Autómata finita determinísticos y no determinísticos: Equivalencia

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Equivalencia entre un AFND y un AFD

Dos máquinas son equivalentes si reconocen el mismo lenguaje

Para todo string de entrada el resultado del cómputo es el mismo.

Ambas aceptan o rechazan las mismas cadenas

Los autómatas finitos deterministas

son un caso especial de los AFND

Donde se limita a un único estado por transición

Poder de computación

¿Cada máquina no determinística, tiene un equivalente determinístico?

¿O existen lenguajes que solo reconoce uno de ellos?

¿Podemos demostrar que cualquier AFND se puede construir con un AFD?

En ese caso ambos modelos de computación son equivalentes

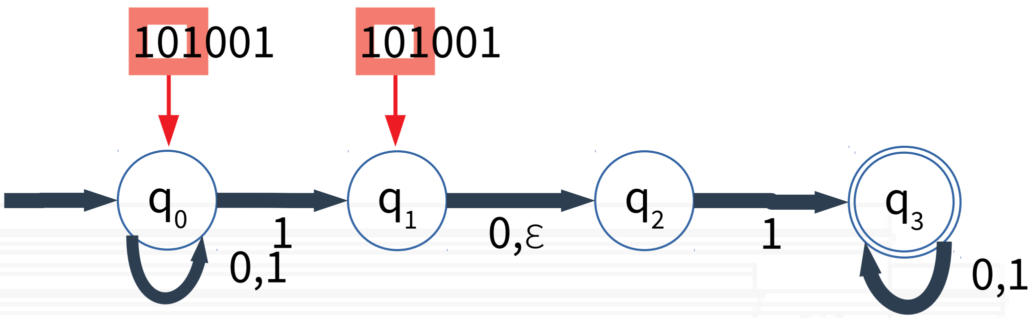
¿ Podemos demostrar que al menos existe un AFND que no se puede construir con un AFD?

En ese caso AFND tienen más poder de computación que las AFD

Ramificaciones de un AFND

Sea

una AFND de n estados,



A medida que se computa un String

Se van ramificando las ejecuciones según la función de transición.

Cada ramificación es

como un puntero a un estado dentro de la maquina.

Ramificaciones de un AFND (cont.)

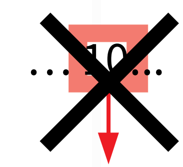
Un AF no tiene memoria

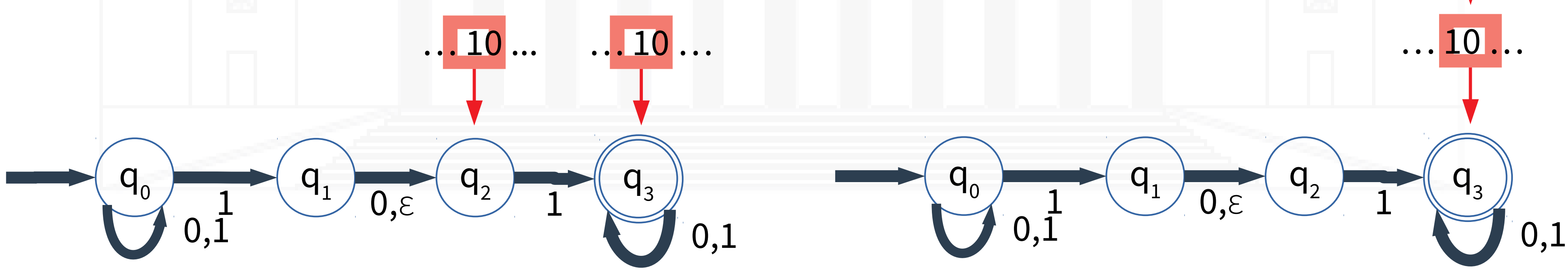
se ejecuta según el símbolo leído y el estado actual

Si dos ramificaciones luego de una transición

coinciden en un mismo estado, se fusionan en solo 1.

La historia de como llegaron no impacta en como se comportaran a partir de ese momento





Ramificaciones de un AFND (cont.)

