

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 I

Automata

- คำว่า Automata มาคำในจากภาษากรีก **αὐτόματα** ซึ่งหมายความว่า ทำงานได้ด้วยตัวเอง

Automaton (เอกพจน์, Automata พหูพจน์) หมายถึงรูปแบบนามธรรมของอุปกรณ์คำนวณที่ทำงานแบบอัตโนมัติตามลำดับของการดำเนินการที่กำหนดไว้ก่อน

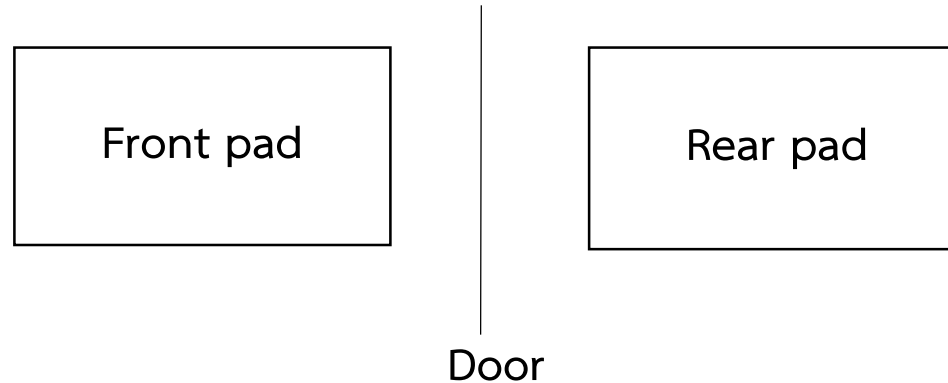
- Automaton ที่มีจำนวนสถานะ (State) จำกัดจะเรียกว่า **Finite Automaton (FA)** หรือ **Finite State Machine (FSM)**

Finite automata

- ❑ Finite automata เป็นโมเดลที่ดีสำหรับคอมพิวเตอร์ที่มีหน่วยความจำที่จำกัด
- ❑ คำถามคือ computer ที่มีหน่วยความจำน้อยมากๆ ทำอะไรได้บ้าง?
- ❑ ในชีวิตประจำวัน เราได้ใช้คอมพิวเตอร์แบบนี้อยู่ตลอด
 - ▶ เช่น เครื่องใช้ไฟฟ้า ตัวควบคุมประตูอัตโนมัติ(เป็นบานพับ)

ประต้อัตโนมัติ : หลักการทำงาน

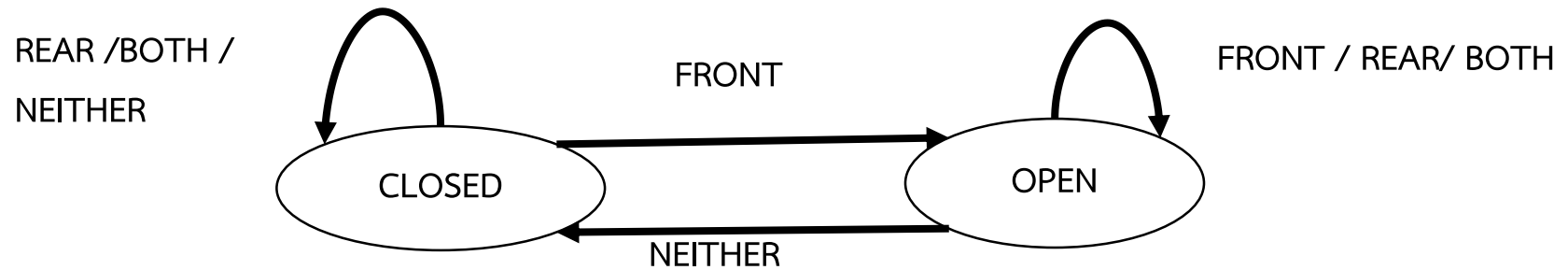
“ประตูจะเปิดเมื่อมันตรวจพบว่ามีคนยืนอยู่”



- ประต้อัตโนมัติจะมีอุปกรณ์ด้านหน้าเพื่อตรวจว่ามีคนเดินผ่านมาทางเข้า และจะมีอุปกรณ์อีกอันอยู่ด้านหลังทางเข้าเพื่อที่ตัวควบคุมจะเปิดประตูค้างไว้นานพอที่คนจะผ่านไปได้ และไม่ชนคนหากมีคนยืนด้านหลังถ้ามันเปิด

