

# Algorithm Design and Analysis

วิชาบังคับก่อน: 204251 หรือ 204252; และ 206183 หรือ 206281

ผู้สอน:     ตอน 1 ผศ. เบญจมาศ ปัญญางาม

              ตอน 2 ผศ. ดร. จักริน ชวชาติ

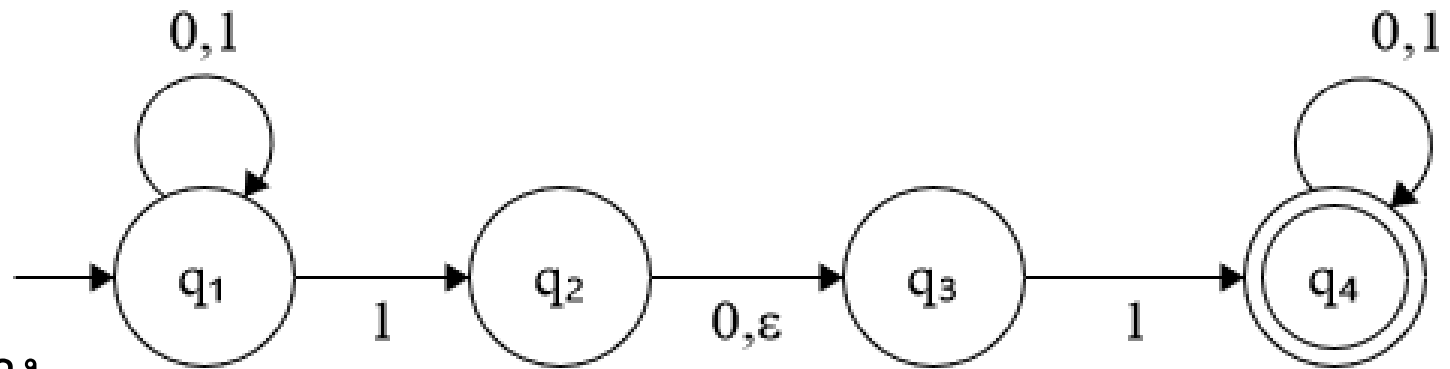
วันสอบปลายภาค : วันพฤหัสบดี ที่ 26 ต.ค. 66

เวลา 12:00 - 15:00 น. (ตามประกาศมหาวิทยาลัย)

บทที่ 12

ออโตมาตา (Automata)

Part III

Nondeterministic Finite Automaton  $N_1$ 

ข้อแตกต่าง

- ❑ ลูกศรออกจาก state มี symbol ซ้ำ (เช่นมี  $1$  ออกสองอันใน  $q_1$ )
- ❑ symbol หาย (เช่น  $q_2$  ถ้ารับ  $1$  มาไปไหนไม่รู้)
- ❑ มี empty string  $\epsilon$

ถ้าถึงจุดต้องเลือก ไม่รู้จะเอาทางไหน เช่นไป  $q_1$  หรือ  $q_2$  ดี ทำอย่างไรดี  
(ลองดู simulation)

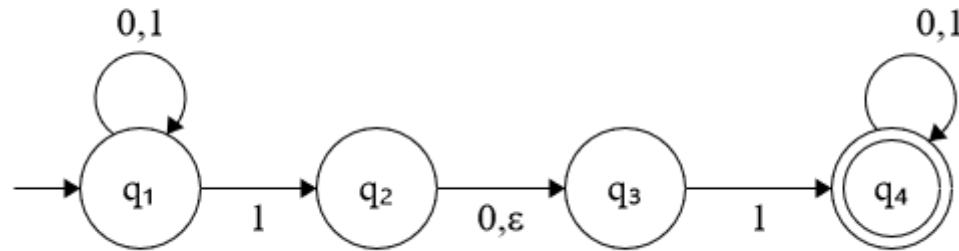
## Deterministic and Nondeterministic Finite Automata

ก่อนหน้านี้เราพิจารณา finite automata ที่ state ถัดไปนั้นถูกตัดสินจาก input alphabet และ current state

