2. Autómatas finitos

2.1. Autómatas Finitos Deterministas (AFD)

Fernando Rosa Velardo

Traducción y adaptación de transparencias de Ananth Kalyanaraman
(http://www.eecs.wsu.edu/~ananth/)

Autómata Finito (AF)

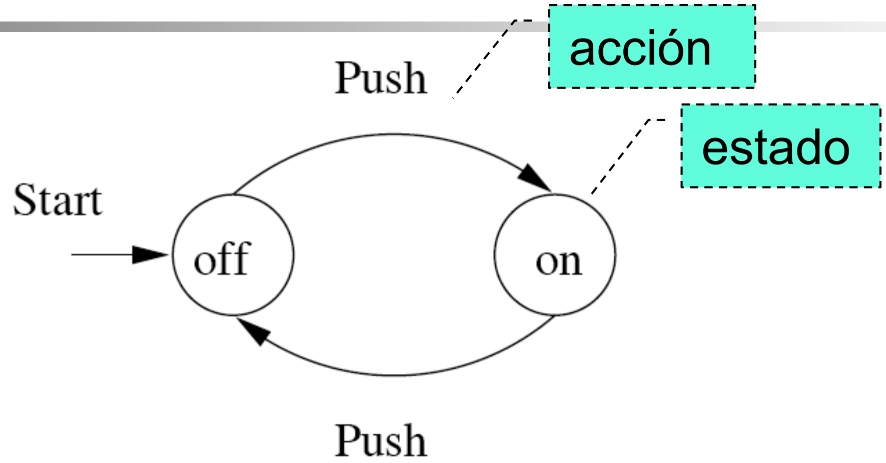
- Informalmente, diagrama de estados que captura de manera exhaustiva todos los posibles estados y transiciones que una máquina puede tomar como respuesta a una secuencia de símbolos de entrada

- Autómatas Finitos Deterministas (AFD)
  - El siguiente estado está determinado por el estado actual y por el símbolo de entrada

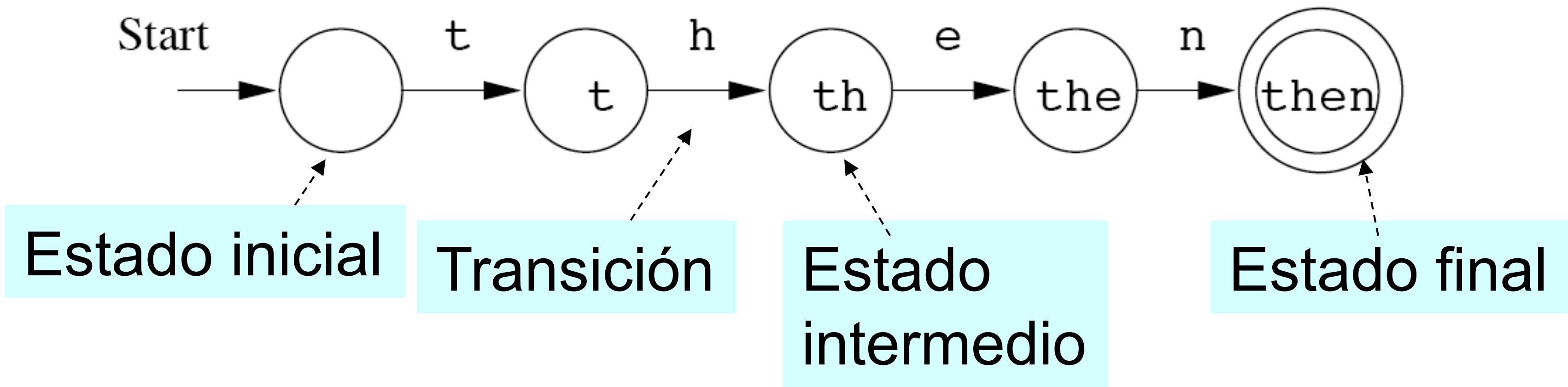
- Automata Finito No-determinista (AFN)
  - Desde un estado, leyendo un símbolo de entrada, es posible pasar a varios estados distintos

Autómatas Finitos: Ejemplos

- Interruptor



- Reconocimiento de la palabra “*then*”



Autómata Finito Determinista - Definición

- Un Autómata Finito Determinista (AFD) viene dado por:
  - Q ==> conjunto finito (y no vacío) de estados
  - ∑ ==> alfabeto
  - q_0 ==> estado inicial
  - F ==> conjunto de estados finales
  - δ ==> función de transición

$$\delta : Q \times \Sigma \rightarrow Q$$

- Un AFD es una tupla:
  - $(Q, \Sigma , q_0, F, \delta)$

¿Qué hace un AFD al leer su entrada?