Existe un máximo de ramificaciones posibles?

Si!, no pueden existir más variantes que todas las combinaciones de estados

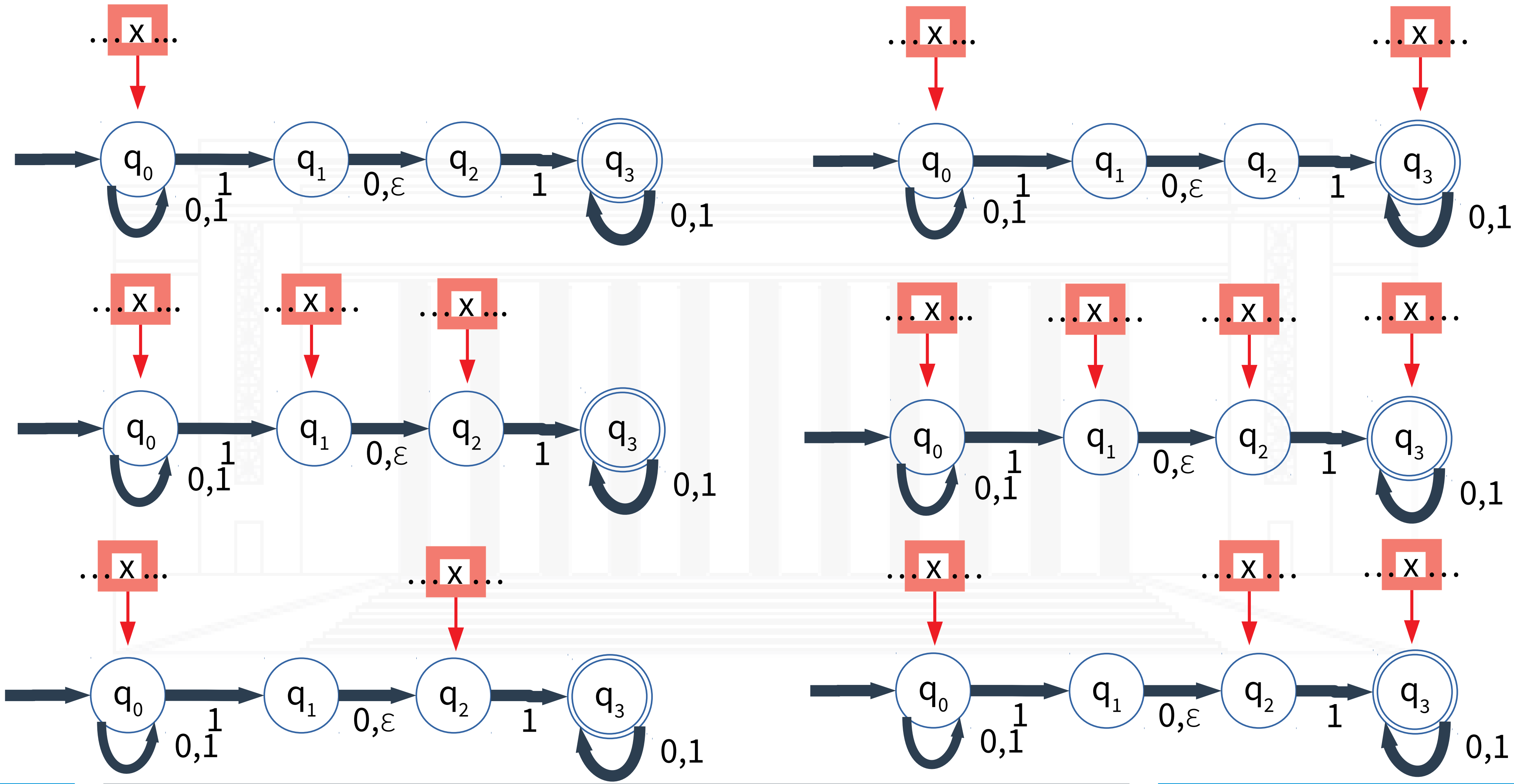
Una AFND de n estados

Tiene un máximo de 2^n combinaciones máximas de posibles “punteros a estados” simultaneos

(Algunas de estas pueden no exisitr para ningún proceso de string particular de esa AFND)

Ejemplo

Existen 2^4 combinaciones de ramificaciones como máximo, pero solo las 6 siguientes son posibles



Equivalencia entre un AFND y un AFD (cont.)

