Instituto Tecnológico de Costa Rica
San José

Compiladores e Intérpretes

Apuntes de la clase del día viernes 24 de marzo del 2017

Apuntador:
Bryan Steve Jiménez Chacón (2014114175)

Profesor:
Dr. Francisco Torres Rojas

I Semestre 2017

Contenido

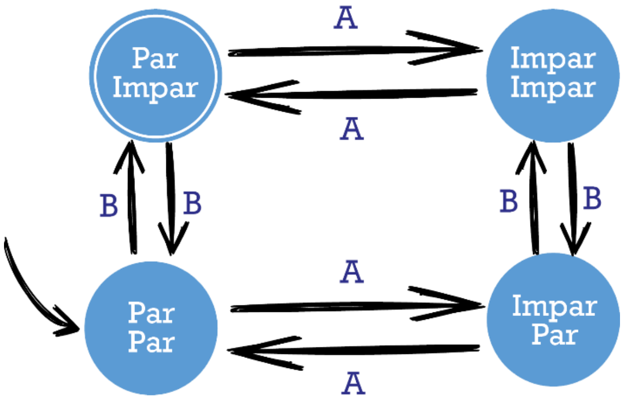
Importante:

- Tenemos de tarea resumir el Cap 4 de Pinker, se entrega el viernes 31 de marzo impreso y hecho en **LaTeX.**
- Para el miércoles 29 de marzo tenemos la entrega de la tarea de Autómatas, la podemos enviar por correo para que no imprimir tantas hojas, y los autómatas tienen que estar en **LaTeX** (no se vale hacer dibujos en Paint e incrustarlos.).
- El miércoles 5 de abril es la entrega y revisión del proyecto 1.
- El viernes 7 de abril es el primer examen parcial.

Continuamos con más ejemplos de autómatas:

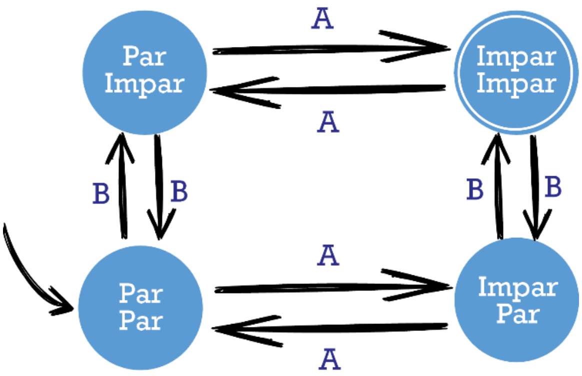
Ejemplo 15

- Sea L el lenguaje sobre Σ = {A, B} de hileras que contengan un número par de As y un número impar de Bs.
- Diseñe un DFA que reconozca a L.



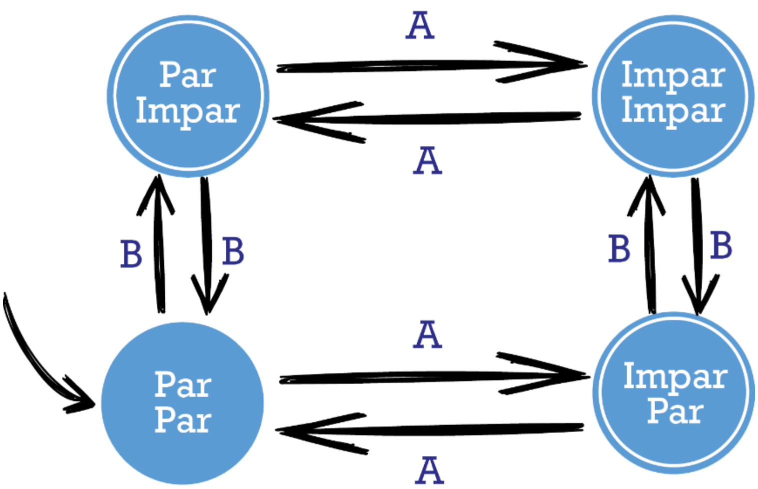
Ejemplo 16

- Sea L el lenguaje sobre Σ = {A, B} de hileras que contengan un número impar de As y un número impar de Bs.
- Diseñe un DFA que reconozca a L.



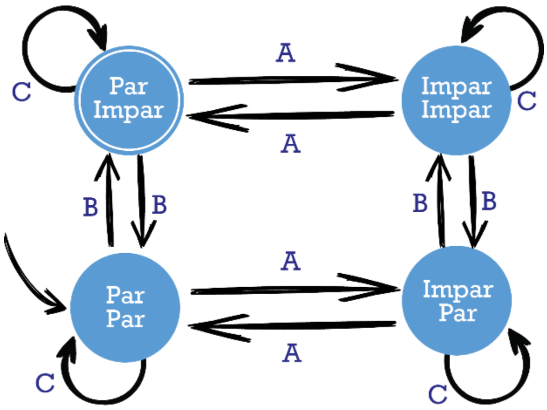
Ejemplo 17

- Sea L el lenguaje sobre $\Sigma = \{A, B\}$ de hileras que contengan un número impar de As **o** un número impar de Bs.
- Diseñe un DFA que reconozca a L.



Ejemplo 18

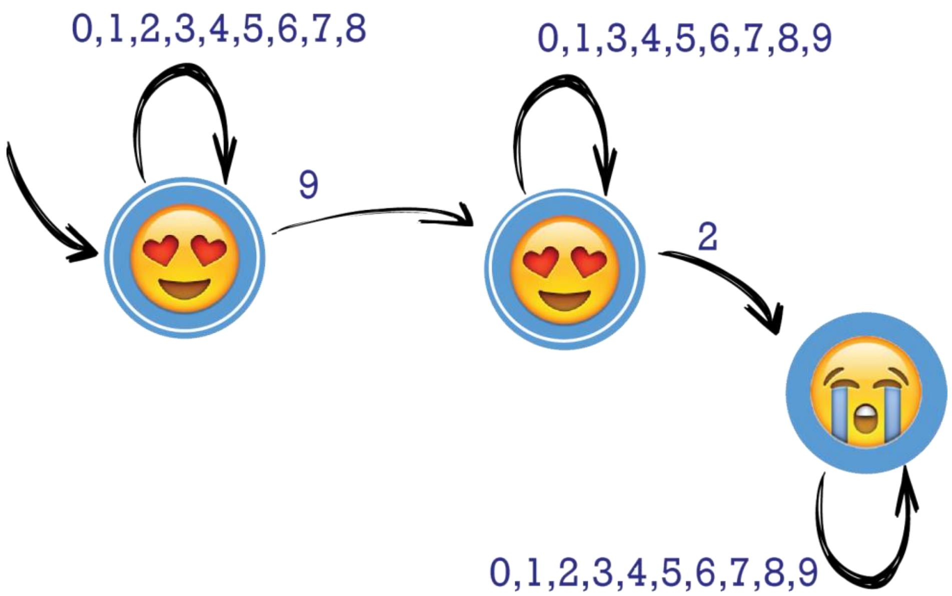
- Sea L el lenguaje sobre Σ = {A, B, C} de hileras que contengan un número par de As y un número impar de Bs.
- Diseñe un DFA que reconozca a L.