- <u>Input:</u> palabra w en Σ^*
- <u>Ouput:</u> Acepta el AFD w?
- <u>Pasos:</u>
  - Comienza en el estado inicial q_0
  - Para cada símbolo en w
    - Calcula el próximo estado, a partir del estado actual y del símbolo actual, usando la función de transición
  - Si al consumir todos los símbolos de w el estado actual es final (F) entonces *acepta w;*
  - Si no, *rechaza w.*

Lenguajes regulares

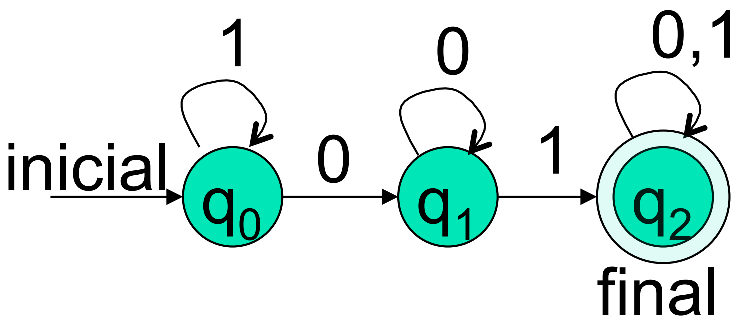
- L(A) conjunto de palabras que acepta o reconoce A
  - A cualquier lenguaje aceptado por un AFD se le llama “*Lenguaje Regular* ”.

Ejemplo #1

- Diseña un AFD para el lenguaje:
  - L = {w | w es una palabra sobre el alfabeto binario que contiene 01 como subpalabra}
- Pasos para diseñar un AFD que acepta L:
  - $\Sigma = \{0,1\}$
  - Decidir conjunto de estados: Q
  - Designar estados inicial y final(es)
  - δ: Decidir las transiciones:
- Estados Finales == estados de aceptación
- Otros estados == estados de no aceptación o de rechazo

AFD para palabras que contienen 01

- ¿Por qué es este AFD determinista?



- ¿Y si le añadimos al lenguaje la palabra vacía?

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0,1\}$
- estado inicial = q_0
- $F = \{q_2\}$
- Tabla de transiciones

símbolos

δ	**0**	**1**
→ **q_0**	q_1	q_0
q_1	q_1	q_2
***q_2**	q_2	q_2

estados

Ejemplo #2

- Diseña un AFD para el lenguaje:

  L = { w | w sobre el alfabeto binario con
  una cantidad par de 0s y 1s}

- ?

Función de transición extendida

- $\hat{\delta}$ *(q ,w)* = estado que alcanza el autómata desde q al leer *w*

- $\hat{\delta}(q\ ,\ \varepsilon\ ) = q$
- $\hat{\delta}(q\ ,wa) = \delta\ (\hat{\delta}(q\ ,w),\ a)$
  - ¿Qué ocurre en el ejemplo #2 tomando w=100 y a=1:
    - $\hat{\delta}(q_0,wa) = ?$

Lenguaje de un AFD

Un AFD A acepta w si hay un camino desde q_0 a un estado final etiquetado por *w*

- *es decir,* $L(A) = \{ w \mid \hat{\delta}(q_0, w) \in F \}$

2. Autómatas finitos

2.2. Autómatas Finitos No-Deterministas (AFN)

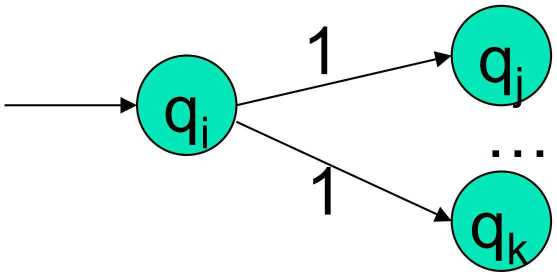
Fernando Rosa Velardo

Traducción y adaptación de transparencias de Ananth Kalyanaraman
(http://www.eecs.wsu.edu/~ananth/)

Autómata Finito No-determinista (AFN)