การคำนวณซึ่งแต่ละขั้นถูกกำหนดอย่างแน่นอนจะเรียกว่า **deterministic** computation

นั่นคือ ใน **nondeterministic** computation นั้น แต่ละขั้นอาจจะมีหลายทางเลือก

ดังนั้นเราจะเรียกให้ต่างกันไปเลยคือ deterministic finite automata(**DFA**) และ nondeterministic finite automata(**NFA**)



แต่ละจุดที่มีทางเลือกไปยังขั้นต่อไปหลายทาง machine จะแบ่งตัวมันเอง  
ออกเป็นหลายๆ copy แล้วแยกกันไปทุกทางที่เป็นไปได้พร้อมๆกัน

- ถ้ามีหลายทางเลือก ให้แบ่งตัวไปทำงาน
- ตัวที่แบ่งจะทำงานต่อไปเรื่อยๆ และจะตายถ้าไม่สามารถเดินต่อไปตาม input ที่ได้รับได้
- จะ accept string เมื่อไร  
เมื่อ input หมดแล้วถ้ามีตัวที่แบ่งสักตัวหนึ่ง(any) ที่อยู่ใน accept state เราจะ **accept** input นั้น

The diagram shows a directed graph with nodes labeled  $q_1$  through  $q_{14}$ . The root node is  $q_1$ . A red path highlights a sequence of nodes:  $q_1$  (root) ->  $q_2$  ->  $q_3$  ->  $q_4$  (goal). The graph illustrates the search space for a 3-disk Tower of Hanoi problem, showing various states and transitions.

# Definition

A **nondeterministic finite automaton** is a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where

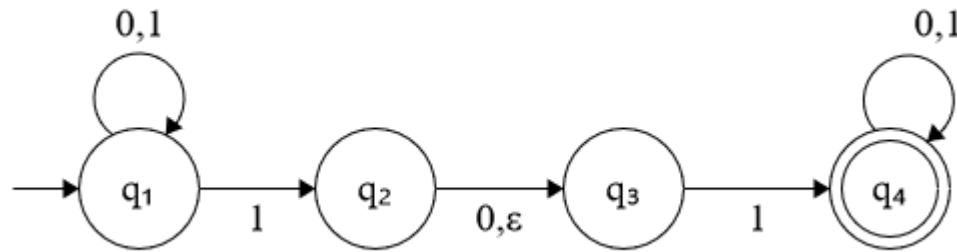
$Q$  is a finite set of states

$\Sigma$  is a finite alphabet

$\delta: Q \times \Sigma_{\varepsilon} \rightarrow P(Q)$  is the transition function

$q_0 \in Q$  is the start state

$F \subseteq Q$  is the set of accept states



$N_1$  is  $(Q, \Sigma, \delta, q_0, F)$  where

$$Q = \{q_1, q_2, q_3, q_4\}$$

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

$\delta$ :

$q_1$  is the start state

$$F = \{q_4\}$$

	0	1	$\varepsilon$
$q_1$	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
$q_2$	$\{q_3\}$	$\emptyset$	$\{q_3\}$
$q_3$	$\emptyset$	$\{q_4\}$	$\emptyset$
$q_4$	$\{q_4\}$	$\{q_4\}$	$\emptyset$



# NFA มีพลังมากกว่า DFA?

ด้วยพลังของ nondeterminism (ความไม่แน่นอน) ทำให้ดูเหมือนว่า NFAs มีพลังมากกว่า เหมือนว่ามันทำได้

แต่ความเป็นจริงแล้ว DFAs และ NFAs นั้น recognize ภาษา class เดียวกัน

เราจะบอกว่า 2 machine นั้น equivalent กันถ้าพวกมัน recognize ภาษาเดียวกัน

# ในการพิสูจน์ว่า equivalence

## พิสูจน์ 2 ทาง

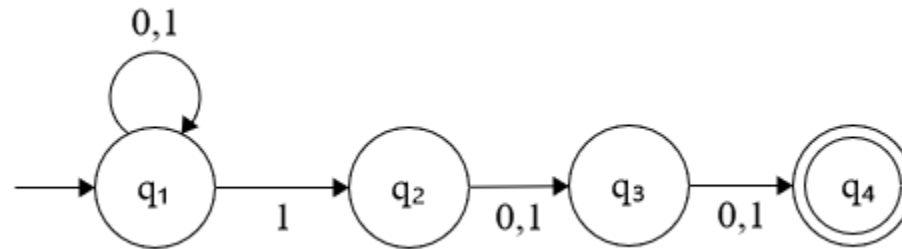
ทางแรก กำหนด DFA มาให้แล้วสร้าง NFA ที่ recognize ภาษาเดียวกัน ทางนี้ง่าย DFA เป็น NFA อยู่แล้ว (ทำไม)