Como vimos estos ejercicios son cajoneros, se parecen mucho en el diseño del autómata y cuando uno ya lo entiende solo es cuestión de cambiar los estados de aceptación.

Ejemplo 19

- Sea L el lenguaje sobre Σ = {1, 2, 3, 4, 5, 6, 7, 8, 9} de hileras donde todos los 2 aparezcan antes que cualquier 9.
- Diseñe un DFA que reconozca a L.



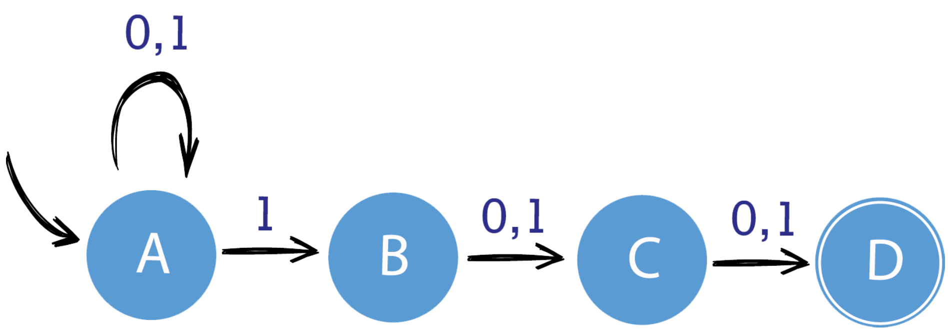
Lo importante acá es el 9, hasta no verlo estoy pura vida, cuando veo uno tengo que recordarlo y desde entonces puedo ver lo que sea menos un 2, si lo vemos entonces se murió y entramos al infierno.

Análisis Léxico
Autómatas No Determinísticos de Estados Finitos

Ejemplo 1

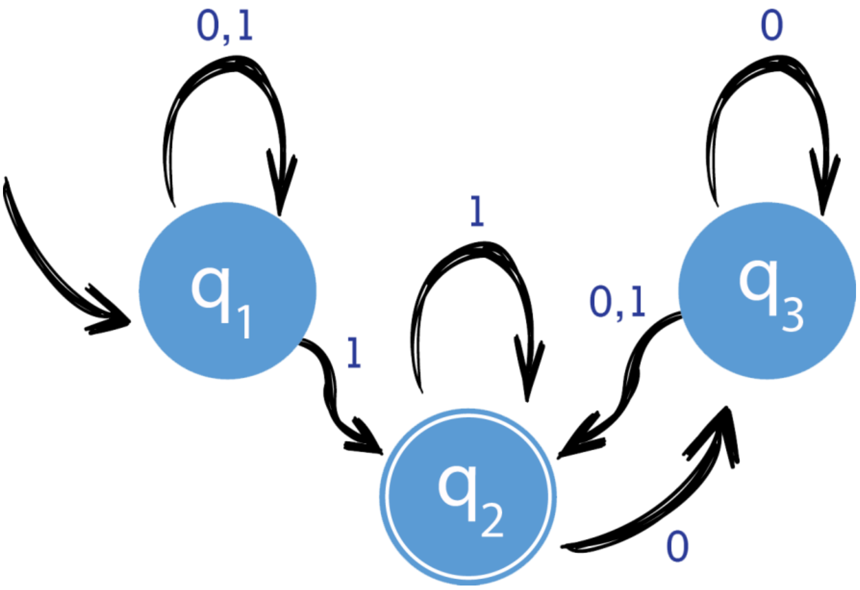
- Sea L el lenguaje sobre $\Sigma = \{0,1\}$ de hileras tales que el tercer símbolo de derecha a izquierda sea un 1.
- Diseñe un autómata que reconozca a L.
- ¿Problema?
- Hay 2 transiciones posibles desde el estado A con el símbolo 1.

El autómata no determinístico adivina cosas.



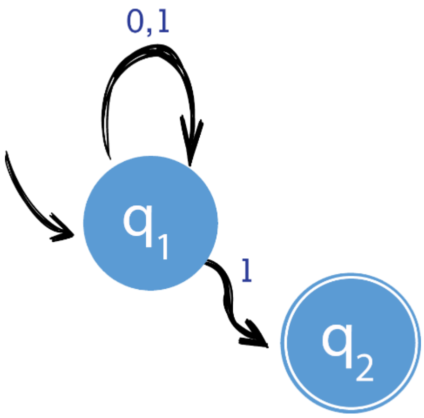
Ejemplo 2

Y siempre se va al camino correcto (si existe)



No Determinismo

- Existen estados donde un mismo símbolo tiene varias transiciones posibles.



Este autómata podemos ver que reconoce hileras de 0, 1 y que terminen en 1.

- **Non-Deterministic Finite States Automaton (NFA)**

Pregunta

¿Ya toda nuestra vida está planteada o no? ¿Somos determinísticos o no?
dios (con minúscula), sabe todo lo que nos va a pasar?

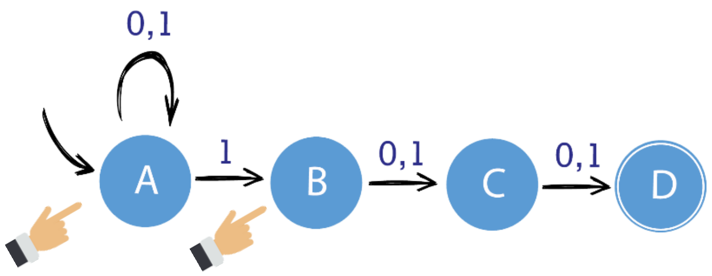
Aceptación de NFAs

- Al procesar una hilera podría haber varias rutas posibles (no determinismo).
- La hilera se podría acabar en diferentes lugares.
- Un NFA **rechaza** a una hilera **al no existir alguna** que partiendo del estado inicial y avanzando con las transiciones disponibles llegue a un estado de aceptación. Ósea que por más que se trató no se encontró una solución.
- Un NFA **acepta** a una hilera si **existe al menos una ruta** que partiendo del estado inicial y avanzando con las transiciones disponibles llegue a un estado de aceptación. Ósea que si por medio de transiciones se llegó a n estado de aceptación se acepta, no importa que existan 1000 fallos, si hay por lo menos 1 acierto.

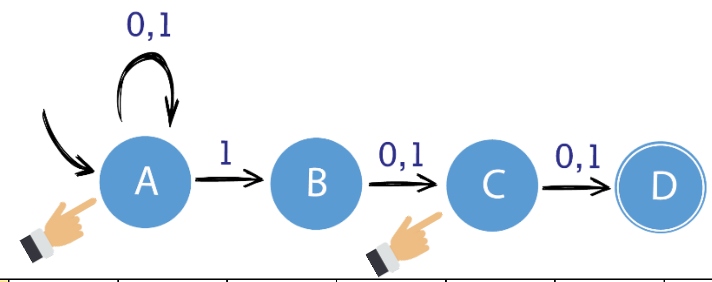
Ejemplo 3

Para estos autómatas usamos un montón de dedos para ver donde podemos estar. Si encontramos una ruta que cumplió entonces aceptamos la hilera. (Como en este caso).