- Un Autómata Finito No-determinista (AFN)
  - es, por supuesto, "no-determinista"
    - En un estado, ante cierto símbolo, puede tener que elegir entre distintos caminos
    - La intuición es que el AFN sabe elegir bien

No "tiene que", sólo puede



- La función de transición asigna a cada estado y símbolo un conjunto de estados

Autómata Finito No-determinista - definición

- Un Autómata Finito No-determinista (AFN) viene dado por:
  - Q ==> conjunto finito (y no vacío) de estados
  - ∑ ==> alfabeto
  - q_0 ==> estado inicial
  - F ==> conjunto de estados finales
  - δ ==> función de transición

    $\delta : Q \times \Sigma \rightarrow$ subconjuntos de Q $= \wp(Q)$
- Un AFN es una tupla:
  - $(Q, \Sigma, q_0, F, \delta)$

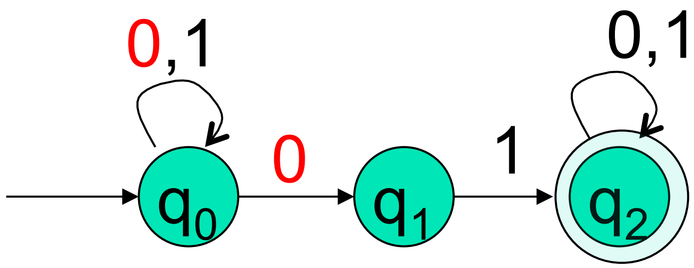
¿Cómo se ejecuta un AFN?

- Input: palabra w en Σ^*
- Output: ¿Acepta el AFN w?
- Pasos:
  - Empieza en el "estado inicial" q_0
  - Para cada símbolo en w
    - Dado el estado actual y el símbolo actual de w, elige según la función de transición algún estado como estado siguiente. Si no hay ninguno, rechaza w en esta ejecución
  - Si después de leer todos los símbolos de w, el AFN está en un estado de aceptación (F), *acepta w en esta ejecución;*
  - Si no, *rechaza w en esta ejecución.*

  - *w es aceptada por el AFN si lo es en alguna ejecución*

AFN para palabras que contienen 01

¿Por qué es no-determinista?



¿Qué pasa si se recibe un 0 en q_1?

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0,1\}$
- estado inicial = q_0
- $F = \{q_2\}$
- Tabla de transiciones

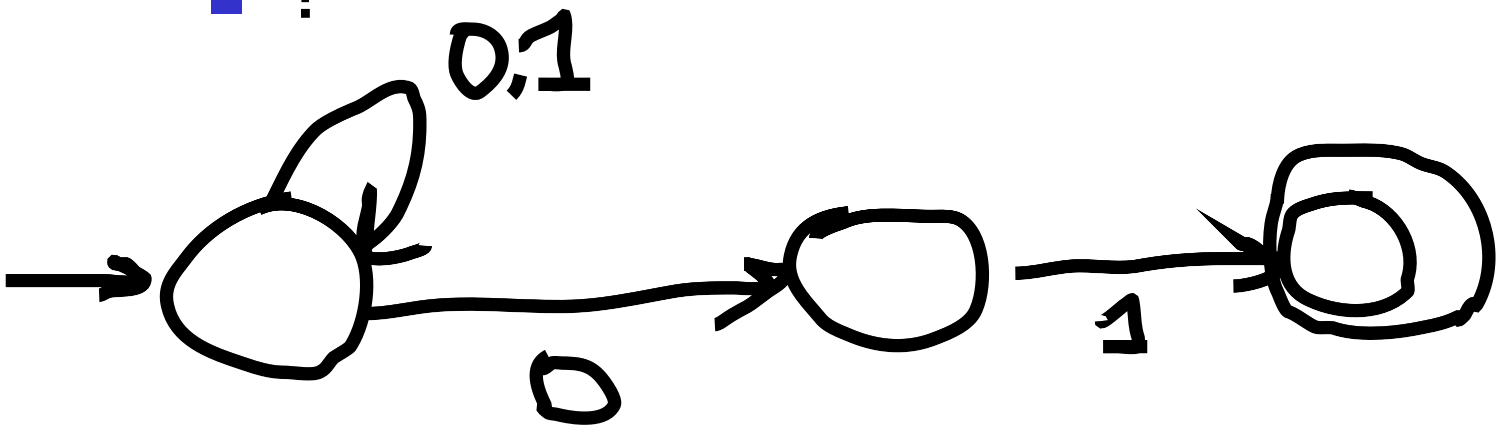
símbolos

estados

δ	**0**	**1**
→ **q_0**	$\{q_0,q_1\}$	$\{q_0\}$
q_1	Φ	$\{q_2\}$
***q_2**	$\{q_2\}$	$\{q_2\}$

Ejemplo #3

- Diseña un AFN para el lenguaje
  L = { w | w acaba en 01}
- ?