ทางที่สอง กำหนด NFA มาให้แล้วสร้าง DFA ที่ recognize ภาษาเดียวกัน ทางนี้ไม่ง่าย

## Theorem

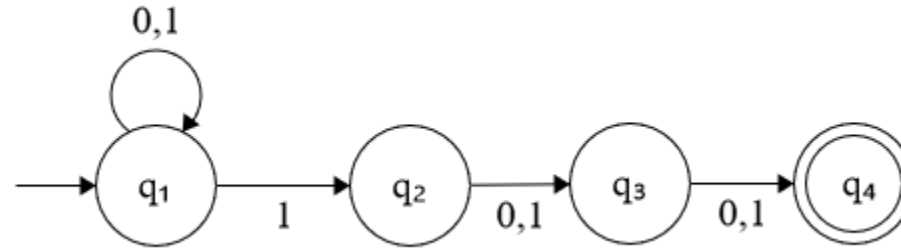
Every nondeterministic finite automaton has an equivalent deterministic finite automaton

- NFA นี้ รับอะไร

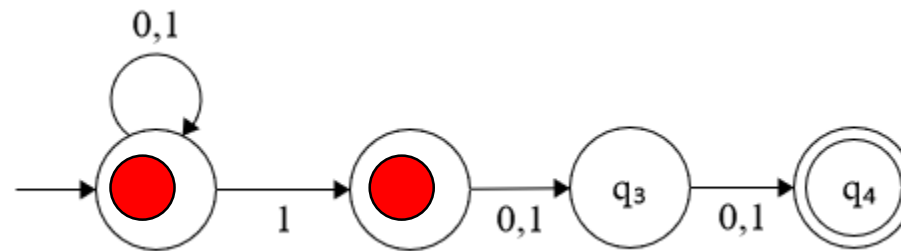


- รับ 1 ในตำแหน่งที่ 3 จากท้าย
- เราจะจำลองมัน

10011



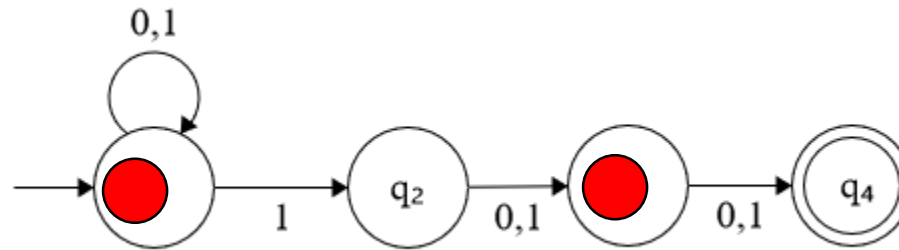
รับ 1 มาเกิดอะไรขึ้น



