetados con su nombre en el interior.





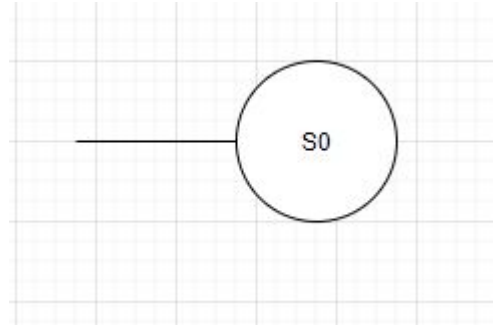
# DIAGRAMA DE ESTADOS FINITOS

Una transición desde un estado a otro, depende de un símbolo del alfabeto y se representa mediante una flecha dirigida que une dos estados (vértices) y en qué está etiquetada con dicho símbolo.



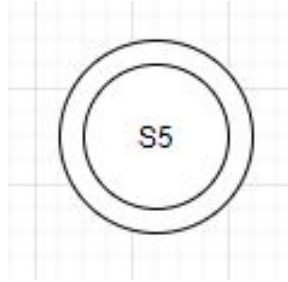
# DIAGRAMA DE ESTADOS FINITOS

El estado inicial se caracteriza por tener una arista que llega a el (esta arista no proviene de ningún otro vértice)



# DIAGRAMA DE ESTADOS FINITOS

El o los estados finales se representa mediante vértices que están encerrados a su vez por otra circunferencia.



# FUNCIONAMIENTO DEL AUTÓMATA FINITO

En el comienzo del reconocimiento de una cadena, el AF se encuentra en el estado inicial y a medida que procesa cada símbolo de la cadena este va cambiando de estado de acuerdo a lo determinado por la función de transición, cuando se ha procesado el último de los símbolos de la cadena de entrada, el autómata se detiene. Si el estado en el que se detuvo es un estado de aceptación, entonces la cadena pertenece al lenguaje, en caso contrario, la cadena no pertenece a dicho lenguaje.

EJEMPLOS

# EJEMPLO 1

Diseñar un Autómata finito que acepte todas las cadenas sobre  $\{0,1\}$  que tenga un número impar de 1's.

0001

100

# EJEMPLO 1

## Sugerencias:

Defina una lista de posibles estados de su cadena:

Par de 1's hasta ahorita

Impar de 1's hasta ahorita

01

# EJEMPLO 1

## Sugerencias

Defina las cadenas mínimas que acepta su lenguaje:

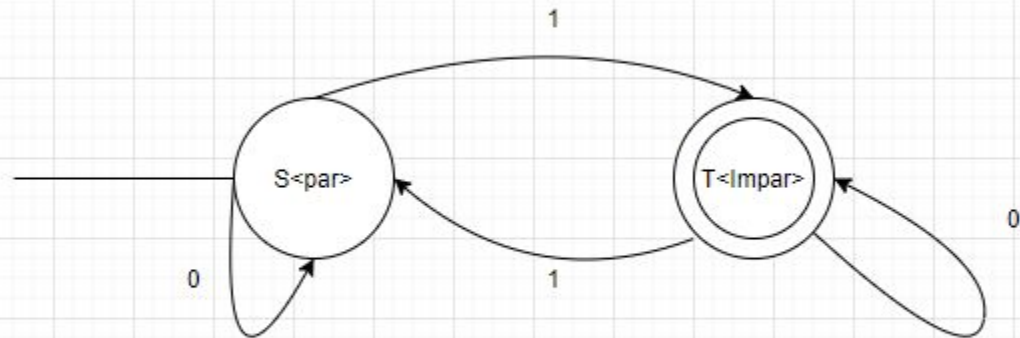
1

01

10



# EJEMPLO 1



# DEMÁS PARTES DEL AUTÓMATA FINITO

Conjunto de terminales

$$\Sigma = \{0,1\}$$

Conjunto de estados

$$S = \{S,T\}$$

Estado Inicial

$$S_0 = \{S\}$$

# DEMÁS PARTE DEL AUTÓMATA FINITO

Conjunto de transiciones

$$T = \{(S,0,S), (S,1,T), (T,0,T), (T,1,S)\}$$

Conjunto de estados de aceptación

$$F = \{T\}$$

# TABLA DE TRANSICIONES

# TABLA DE TRANSICIONES

- Los estados de aceptación se marcan con un asterisco (\*)
- Las columnas se colocan los terminales.
- En las filas se colocan los estados.