Función de transición de un AFN extendida

- Base: $\hat{\delta}(q,\varepsilon) = \{q\}$

- Inducción:
  - Sea $\hat{\delta}(q,w) = \{p_1,p_2\ldots,p_k\}$
  - $\delta(p_i,a) = S_i$ para $i=1,2\ldots,k$
  - *Entonces,* $\hat{\delta}(q,wa) = S_1 \cup S_2 \cup \ldots \cup S_k$

Lenguaje de un AFN

- Un AFN N acepta *w* si existe algún camino del estado inicial a algún estado final etiquetado por *w, es decir*

- $L(N) = \{ w \mid \hat{\delta}(q_0,w) \cap F \neq \Phi \}$

Ventajas y desventajas de los AFN

- Potencia del no-determinismo
- Pero “imaginarios”, en el sentido de que en la práctica han de implementarse de manera determinista

Diferencias: AFD vs. AFN

- **AFD**
1. Todas las transiciones son deterministas
   - Cada transición lleva a un único estado
2. La función de transición ha de estar definida para cada estado y símbolo
3. Acepta el input si el último estado está en F
4. A veces, más difícil de construir por el número de estados
5. Implementación factible

- **AFN**
1. Algunas transiciones pueden ser no-deterministas
   - Una transición puede llevar a un conjunto de estados
2. La función de transición no ha de estar definida para cada estado y símbolo
3. Acepta el input si *alguno* de los últimos estados está en F
4. En general, más fácil de construir que un AFD
5. La implementación ha de ser determinista (hay que convertirlo a AFD)

Sin embargo, ¡los AFD y los AFN son equivalentes!

2. Autómatas finitos

2.3. Equivalencia entre AFD y AFN

Fernando Rosa Velardo

Traducción y adaptación de transparencias de Ananth Kalyanaraman
(http://www.eecs.wsu.edu/~ananth/)

Equivalencia entre AFDs y AFNs

Ha de ser cierto para todo L

- <u>Teorema</u>: Para cualquier lenguaje L

  L aceptado por un AFD ⇔ L aceptado por un AFN

- <u>Demostración</u>:

1. =>
   - Todo AFD es un AFN para el que cada estado tiene exactamente una transición para cada símbolo. Por lo tanto, si L es aceptado por un AFD, es aceptado por el correspondiente AFN.
2. <=:
   - Hay que probar que para cada AFN existe un AFD equivalente, es decir, que acepta el mismo lenguaje (en las próximas transparencias…)

□

Demostración de <=

- <=: Si L es aceptado por un AFN también es aceptado por algún AFD
- En otras palabras…
- Dado un AFN N, podemos construir un AFD D tal que L(N)=L(D)

- ¿Cómo podemos convertir un AFN en AFD?
  - Observación: cada transición del AFN devuelve un *subconjunto* de estados
  - Idea: Consideramos un "estado" en el AFD para cada posible subconjunto de estados del AFN

Construcción de subconjuntos

De AFN a AFD: construcción de subconjuntos

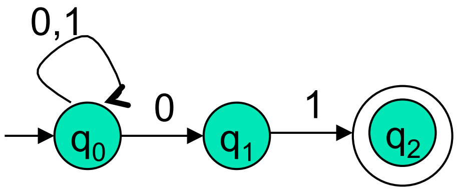
- Sea N = $(Q_N,\Sigma,\delta_N,q_0,F_N)$
- Objetivo: Construir D=$(Q_D,\Sigma,\delta_D,\{q_0\},F_D)$ tal que L(D)=L(N)
- Construcción:
  1. Q_D= subconjuntos de Q_N (conjunto potencia)
  2. F_D= subconjuntos S de Q_N tales que $S \cap F_N \neq \Phi$
  3. δ_D: para cada subconjunto S de Q_N y para cada símbolo a de Σ:
     - $\delta_D(S,a) = \bigcup_{p \text{ en } s} \delta_N(p,a)$

Idea: para evitar tener que enumerar todos los subconjuntos, creamos los estados de manera "perezosa"