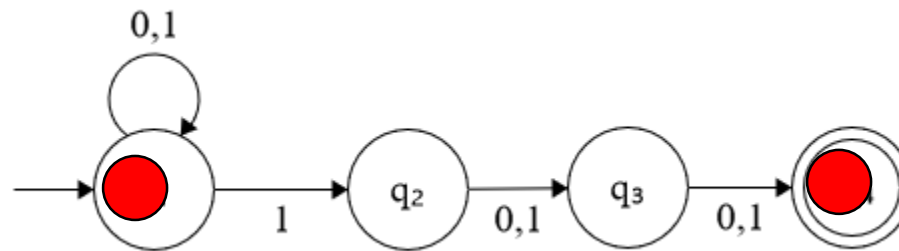
บทที่  
12

10011

รับ 0



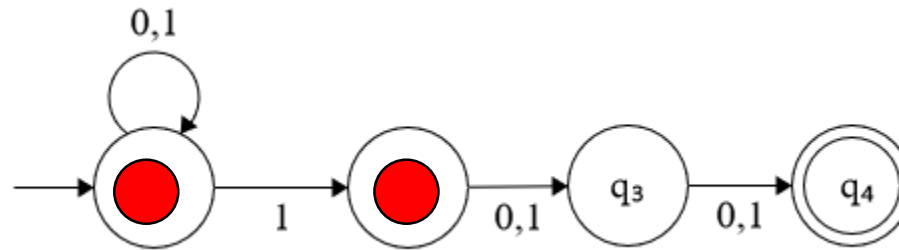
รับ 0



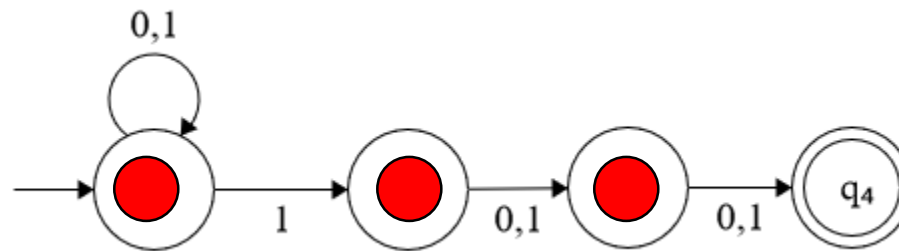
บทที่  
12

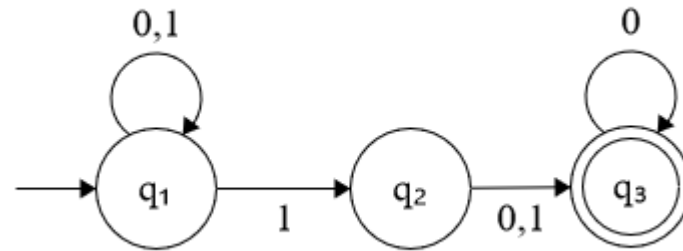
10011

รับ 1

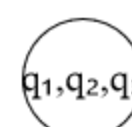
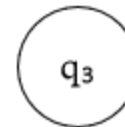
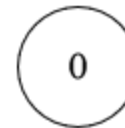
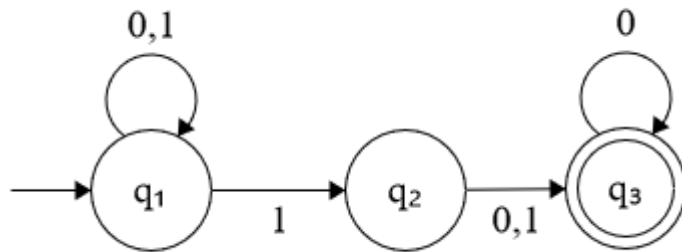