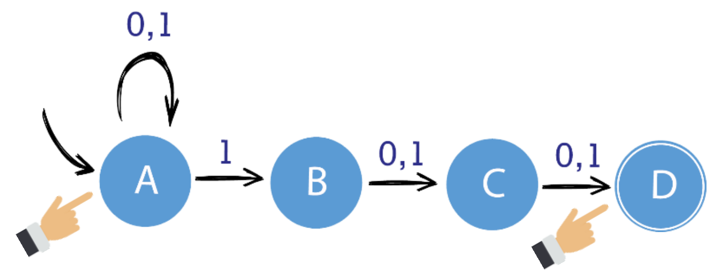
1	0	0	1	0	1	1	1	0	1	1	0	1



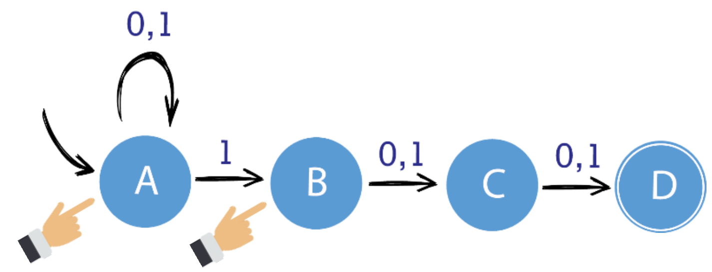
1	0	0	1	0	1	1	1	0	1	1	0	1



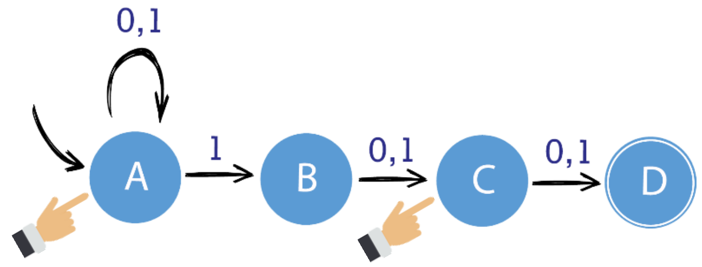
1	0	0	1	0	1	1	1	0	1	1	0	1



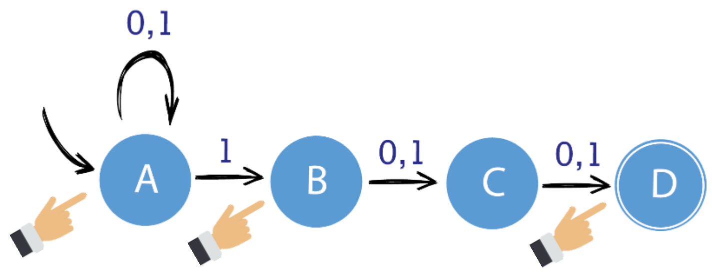
1	0	0	1	0	1	1	1	0	1	1	0	1



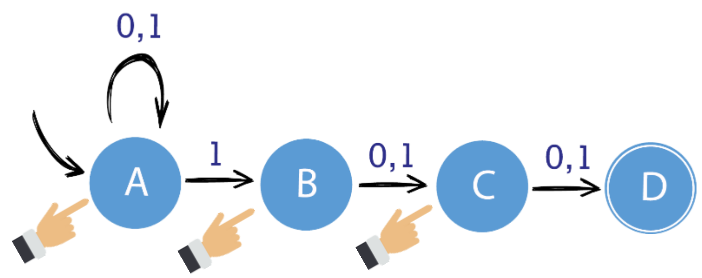
1	0	0	1	0	1	1	1	0	1	1	0	1



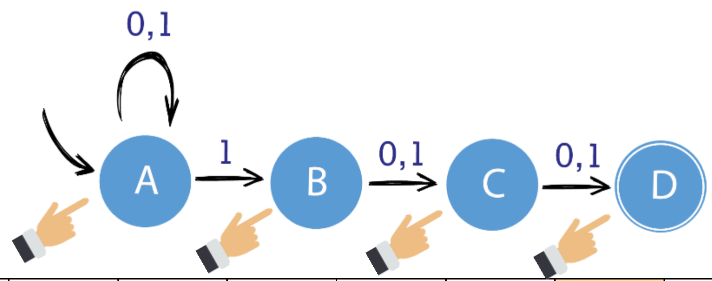
1	0	0	1	0	**1**	1	1	0	1	1	0	1



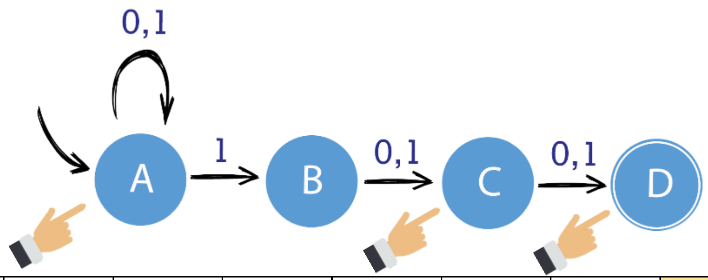
1	0	0	1	0	1	**1**	1	0	1	1	0	1



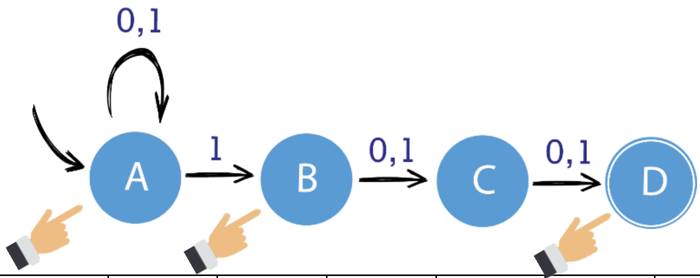
1	0	0	1	0	1	1	**1**	0	1	1	0	1



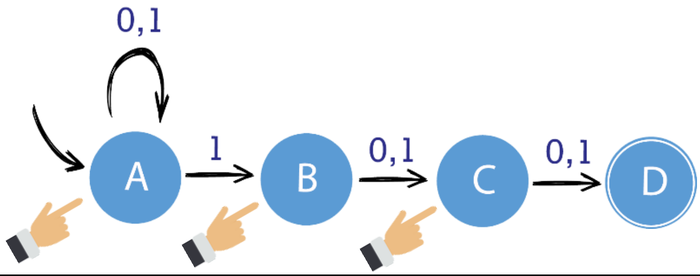
1	0	0	1	0	1	1	1	**0**	1	1	0	1



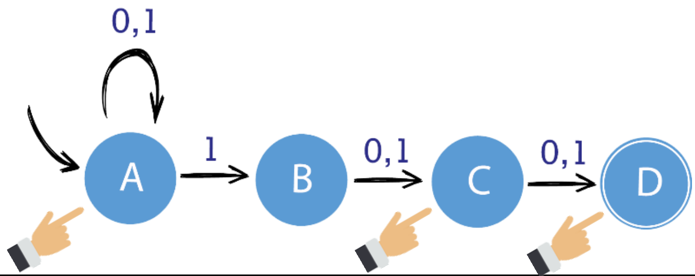
1	0	0	1	0	1	1	1	0	**1**	1	0	1



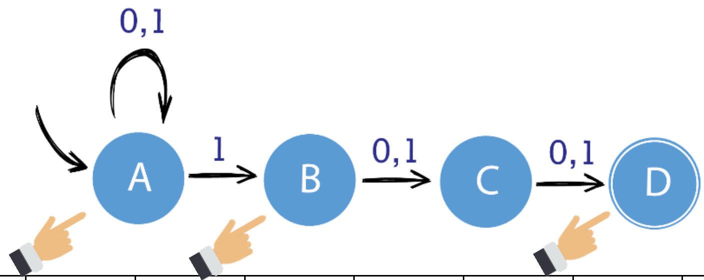
1	0	0	1	0	1	1	1	0	1	1	0	1



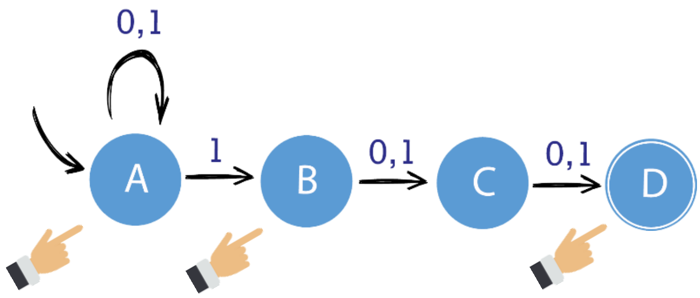
1	0	0	1	0	1	1	1	0	1	1	0	1



1	0	0	1	0	1	1	1	0	1	1	0	1



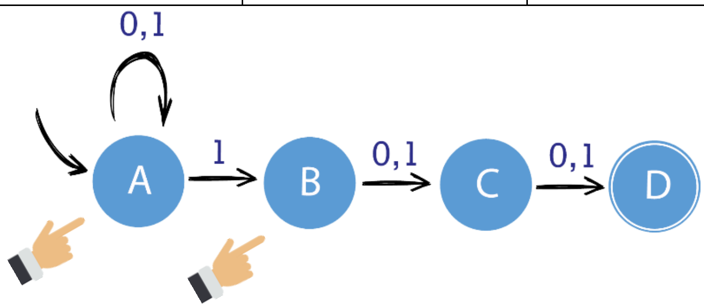
1	0	0	1	0	1	1	1	0	1	1	0	1	



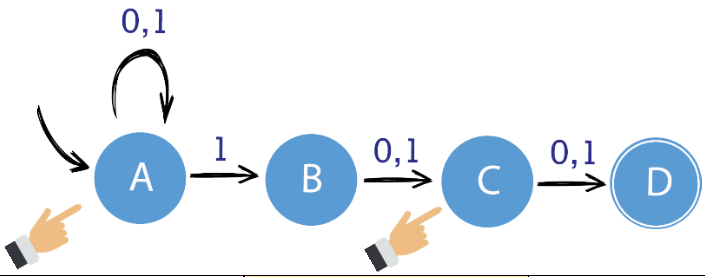
Ejemplo 4

Acá terminamos rechazando la hilera porque no existe una ruta que de un resultado de aceptación.