# DEL EJEMPLO ANTERIOR

Estado \ Terminales	0	1
S<par 1's>	S	T
*T<impar 1's>	T	S

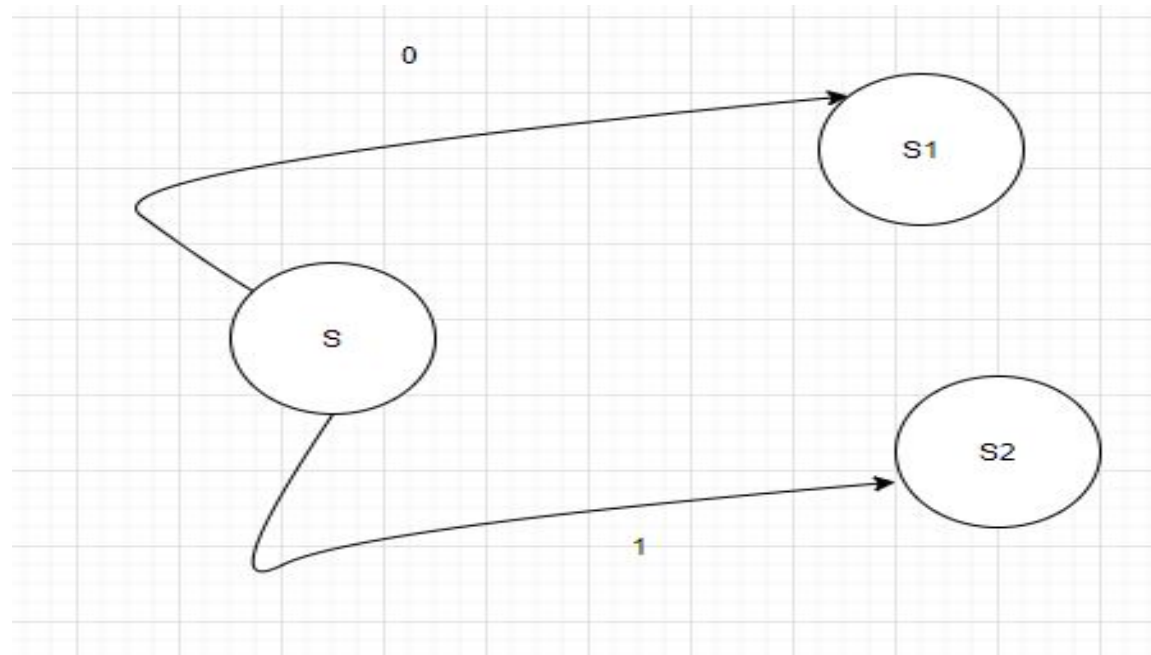
# AUTÓMATA FINITO DETERMINISTA (DFA/AFD)

# CARACTERÍSTICAS DFA

- Existe exactamente una transición posible de cada estado, dado un posible símbolo de entrada, no contiene ambigüedad.
- Un DFA no tiene transiciones con epsilon (cadenas vacías)



# EJEMPLOS

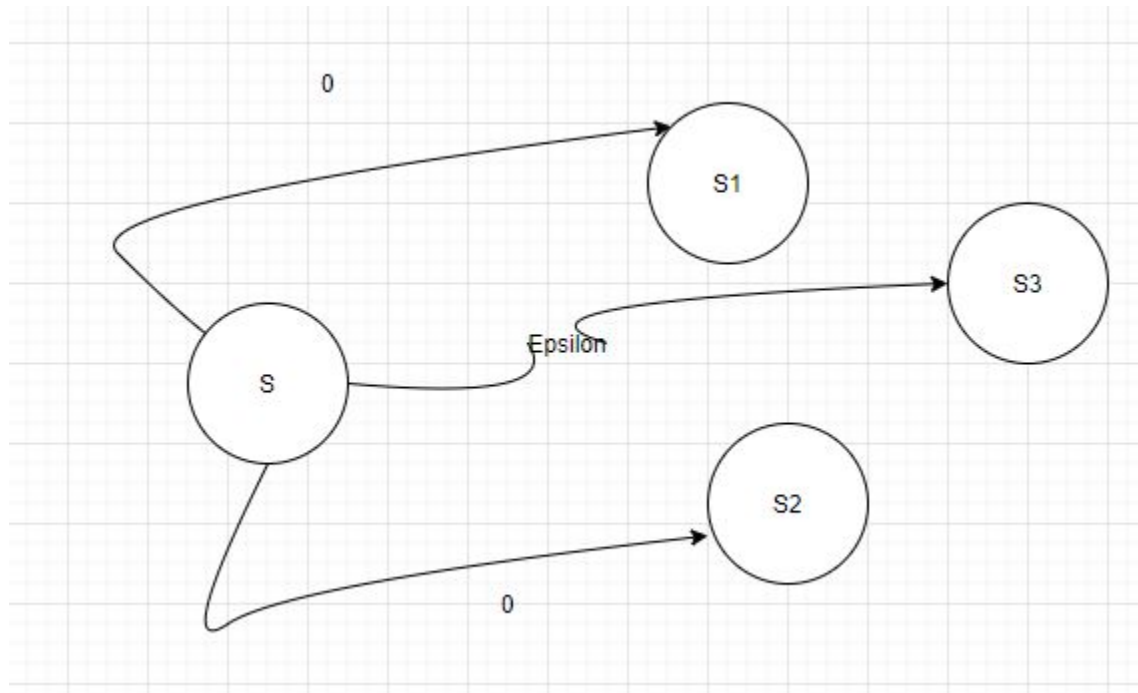


AUTÓMATA FINITO  
NO  
DETERMINISTA (NFA/  
AFN)

# CARACTERÍSTICAS

- Si se ofrece multiples selecciones de un estado para algún símbolo Terminal, es decir desde un estado tenemos varias transiciones para el mismo símbolo terminal. Posee ambigüedad.
- Si puede contener transiciones con epsilon

# EJEMPLOS



# EJEMPLO PRÁCTICA

<https://github.com/Feliciano07/2S2021-LFP.git>