รับ 1

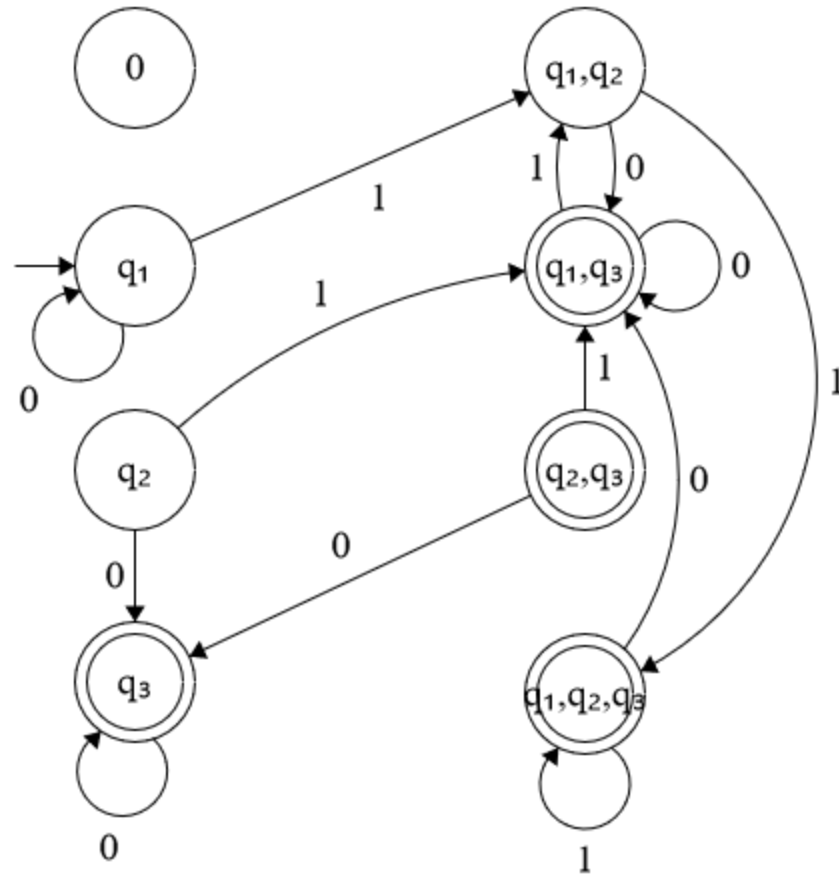




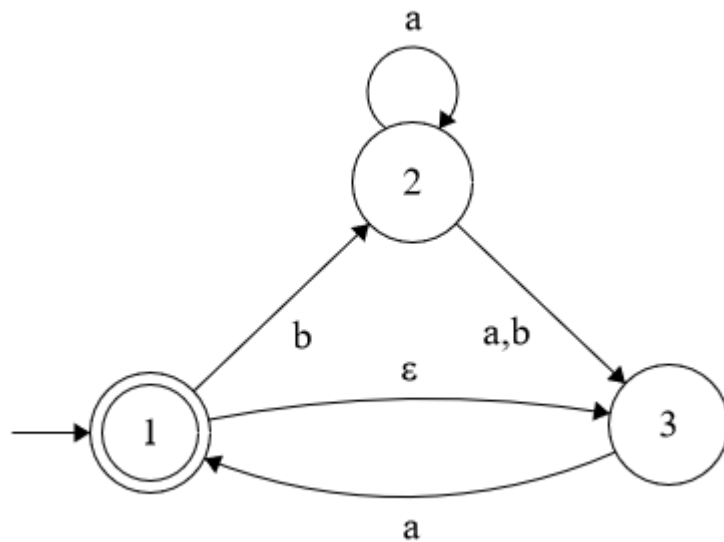
- ❑ เราจะต้องจำว่า input แต่ละอันสถานะก่อนอยู่ที่ไหนบ้าง
- ❑ นอกจากนี้มันอาจจะเป็นไปได้ว่าบาง state อาจจะมี copy มารวมกันก็รวมกันเป็น copy เดียว
- ❑ ดังนั้น state ทั้งหมดที่เป็นไปได้จะมี( $2^n$ )  
 $\{\emptyset, \{q_1\}, \{q_2\}, \{q_3\}, \{q_1, q_2\}, \{q_1, q_3\}, \{q_2, q_3\}, \{q_1, q_2, q_3\}\}$







- หากมี  $\mathcal{E}$  จะจัดการอย่างไร
- เนื่องจากเมื่อมี  $\mathcal{E}$  อยู่บนลูกศร  $N$  จะสามารถย้ายได้อย่างอิสระไปตามลูกศรโดยที่ไม่ต้องรับ input
- ตกลงกันก่อนว่า เดินแล้วค่อย copy ตัวเอง