State Diagram สำหรับตัวควบคุมประตูอัตโนมัติ

ตัวควบคุมจะอยู่ใน สถานะ (state) “OPEN” หรือ “CLOSED”



พบว่ามีข้อมูลเข้า(input) ที่เป็นไปได้ 4 อย่าง

FRONT : หมายถึง คนยืนหนึ่งที่ Front pad

REAR : หมายถึง คนยืนหนึ่งที่ Rear pad

BOTH : หมายถึง คนยืนที่ pad ทั้งสอง

NEITHER : หมายถึง ไม่มีใครยืนที่ pad ใดเลย

State transition table ของตัวควบคุมประตูอัตโนมัติ

Input signal

state

	NEITHER	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	CLOSED	CLOSED
OPEN	CLOSED	OPEN	OPEN	OPEN

state ของตัวควบคุม จะเปลี่ยน (หรือเหมือนเดิม) หลังจากได้รับ input

- เมื่อประตูอยู่ในสถานะ OPEN
 - หากได้รับ input เป็น FRONT REAR หรือ BOTH ประตูยังคงเปิดเหมือนเดิม
 - แต่เมื่อได้รับ input เป็น NEITHER จะเปลี่ยนเป็นสถานะเป็น CLOSED

State transition table ของตัวควบคุมประตูอัตโนมัติ

Input signal

state

	NEITHER	FRONT	REAR	BOTH
CLOSED	CLOSED	OPEN	CLOSED	CLOSED
OPEN	CLOSED	OPEN	OPEN	OPEN

state ของตัวควบคุม จะเปลี่ยน (หรือเหมือนเดิม) หลังจากได้รับ input

- เมื่อประตูอยู่ในสถานะ CLOSED
 - หลังได้รับ input เป็น NEITHER หรือ REAR มันจะยังคงอยู่ในสถานะ CLOSED
 - หาก input เป็น BOTH มันจะยังคง CLOSED เพราะว่าการเปิดประตูเสี่ยงที่จะไปชนคนที่ยืนอยู่ที่ Rear pad
 - แต่หาก input เป็น FRONT มันจะย้ายสถานะเป็น OPEN

การเปลี่ยน state ของตัวควบคุมประตูอัตโนมัติ

จาก Input ที่กำหนดให้ จะมีการเปลี่ยนแปลงค่า State ของตัวควบคุมอย่างไร

Input	สถานะ
เริ่มต้น	CLOSED
FRONT	OPEN
REAR	OPEN
NEITHER	CLOSED
FRONT	OPEN
BOTH	OPEN
NEITHER	CLOSED
REAR	CLOSED
NEITHER	CLOSED

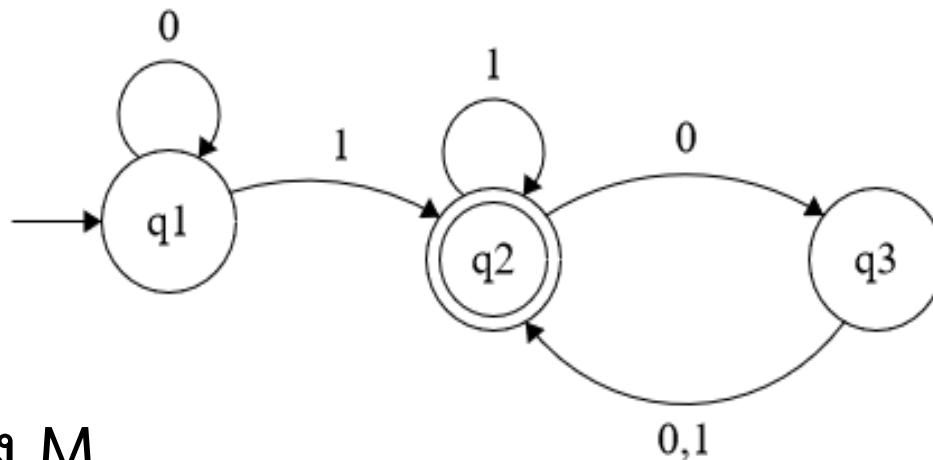
การเปลี่ยน state ของตัวควบคุมประตูอัตโนมัติ