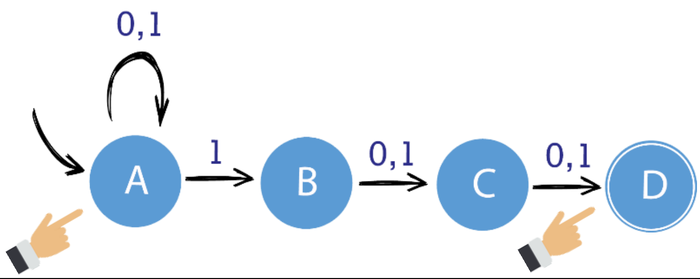
1	0	0	1	0



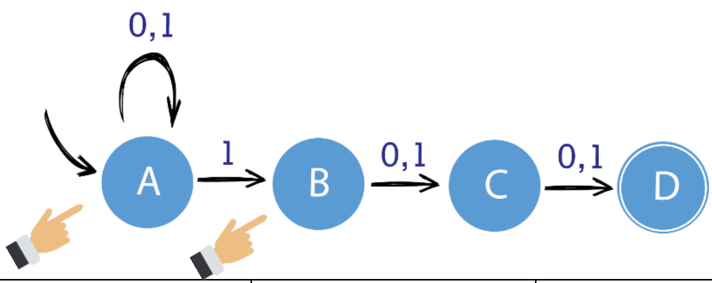
1	0	0	1	0



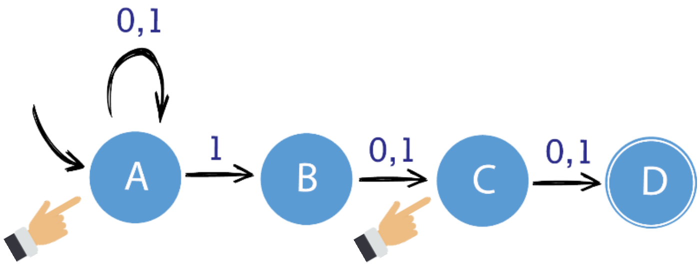
1	0	0	1	0



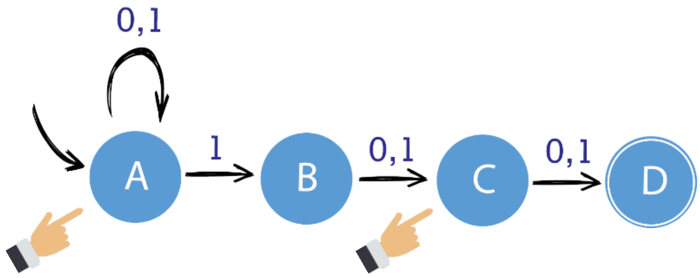
1	0	0	1	0



1	0	0	1	0

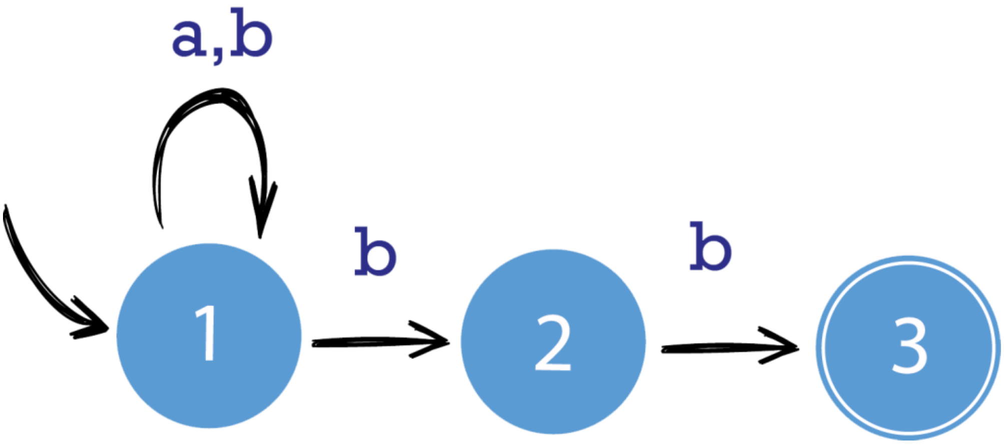


1	0	0	1	0	



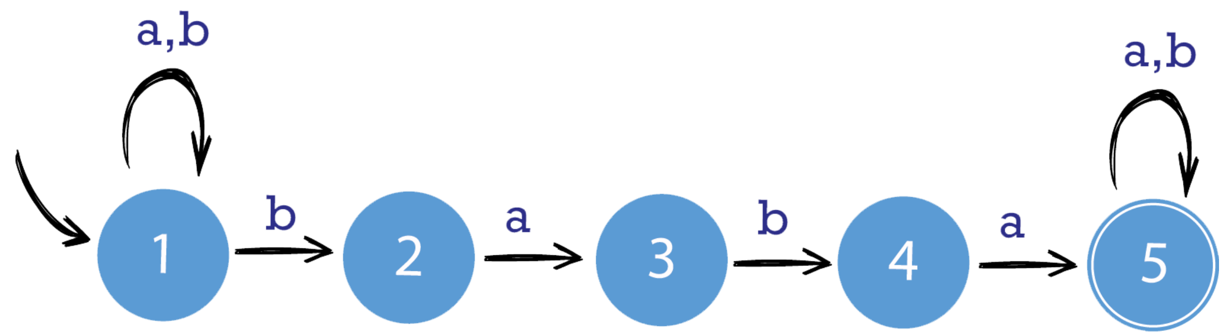
Ejemplo 5

- Sea L el lenguaje sobre Σ = {a, b} de hileras tales que terminen en “bb”
- Diseñe un NFA que reconozca a L.



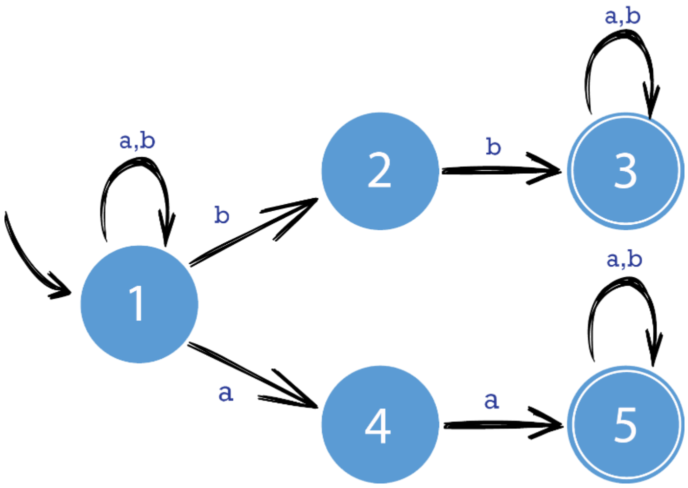
Ejemplo 6

- Sea L el lenguaje sobre Σ = {a, b} de hileras tales que terminen en “baba”
- Diseñe un NFA que reconozca a L.



Ejemplo 7

- Sea L el lenguaje sobre Σ = {a, b} de hileras que contengan las subhileras "aa" o "bb"
- Diseñe un NFA que reconozca a L.



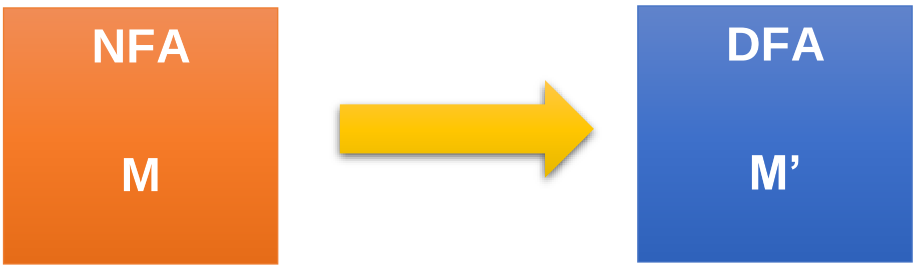
Definición Formal

- Un autómata No Determinístico de Estados Finitos (NFA) es un quinteto M = (Q, Σ, δ, q_0, F), donde:
  - Q es un conjunto finito de **estados**
  - Σ es un **alfabeto**
  - δ: Q x Σ → P(Q) es la **función de transición**
  - $q_0 \in$ Q es el **estado inicial**
  - F ⊆ Q conjunto de **estados de aceptación**

La tercera es el cambio con la definición anterior de autómatas, porque ahora nos dice que con un estado un con un símbolo me puedo mover a varios destinos, antes solo podíamos movernos a uno solo.

Teorema