Construcción de subconjuntos: ejemplo

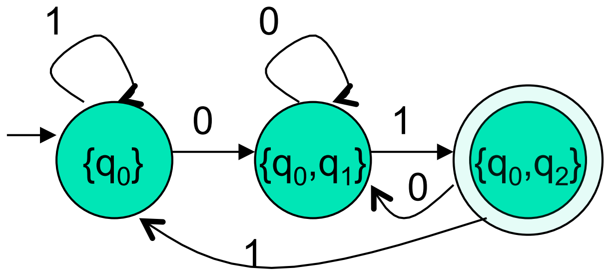
- *L = {w | w acaba en 01}*

AFN:



δ_N	0	1
→q_0	$\{q_0,q_1\}$	$\{q_0\}$
q_1	∅	$\{q_2\}$
$*q_2$	∅	∅

AFD:



δ_D
~~∅~~
→$\{q_0\}$
~~$\{q_1\}$~~
~~$*\{q_2\}$~~
$\{q_0,q_1\}$
$*\{q_0,q_2\}$
~~$*\{q_1,q_2\}$~~
~~$*\{q_0,q_1,q_2\}$~~

δ_D	0	1
→$\{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\{q_0,q_1\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$
$*\{q_0,q_2\}$	$\{q_0,q_1\}$	$\{q_0\}$

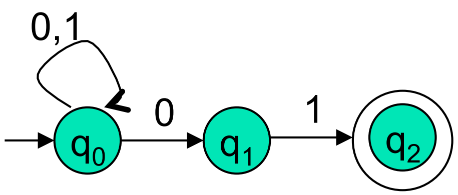
1. Determina las transiciones

2. Conserva sólo aquellos estados alcanzables desde $\{q_0\}$

Mismo ejemplo, con PEREZA

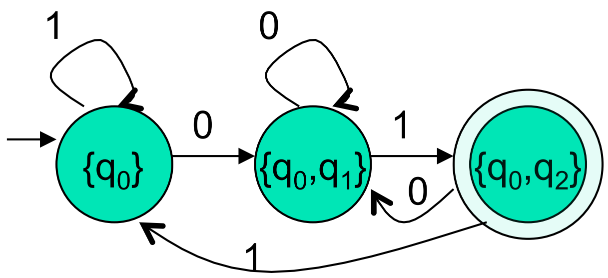
- *L = {w | w acaba en 01}*

AFN:



δ_N	0	1
$\rightarrow q_0$	$\{q_0,q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$

AFD:



δ_D	0	1
$\rightarrow \{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$

Idea principal:

Considerar estados conforme se vayan necesitando

Corrección de la construcción de subconjuntos

Teorema: *Si D es el AFD construido a partir del AFN N usando la construcción de subconjuntos, entonces L(D)=L(N)*

- Demostración:
  - Basta probar que para cada w se tiene

$$\hat{\delta}_D(\{q_0\},w) \equiv \hat{\delta}_N(q_0,w)$$

  - Se prueba mediante inducción sobre la longitud de w

2. Autómatas finitos

2.4. Autómatas Finitos No-Deterministas con transiciones ε (ε -AFN)

Fernando Rosa Velardo

Traducción y adaptación de transparencias de Ananth Kalyanaraman (http://www.eecs.wsu.edu/~ananth/)

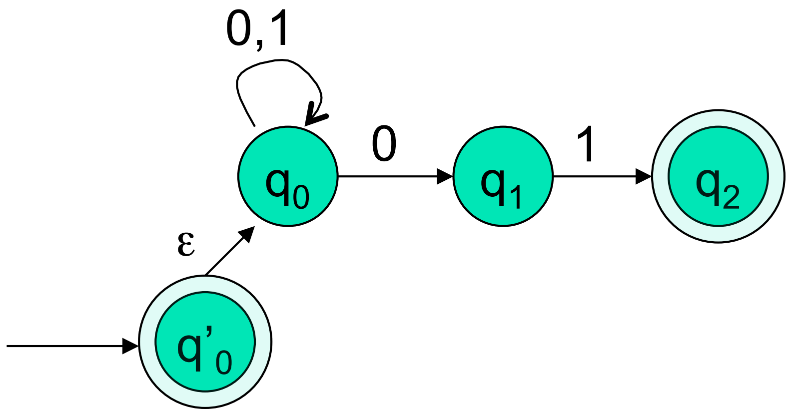
AFN con transiciones ε

- Extendemos los AFN con transiciones ε
  - es decir, el AFN puede saltar de un estado a otro sin consumir ningún símbolo de entrada
- <u>Objetivo:</u>
  - Así es más sencillo diseñar AFN
- ε-AFN tienen una columna más en su tabla de transiciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \text{subconjuntos de } Q$$