0

{1,2}

{1}

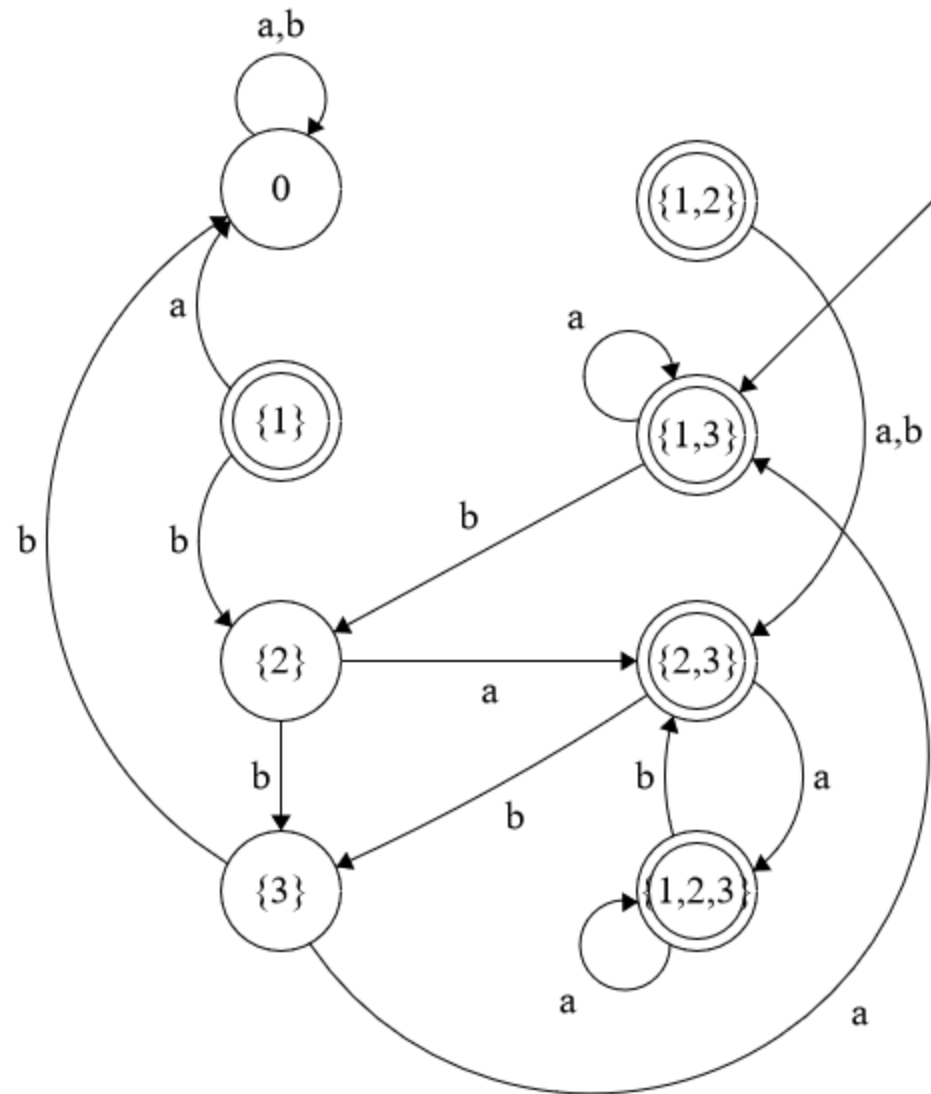
{1,3}

{2}

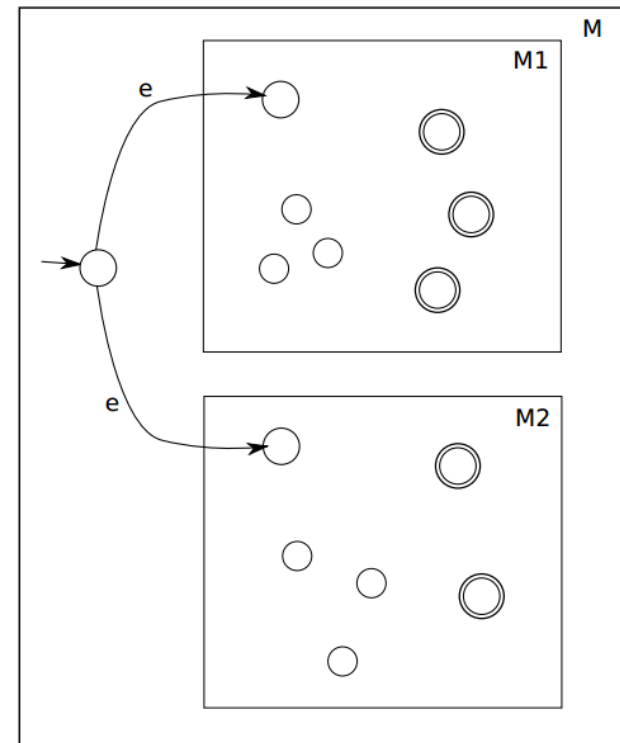
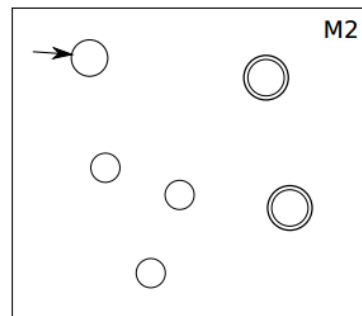
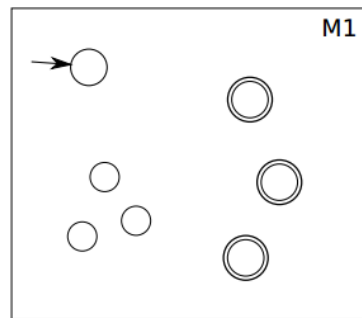
{2,3}