- Sea M = (Q, Σ, δ, q_0, F) un NFA. Siempre existe un DFA M' = (P(Q), Σ, δ', {q_0}, F') que reconoce exactamente el mismo lenguaje que M.
- Todo NFA tiene un DFA equivalente que reconoce exactamente el mismo lenguaje.



Observaciones

- El alfabeto Σ es el mismo en las dos máquinas
- Los estados de M' son subconjuntos del conjunto Q de M
- Hay que calcular la nueva función de transición δ'
- F' es un conjunto de subconjuntos del conjunto Q de M

Cálculo de Nueva Función de Transición

Esta parte no se va a entender, pero no importa, porque es un enredo hasta que luego veamos un ejemplo y entonces nos iluminemos y todos tenga sentido.

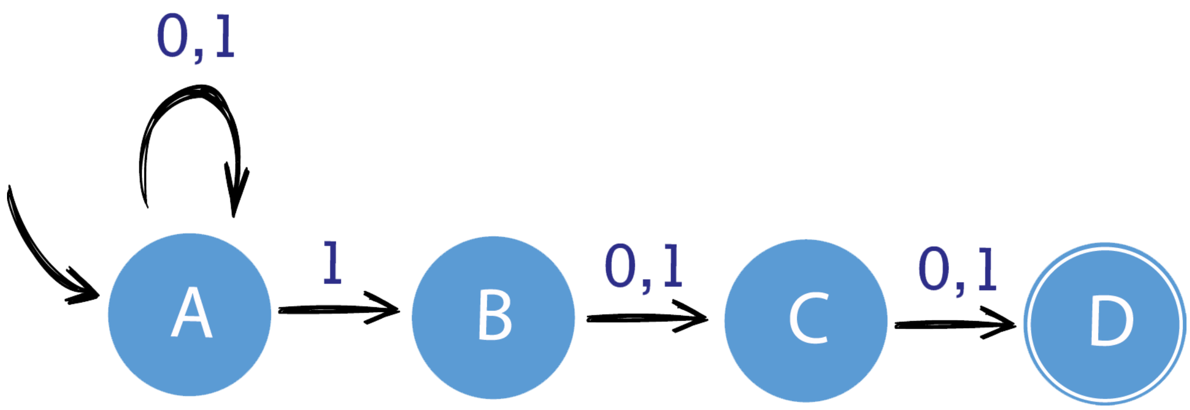
1. El estado inicial de M' es {q_0}, o sea un conjunto que contiene al estado inicial de M, esta es la primera fila de la taba asociada a δ'
2. En la tabla asociada a la función de transición δ' coloque en la columna de todo símbolo de Σ el conjunto de todos los lugares a las que se puede llegar desde q_0 con dicho símbolo (el conjunto vacío es una opción válida)
3. Tome el primero de estos subconjuntos y póngalo como una nueva fila de la tabla de δ'
4. Llene las columnas con el conjunto de todos los destinos a los que se llega con el símbolo de la columna desde todos los miembros de este subconjunto
5. Repita mientras queden subconjuntos pendientes.