- ประตูอัตโนมัตินี้เป็น Finite automata โดยมีตัวควบคุมเป็น computer ที่มีหน่วยความจำ 1 bit สามารถจดจำสถานะของตัวควบคุมได้ 2 สถานะ
- ตัวควบคุมอาจมีหน่วยความจำหลาย bit ในการเก็บข้อมูลก็ได้ ตัวอย่าง computer ที่มีหน่วยความจำจำกัดอุปกรณ์อื่น เช่น ลิฟต์ เครื่องล้างจาน เครื่องซักผ้า เครื่องขายอัตโนมัติ เป็นต้น
- ตัวควบคุมของลิฟต์ อาจต้องเก็บสถานะที่หมายถึงชั้นที่ลิฟต์อยู่ และ input อาจจะเป็นสัญญาณจากปุ่มกด

Finite automata

ในมุมมองทางคณิตศาสตร์ อาจให้นิยามที่ชัดเจนของ finite automata และคำศัพท์ต่างๆ ที่เกี่ยวข้องสำหรับอธิบายการดำเนินการของ finite automata

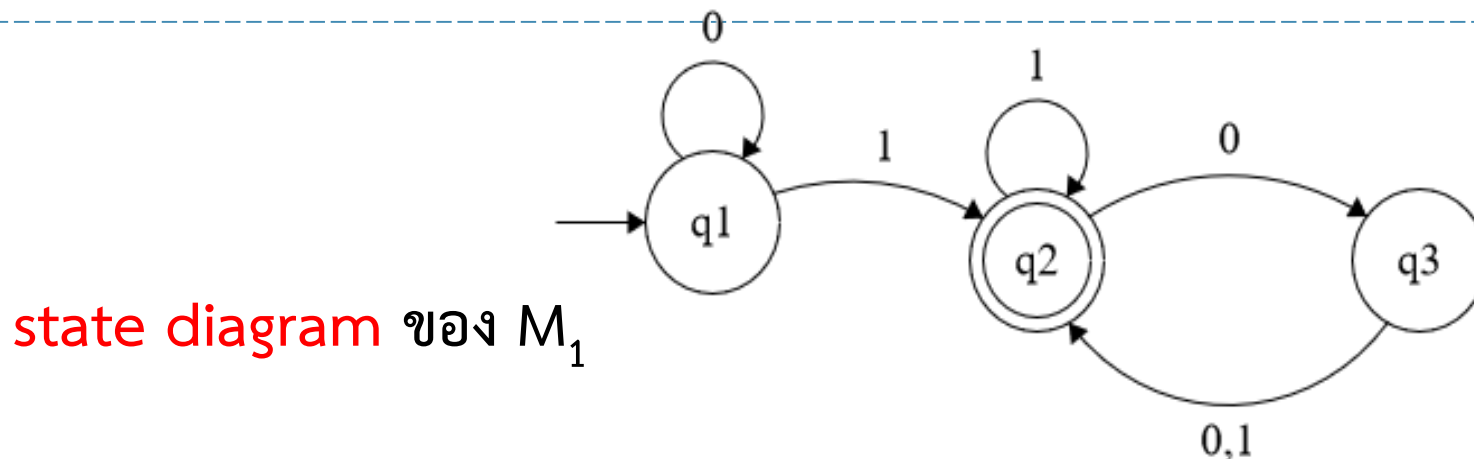
ตัวอย่าง Finite automaton ชื่อ M_1
ประกอบด้วย 3 state



state diagram ของ M_1

State diagram

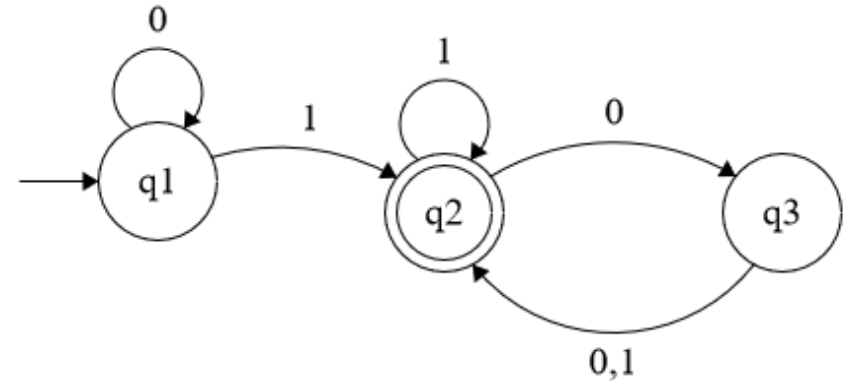
- **start state** เป็น state ที่มีลูกศรชี้เข้าที่ไม่มีต้นทาง
- **accept state** เป็น state ที่มีวงสองวงซ้อนกัน
- ลูกศรที่ชี้จาก state หนึ่งไปอีก state หนึ่ง เรียกว่า **transitions**