Ejemplo de ε-AFN

L = {w | w es vacía, o acaba en 01}

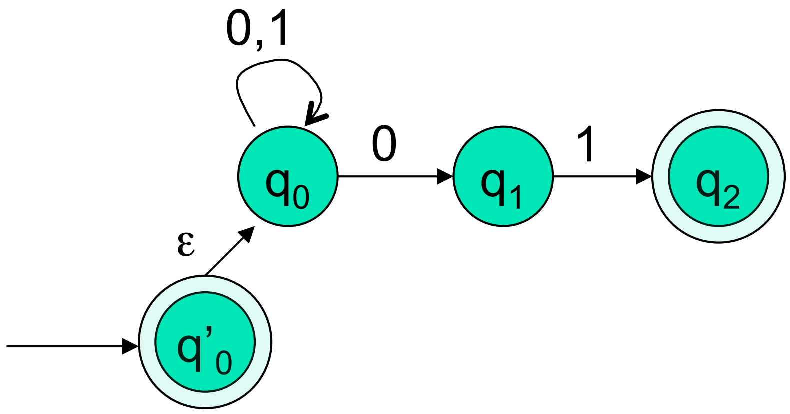


δ_E	0	1	ε
→ $*q'_0$	Ø	Ø	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$	Ø
q_1	Ø	$\{q_2\}$	Ø
$*q_2$	Ø	Ø	Ø

- ε-clausura de un estado q, ***CLAUS(q)***, es el conjunto de todos los estados (incluido q) que se pueden alcanzar desde q siguiendo cualquier número de transiciones ε.

Ejemplo de ε-AFN

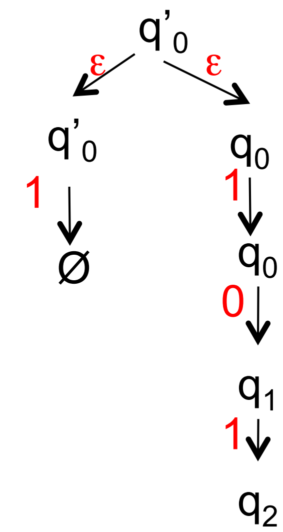
L = {w | w es vacía, o acaba en 01}



δ_E	0	1	ε
→ $*q'_0$	Ø	Ø	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$	Ø
q_1	Ø	$\{q_2\}$	Ø
$*q_2$	Ø	Ø	Ø