Sea una AFND de "n" estados, construimos un AFD con:

Q: todas las combinaciones posibles de ramificaciones: 2^k

Σ: alfabeto de la AFND

$\delta: Q x \Sigma \to Q$ la transición de un estado de una ramificación a otra (única)

$q_0 \in Q$ "ramificación" inicial de la AFND

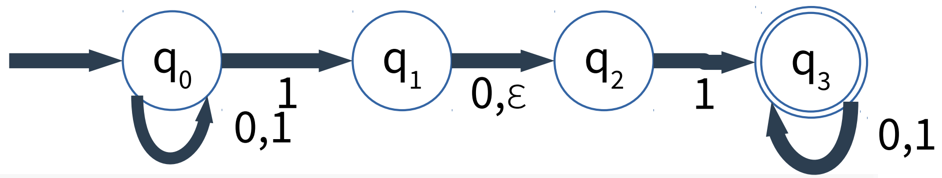
F: cada ramificación posible que tenga al menos 1 estado de finalización.

Este nuevo AFD es equivalente a la AFND

cumple las características de la AFD.

El poder de "expresión" entre AFD y AFND es <u>equivalente!</u>

Ejemplo



Para la AFND de la figura

El alfabeto

Sera el mismo: {0,1}

Creamos los “meta” estados

$\{q_0\}, \{q_0,q_1\}, \{q_0,q_2\}, \{q_0,q_3\}, \{q_0,q_1,q_2\}, \{q_0,q_2,q_3\}, \{q_0,q_1,q_3\}, \{q_0,q_1,q_2,q_3\}, \{q_1\}, \{q_1,q_2\}, \{q_1,q_3\}, \{q_1,q_2,q_3\}, \{q_2\}, \{q_2,q_3\}, \{q_3\},\varnothing$

Será el estado inicial

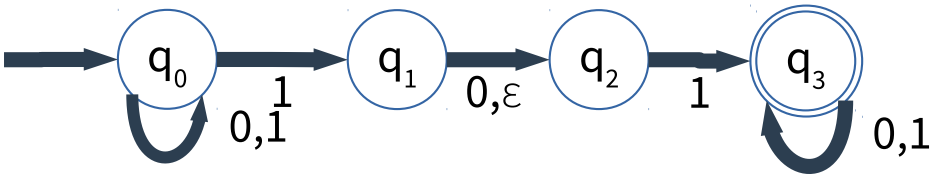
$\{q_0\}$

Serán estados finales aquellos que tengan a q_3

$\{q_0,q_3\}, \{q_0,q_2,q_3\}, \{q_0,q_1,q_3\}, \{q_0,q_1,q_2,q_3\}, \{q_1,q_3\}, \{q_1,q_2,q_3\}, \{q_2,q_3\}, \{q_3\},$

Ejemplo (cont.)

La función de transición



	0	1
$\{q_0\}$	$\{q_0\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$\{q_0,q_3\}$	$\{q_0,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_3\}$
$\{q_0,q_2,q_3\}$	$\{q_0,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$\{q_0,q_1,q_3\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$

	0	1
$\{q_1\}$	$\{q_2\}$	$\varnothing$
$\{q_1,q_2\}$	$\{q_2\}$	$\{q_3\}$
$\{q_1,q_3\}$	$\{q_2,q_3\}$	$\{q_3\}$
$\{q_1,q_2,q_3\}$	$\{q_2,q_3\}$	$\{q_3\}$
$\{q_2\}$	$\varnothing$	$\{q_3\}$
$\{q_2,q_3\}$	$\{q_3\}$	$\{q_3\}$
$\{q_3\}$	$\{q_3\}$	$\{q_3\}$
$\varnothing$	$\varnothing$	$\varnothing$

Presentación realizada en Julio de 2020