จากรูปตัวอย่าง M_1 ที่มี 3 state ชื่อ q_1 , q_2 , q_3 โดยมี q_1 เป็น **start state** และ q_2 เป็น **accept state**

กระบวนการทำงานของ Automata

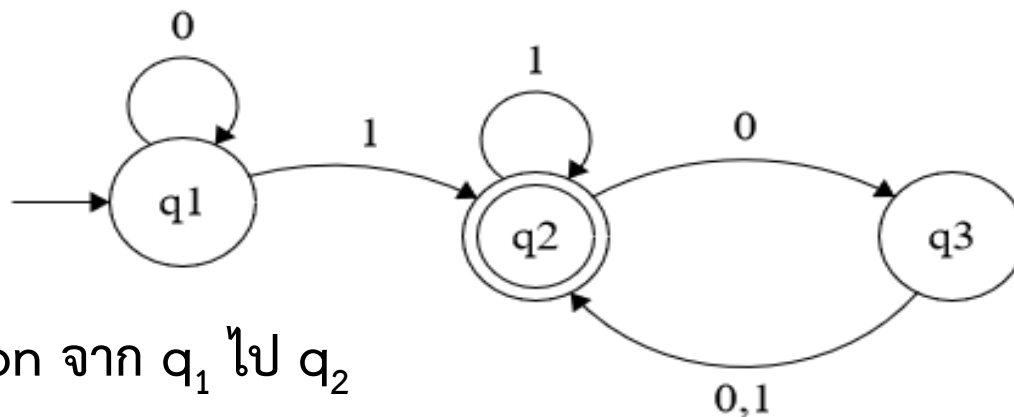
FA จะดำเนินการตาม input string ที่ได้รับ
และจะให้ output เป็นได้ **accept** หรือ **reject**



- ❑ เริ่มจาก start state
- ❑ รับ symbols จาก input string ทีละตัวจากซ้ายไปขวา
- ❑ แต่ละ symbol ที่รับเข้ามา FA จะย้ายจาก state หนึ่งไปอีก state หนึ่งตาม transition ที่ตรงกับ symbol
- ❑ หลังรับ symbol ตัวสุดท้าย FA จะให้ output เป็น accept output หากหยุดที่ accept state นอกนั้น reject

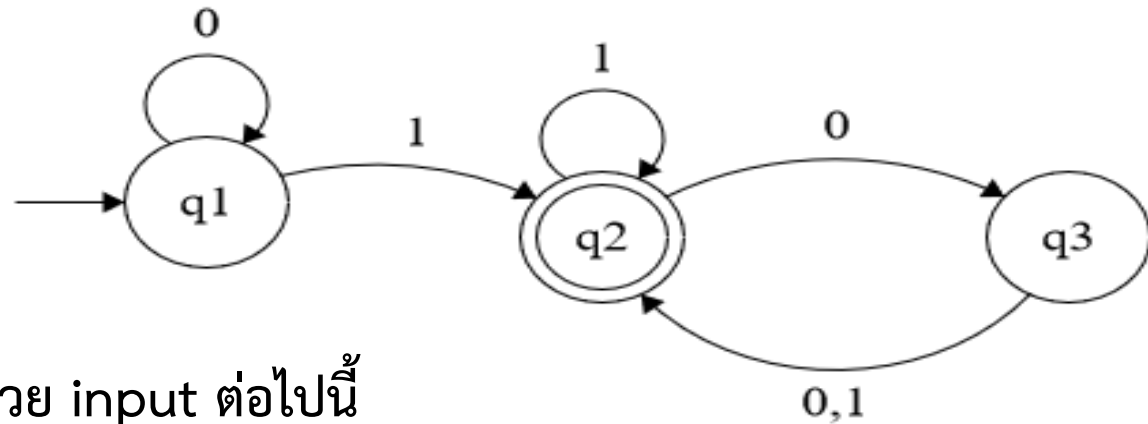
กระบวนการทำงานของ Automata

ตัวอย่างการทำงานของ M_1 เมื่อรับ input string เป็น 1101



1. เริ่มที่ q_1
2. อ่าน 1, เปลี่ยน transition จาก q_1 ไป q_2
3. อ่าน 1, เปลี่ยน transition จาก q_2 ไป q_2
4. อ่าน 0, เปลี่ยน transition จาก q_2 ไป q_3
5. อ่าน 1, เปลี่ยน transition จาก q_3 ไป q_2
6. Accept เพราะว่า M_1 อยู่ใน accept state q_2 เมื่อ input หมด

