



TECNOLÓGICO  
NACIONAL DE MÉXICO®



# TECNOLÓGICO NACIONAL DE MÉXICO

## INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INTEGRANTES:

GUTIERREZ ARELLANO RAFAEL  
181080022

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 7



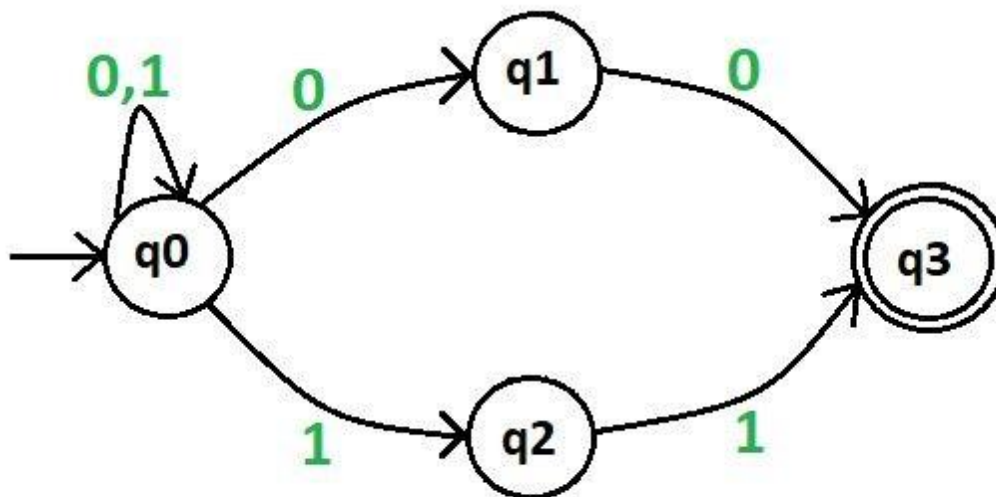
GUTIERREZ ARELLANO RAFAEL

### Autómata finito NO-Determinista.

Este tipo de autómata es aquel que permite al autómata finito (AF) tiene 0 o más estados consecutivos para cada estado y su entrada.

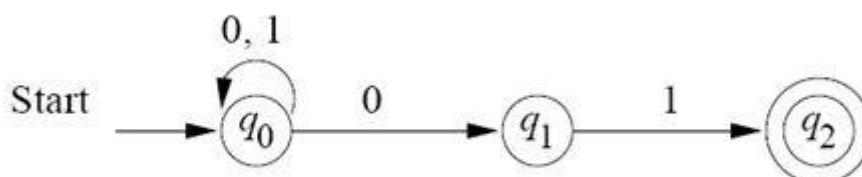
Con lo anterior podemos decir que un NFA puede estar en varios estados a la vez o que se puede que 'adivina' o que estado debe ir.

NFA



Los autómatas finitos se pueden representar mediante [grafos](#) [s](#) particulares, también llamados diagramas de estados finitos

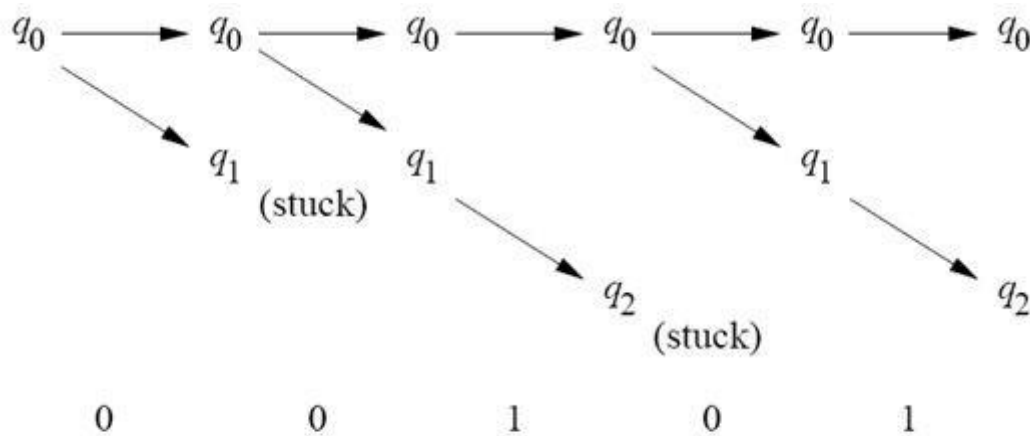
Por ejemplo, el siguiente autómata acepta todas las cadenas que terminan en 01:





Lo que pasa al procesar como entrada a 00101

es:



### EN SU FUNCIONAMIENTO:

1. En el comienzo del proceso de reconocimiento de una cadena de entrada.
2. el autómata finito se encuentra en el *estado inicial* y a medida que procesa cada símbolo de la cadena va cambiando de estado de acuerdo a lo determinado por la *función de transición*.
3. Cuando se ha procesado el último de los símbolos de la cadena de entrada, el autómata se detiene en el estado final del proceso.
4. Si el estado final en el que se detuvo es un estado *de aceptación*, entonces la cadena pertenece al lenguaje reconocido por el autómata; en caso contrario, la cadena no pertenece a dicho lenguaje.

Otra manera de describir el funcionamiento de un autómata finito es mediante el uso de tablas de transiciones o matrices de estados.

### AUTÓMATA DFA

Autómata finito determinista, un autómata finito determinista ( DFA ), también conocido como aceptor finito determinista ( DFA ), máquina de estado finito determinista ( DFSM ) o autómata de estado finito determinista ( DFSA ) - es una máquina de estados finitos que acepta o rechaza una determinada cadena de



símbolos, mediante la ejecución de una secuencia de estados determinada únicamente por la cadena.

## Complemento

Los autómatas finitos deterministas siempre están completos : definen una transición para cada estado y cada símbolo de entrada.

Si bien esta es la definición más común, algunos autores usan el término autómata finito determinista para una noción ligeramente diferente: un autómata que define como máximo una transición para cada estado y cada símbolo de entrada; se permite que la función de transición sea parcial <sup>[5]</sup>. Cuando no se define ninguna transición, dicho autómata se detiene.

Los DFA son equivalentes en potencia de cálculo a los autómatas finitos no deterministas (NFA). Esto se debe a que, en primer lugar, cualquier DFA también es un NFA, por lo que un NFA puede hacer lo que un DFA puede hacer. Además, dada una NFA, utilizando la construcción <sup>[13]</sup> <sup>[14]</sup> del conjunto de potencias <sup>[5]</sup> se puede construir una DFA que reconozca el mismo lenguaje que la NFA, aunque la DFA podría tener un número exponencialmente mayor de estados que la NFA. <sup>[13]</sup> <sup>[14]</sup> Sin embargo, aunque los NFA son computacionalmente equivalentes a los DFA, los problemas mencionados anteriormente no se resuelven necesariamente de manera eficiente también para los NFA. El problema de no universalidad de los NFA es el PSPACE completo, ya que hay NFA pequeños con la palabra de rechazo más corta en tamaño exponencial. Un DFA es universal si y sólo si todos los estados son estados finales, pero esto no es válido para los NFA. Los Problemas de Igualdad, Inclusión y Minimización también son PSPACE completos ya que requieren formar el complemento de una NFA lo que resulta en una explosión exponencial de tamaño.

## MINIMIZACIÓN.

### Pasos a la minimización

1. Eliminación de los estados no alcanzables
2. Búsqueda de estados equivalentes
3. Construcción de un autómata a partir de los grupos de estados

equivalentes EJEMPLO:



	0	1	
-----			
->	A	B	A
	B	A	C
	C	D	B
*	D	D	A
	E	D	F
	F	G	E
	G	F	G
	H	G	D

La idea es ir marcando dentro de una tabla qué pares de estados no son equivalentes. Cuando terminamos de ejecutar este algoritmo, las casillas no marcadas de la tabla nos indican los estados equivalentes.

primer paso: marcar como distinguibles los pares de estados finales/no finales.

paso iterativo: si  $r$  y  $s$  son dos estados ya marcados como distinguibles, y si para un símbolo de entrada  $a$  tenemos que  $\delta(p, a) = r$  y  $\delta(q, a) = s$ , entonces marcamos  $p$  y  $q$  como distinguibles.

B	x						
C	x	x					
D	x	x	x				
E		x	x	x			
F	x	x	x		x		
G	x	x	x	x	x	x	
H	x		x	x	x	x	x
	A	B	C	D	E	F	G



TECNOLÓGICO  
NACIONAL DE MÉXICO®