Ejemplo

La clave de esto es hacer una función de transición usando una tabla, con dos columnas porque solo acepta 0, 1 y la cantidad de filas es igual a un montón, no se sabe cuántas vamos a hacer exactamente.

Ponemos el estado inicial como hacíamos antes. Pero lo importante aquí es no repetir elementos. Tenemos que revisar a que podemos llegar con 0 y hacer un conjunto con eso, luego a que podemos llegar con 1 y hacer otro conjunto con eso para rellenar la fila. Y empezamos a hacer lo mismo con todos los conjuntos que obtengamos en la respuesta que no hayamos evaluado ya.
Esto nos da la función delta, la función de transiciones.

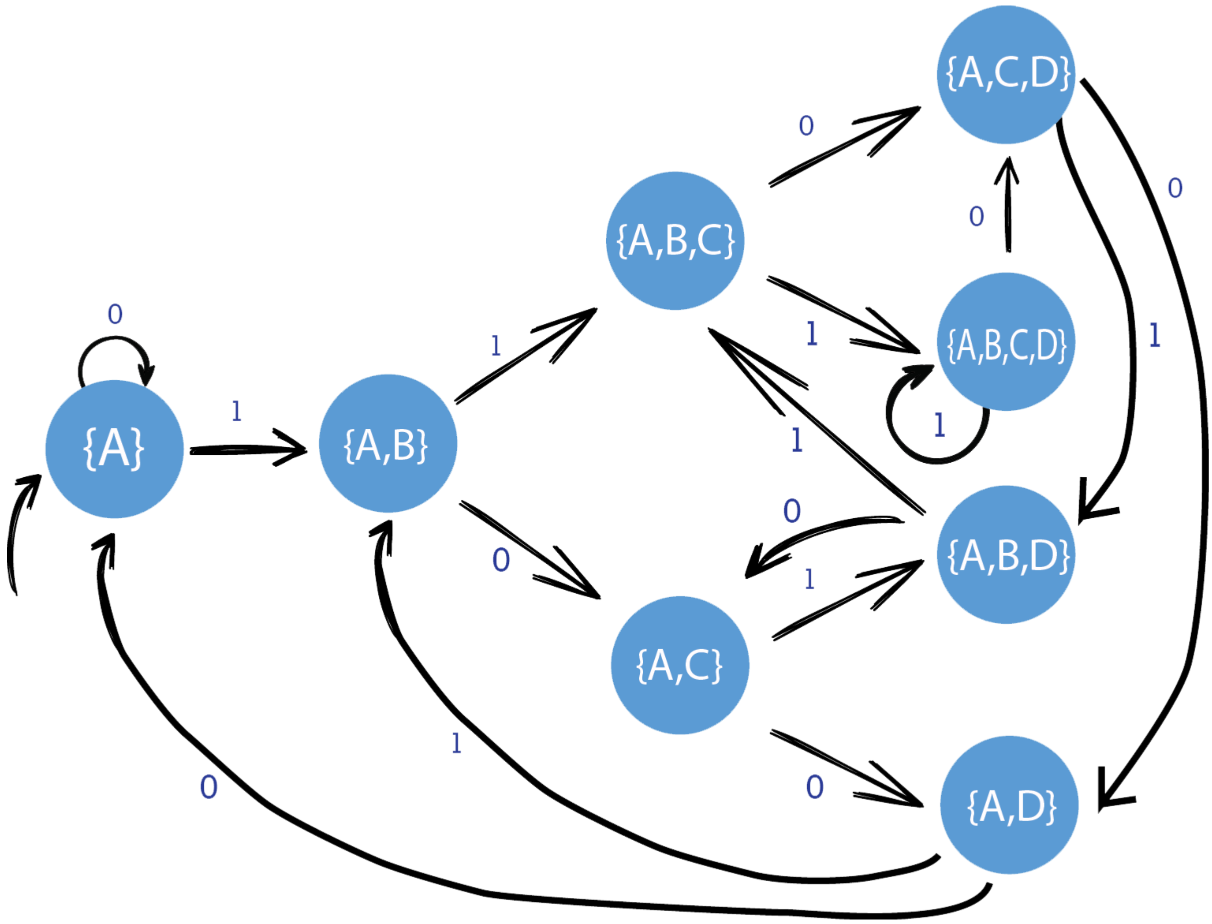
	0	1
{A}	{A}	{A, B}
{A, B}	{A, C}	{A, B, C}
{A, B, C}	{A, C, D}	{A, B, C, D}
{A, C}	{A, D}	{A, B, D}
{A, C, D}	{A, D}	{A, B, D}
{A, B, C, D}	{A, C, D}	{A, B, C, D}
{A, D}	{A}	{A, B}
{A, B, D}	{A,C}	{A, B, C}



Grafo de Transiciones

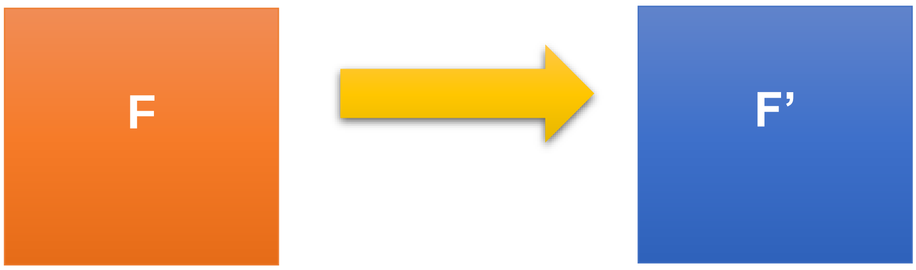
Los estados son igual al número de filas de la tabla, y el límite de estados que podemos tener es igual a 2 a la n, así que si vamos por 45 estados en el examen y era dos a la 5 algo está mal y Chuck lo desaprobaría.