<u>Ejecuciones para w=101:</u>

$CLAUS(q'_0)=\{q_0,q'_0\}$



Clausura de un estado - Definición

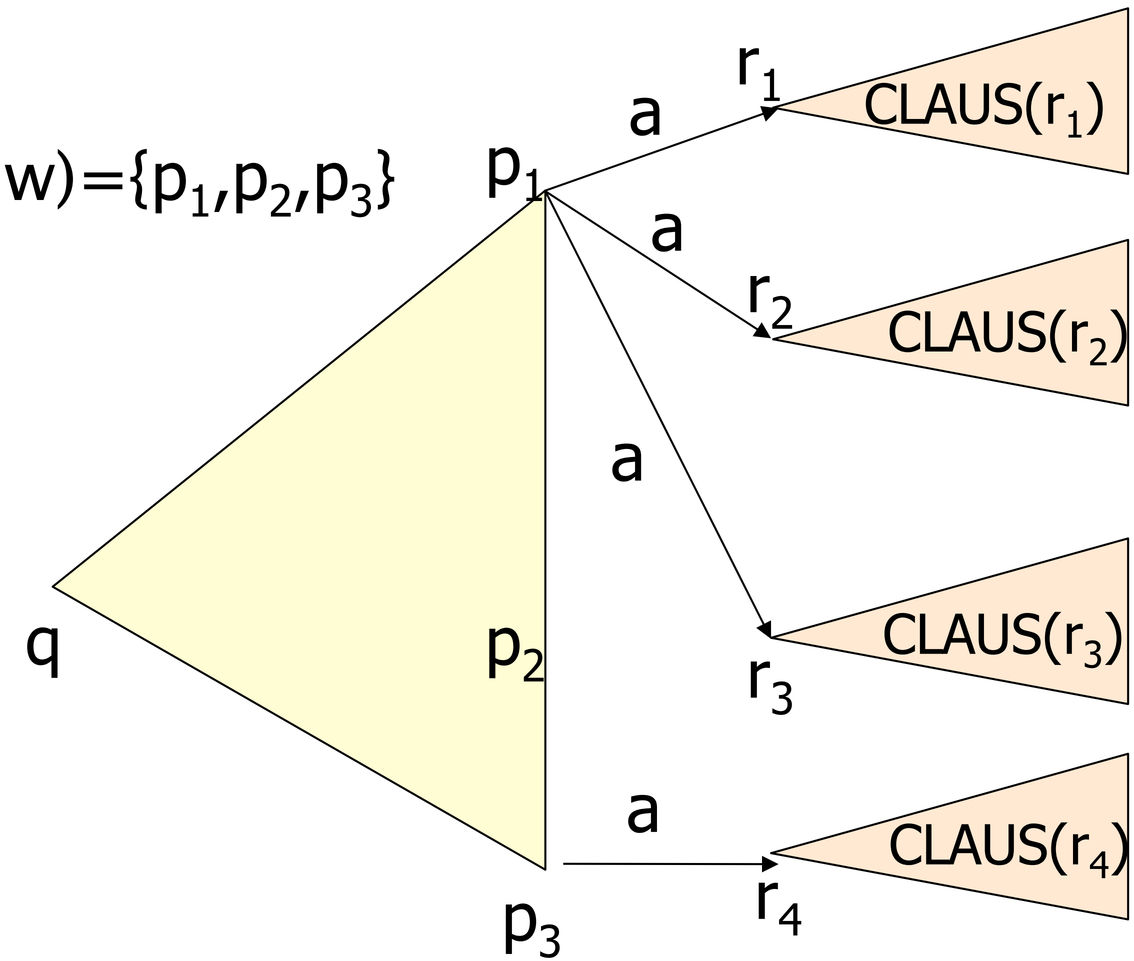
- Base: *q pertenece a CLAUS(q)*

- Inducción:
  - Si p pertenece a CLAUS(q)
  - y r pertenece a $\delta(p, \varepsilon)$

  - *Entonces r pertenece a CLAUS(q)*

Función de transición de un ε- AFN extendida: $\hat{\delta}(q,wa)=?$

$\hat{\delta}(q,w)=\{p_1,p_2,p_3\}$



Función de transición de un ε- AFN extendida

- <u>Base:</u> $\hat{\delta}\ (q,\varepsilon) = CLAUS(q)$

- <u>Inducción:</u>
  - Sea $\hat{\delta}\ (q\ ,w) = \{p_1,p_2 \ldots,p_k\}$
  - $\delta\ (p_1,a)\ U\ \delta(p_2,a)\ U\ \ldots\ U\ \delta\ (p_k,a)\ = \{r_1,r_2 \ldots,r_m\}$
  - *Entonces,*

$$\hat{\delta}\ (q\ ,wa)= CLAUS(r_1)U \ldots\ U\ CLAUS(\ r_m\ )$$

Lenguaje de un ε- AFN

- Un ε-AFN E acepta *w* si existe algún camino del estado inicial a algún estado final etiquetado por *w* (posiblemente con ε intermedios), *es decir*

- $L(E) = \{ w \mid \hat{\delta}(q_0,w) \cap F \neq \Phi \}$

2. Autómatas finitos

2.5. Equivalencia de los AFN y los ε− AFN (eliminación de transiciones ε)

Fernando Rosa Velardo

Traducción y adaptación de transparencias de Ananth Kalyanaraman (http://www.eecs.wsu.edu/~ananth/)

Equivalencia entre AFD, AFN y ε-AFN

- Teorema: L es el lenguaje aceptado por algún ε-AFN si y solo si L es el lenguaje aceptado por algún AFN

- Consecuencia:
  - AFD ≡ AFN ≡ ε-AFN
  - (todos aceptan Lenguajes Regulares)

Eliminación de transiciones ε

- Sea $E = (Q,\Sigma,\delta_E,q_0,F_E)$ un ε−AFN
- Objetivo: Construir un AFN $N=($$Q,\Sigma,\delta_N,q_0,F_N$$)$ tal que $L(N)=L(E)$
- Construcción:

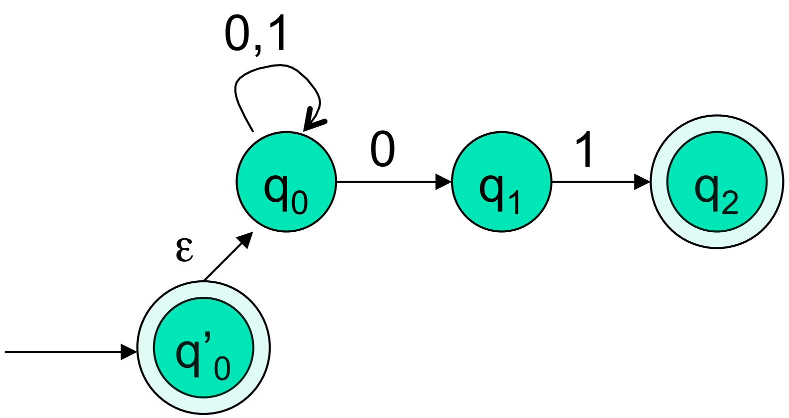
1. $F_N=\{ q \mid CLAUS(q)\cap F_E\neq\Phi \}$

2. $\delta_N(q,a)= \bigcup_{p\in CLAUS(q)} \delta_E(p,a)$

¡¡OJO: Distinto al libro!!

ε-AFN ➔ AFN (con pereza)

L = {w | w es vacía o acaba en 01}

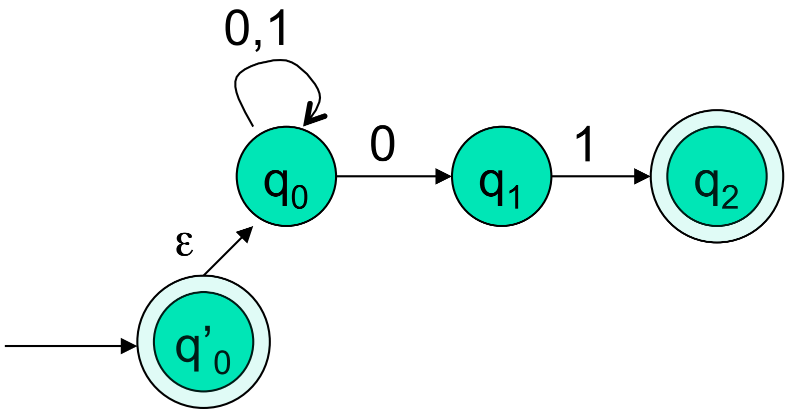


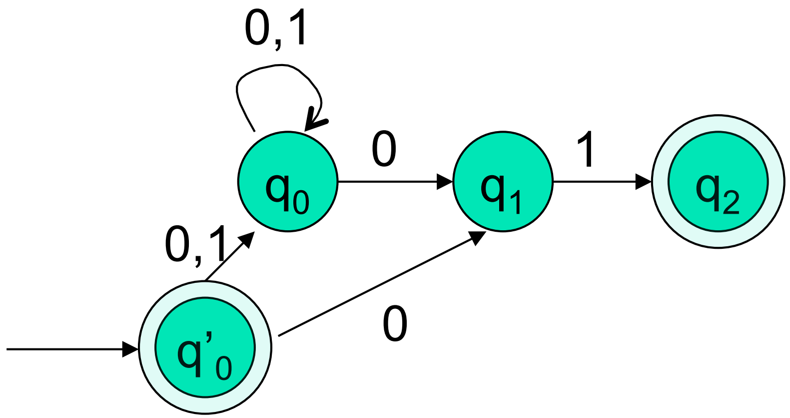
δ_E	0	1	ε
→ $*q'_0$	Ø	Ø	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$	Ø
q_1	Ø	$\{q_2\}$	Ø
$*q_2$	Ø	Ø	Ø

δ_N	0	1
→ $*q'_0$		
…		

ε-AFN ➔ AFN (con pereza)

L = {w | w es vacía o acaba en 01}





δ_E	0	1	ε
→ *q'_0	∅	∅	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$	∅
q_1	∅	$\{q_2\}$	∅
*q_2	∅	∅	∅

δ_N	0	1
→ *q'_0	$\{q_0,q_1\}$	$\{q_0\}$
q_0	$\{q_0,q_1\}$	$\{q_0\}$
q_1	∅	$\{q_2\}$
*q_2	∅	∅