{3}

{1,2,3}

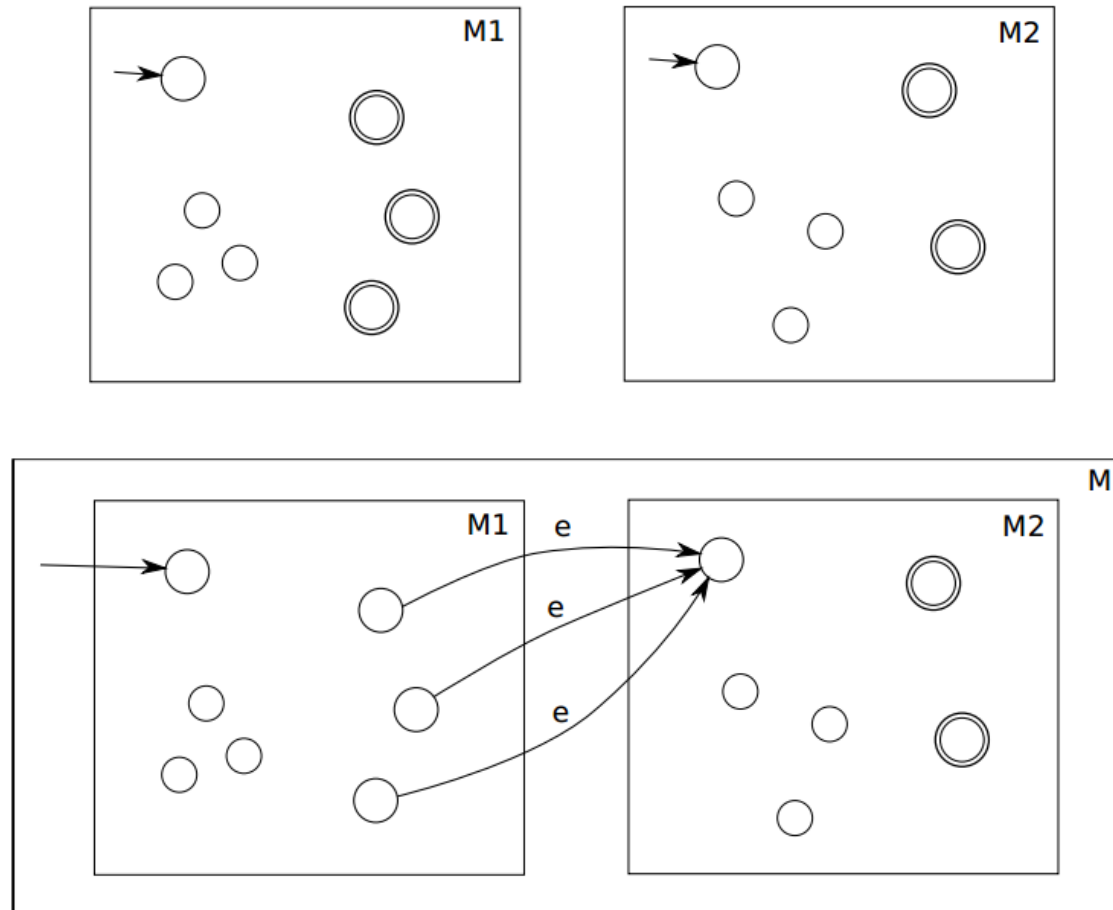


- เนื่องจาก NFA-DFA equivalence ทำให้เรา ไม่ต้องสร้าง deterministic finite automata สำหรับแสดงว่า regular operations (concatenation union star) มีคุณสมบัติปิด เราก็สร้าง nondeterministic finite automata แทน
- ถ้าเจอ Union ทำอย่างไร