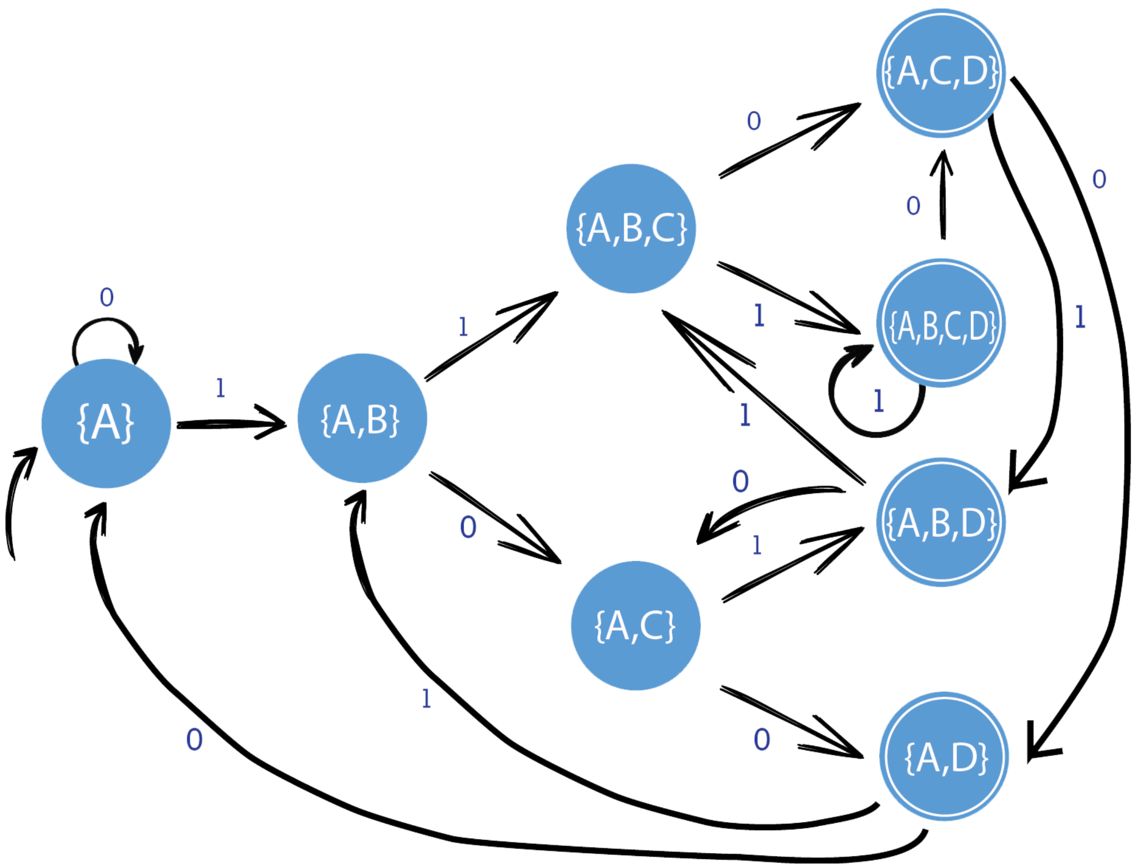
En este caso podemos ver que cualquiera que tenga el estado de aceptación del DFA se acepta, ósea los que tenga D son de aceptación.



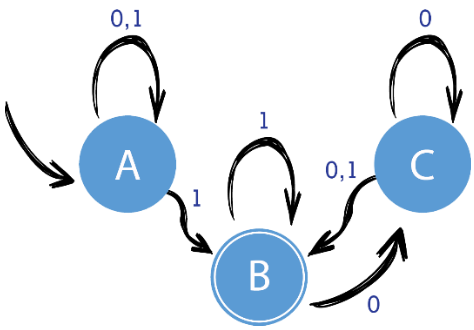
Estados de aceptación

Cualquier estado en M’ que contenga al menos un elemento que sea miembro de F es miembro de F’