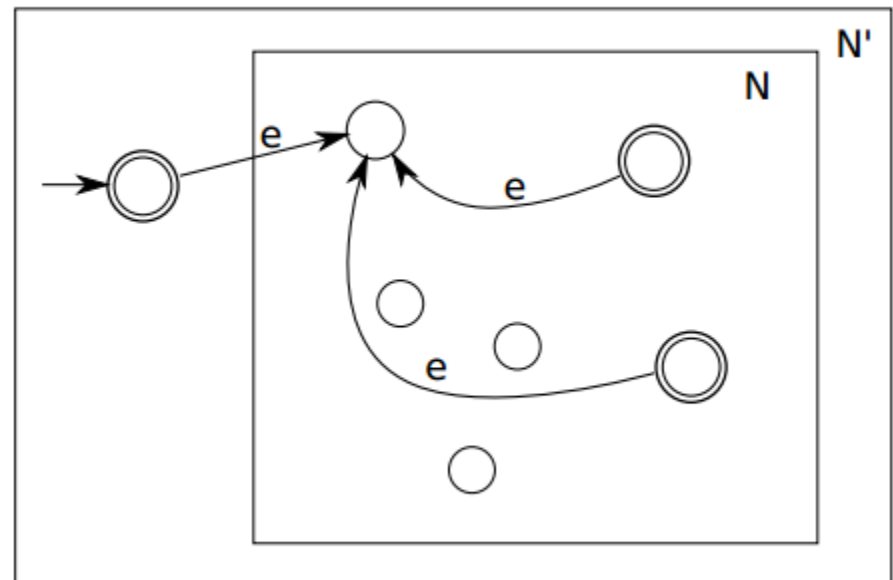
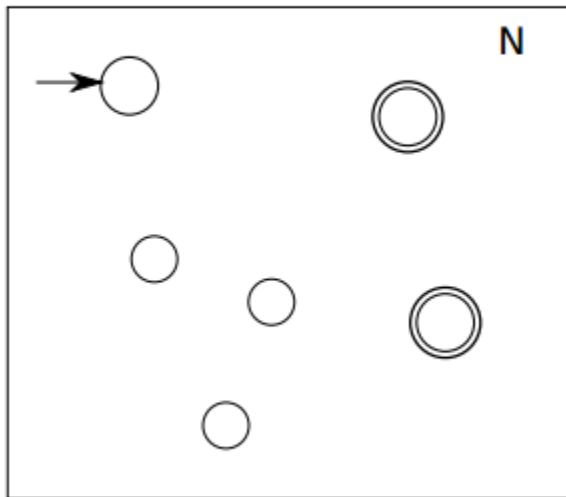
## concatenation

- ถ้าเจอ concatenation ทำอย่างไร



## star

- ถ้าเจอ star ทำอย่างไร



## Assignment#10 ส่วนที่ 1

จากนิยาม Finite Automaton, FM1 ที่กำหนดให้

จงวาด state diagram of machine FM1

☐  $FM1 = \{Q, \Sigma, \delta, q_0, F\}$

☐  $Q = \{A, B, C, D\}$

☐  $\Sigma = \{0, 1\}$

☐  $\delta$  is described as:

	0	1	$\epsilon$
A	{C}	$\emptyset$	{B}
B	$\emptyset$	{C}	$\emptyset$
C	{D}	{D}	$\emptyset$
D	{D}	{D}	$\emptyset$

☐  $q_0 = A$

☐  $F = \{D\}$



## Assignment#10 ส่วนที่ 2

กำหนดให้  $\Sigma = \{0, 1\}$

- ☐ จงหา FA  $M_1$  ที่ recognize  $01^+$
- ☐ จงหา FA  $M_2$  ที่ recognize  $(10)^*$
- ☐ จงหา FA  $M_3$  ที่ recognize  $(01^+) \cup (10)^*$