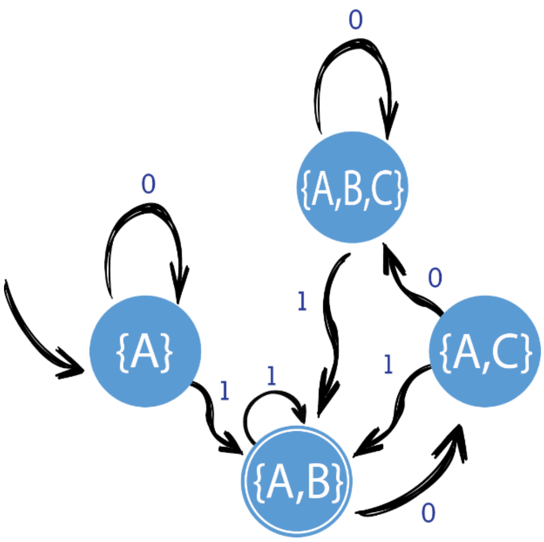




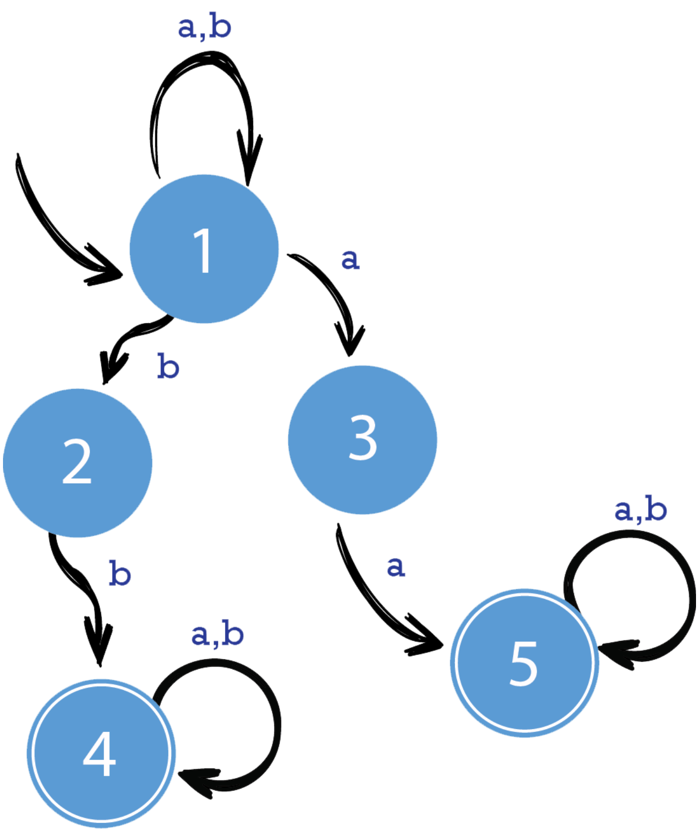
Ejemplo 2



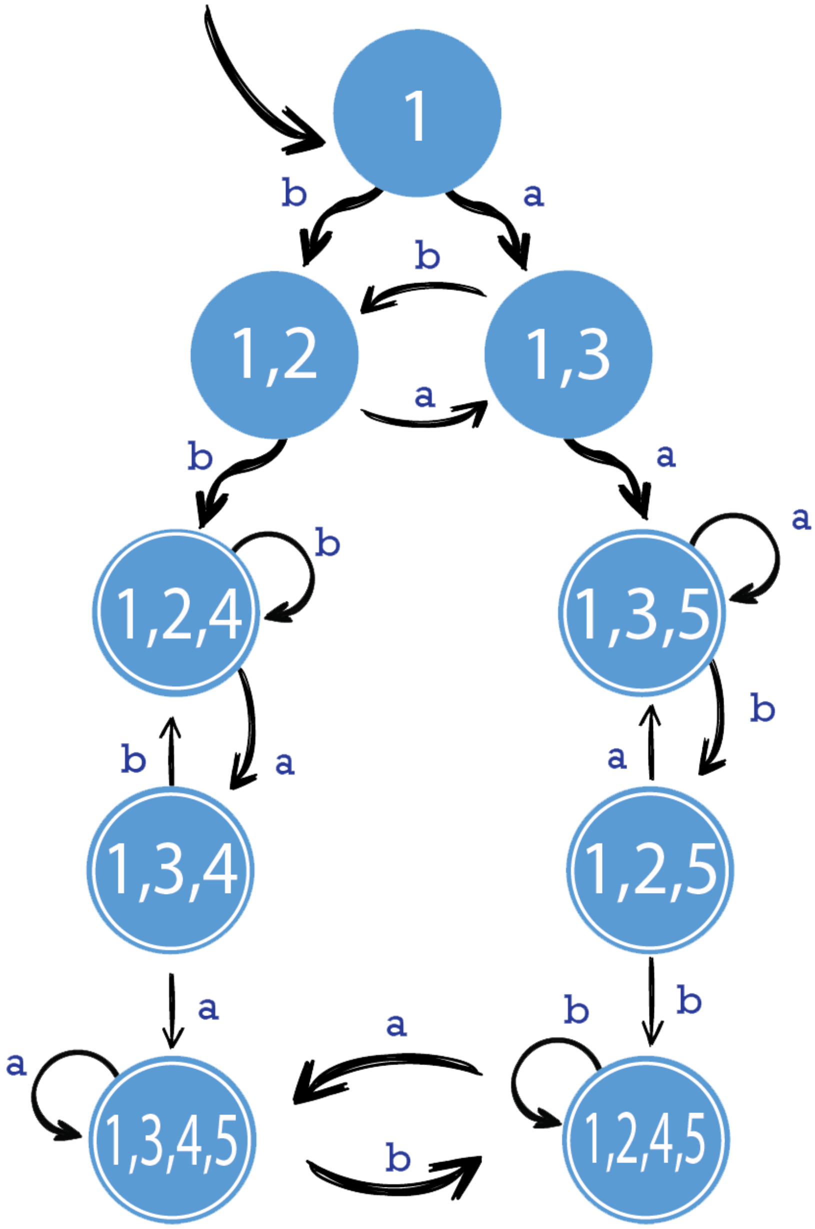
	0	1
{A}	{A}	{A, B}
{A, B}	{A, C}	{A, B}
{A, C}	{A, B, C}	{A, B}
{A, B, C}	{A, B, C}	{A, B}



Ejemplo 3



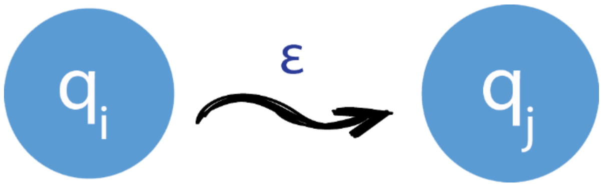
	a	b
{1}	{1, 3}	{1, 2}
{1, 3}	{1, 3, 5}	{1, 2}
{1, 2}	{1, 3}	{1, 2, 4}
{1, 3, 5}	{1, 3, 5}	{1, 2, 5}
{1, 2, 4}	{1, 3, 4}	{1, 2, 4}
{1, 2, 5}	{1, 3, 5}	{1, 2, 4, 5}
{1, 3, 4}	{1, 3, 4, 5}	{1, 2, 4}
{1, 2, 4, 5}	{1, 3, 4, 5}	{1, 2, 4, 5}
{1, 3, 4, 5}	{1, 3, 4, 5}	{1, 2, 4, 5}



Y listo, ya sabemos convertir un dfa en un nfa pero como eso sería muy fácil vamos a ver otra cosa para embarrialar más la cancha.

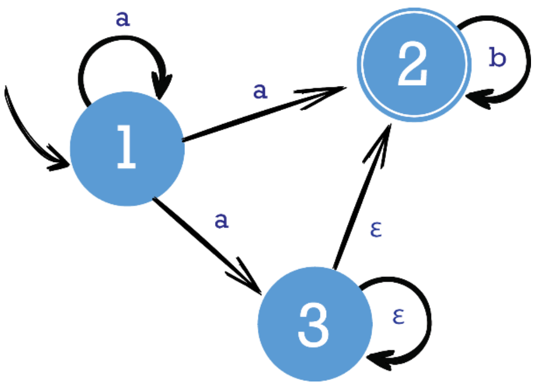
Transiciones ε (épsilon)

- Hasta ahora, en cada transición se consume un símbolo de la hilera y se avanza al siguiente estado.
- Existen autómatas que permiten ir desde ciertos estados a otros… sin consumir símbolo
- Conectamos dos estados con una transición etiquetada "ε"
  - Transiciones ε



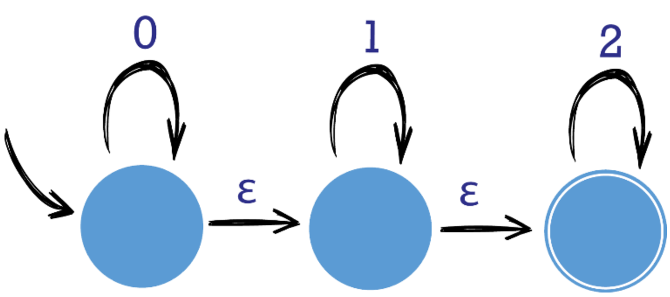
Ejemplo 1

- Sea L el lenguaje sobre Σ = {a, b, c} de hileras que empiecen con una serie de una o más "a", seguida de cero o más "c" y terminada en cero o más "b".
- Diseñe un NFA-ε que reconozca a L.



Ejemplo 2

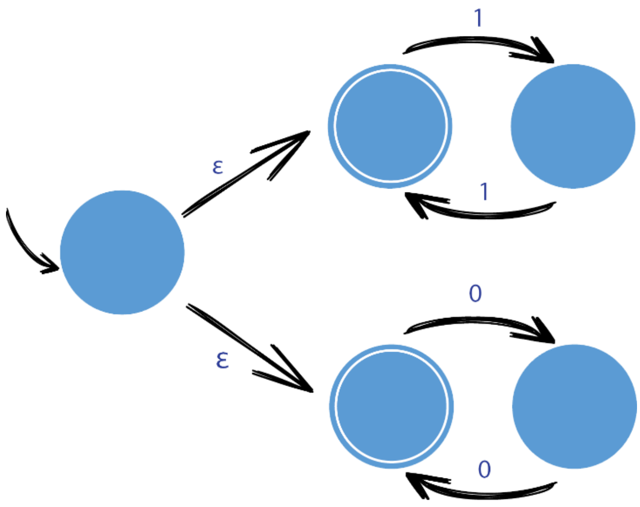
- Sea L el lenguaje sobre Σ = {0, 1, 2} de hileras que empiecen con cualquier número de 0's, seguidos por cualquier número de 1's y terminada en cualquier número de 2's.
- Diseñe un NFA-ε que reconozca a L.



En este grafo podemos ver que por ejemplo se acepta la hilera 1,1. Pero no se acepta la hilera 0,2,1.

Ejemplo 3

- Sea L el lenguaje sobre Σ = {0,1} de hileras de longitud par con un único tipo de símbolo.
- Diseñe un NFA-ε que reconozca a L.



Este grafo es como un OR, con la unión de dos autómatas que verifiquen si es 000 o 111 pares. Por ejemplo, 1, 0, 0 no es aceptada.

Ahora podemos hacer un nfa-épsilon a dfa o si preferimos pasar a un nfa y luego al dfa.