กระบวนการทำงานของ Automata



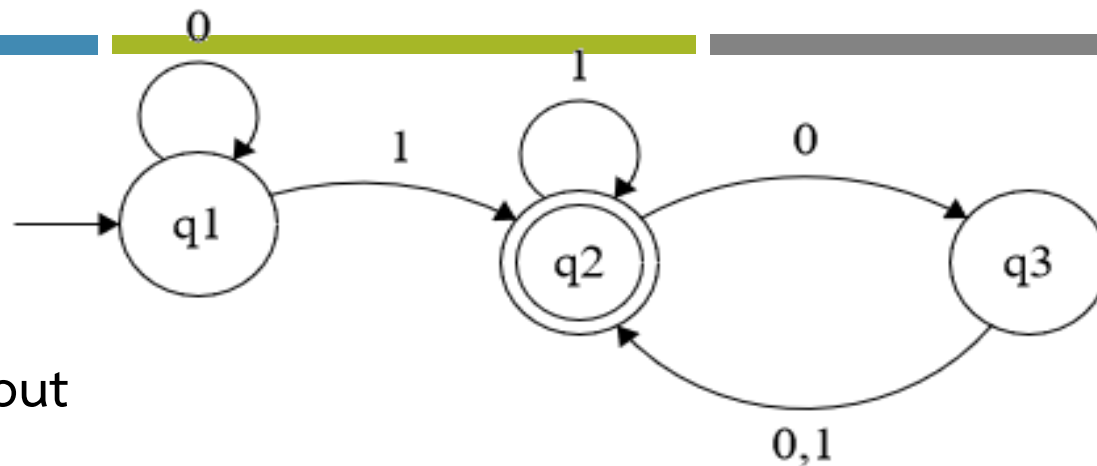
ทดลองกับ machine นี้ ด้วย input ต่อไปนี้

1, 01, 11, 010101010101

00, 10, 110, 11000

100, 1100, 110000

กระบวนการทำงานของ Automata



พบว่า M_1 จะให้ output

- 1) เป็น accept เมื่อ Input string
 - ลงท้ายด้วย 1 หรือ
 - ลงท้ายด้วย 0 เป็นจำนวนคู่แบบมี 1 นำหน้าอย่างน้อย 1 ตัว
- 2) เป็น reject เมื่อรับ string อื่นๆ เช่น 0, 10, 101000

จะอธิบายนิยามภาษาที่ M_1 นี้ accept ว่าอย่างไรดี

Formal Definition of a Finite Automata

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

1. Q is a finite set called the states,
2. Σ is a finite set called the alphabet,
3. $\delta: Q \times \Sigma$ is the **transition function**,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

ทำไมต้องมี Formal definition

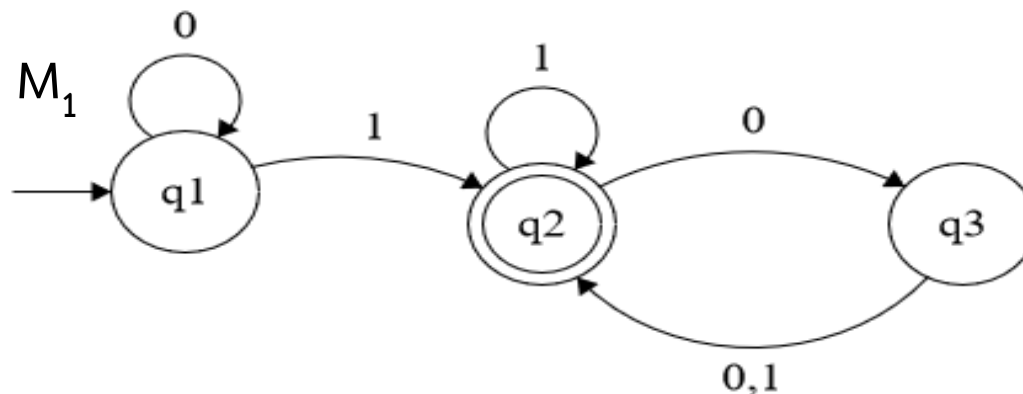
- เพื่อความถูกต้อง (precision)
- เพื่อเป็น Notation

Transition Function

- ❑ Transition Function เรียกว่ากฎในการย้ายสถานะ
- ❑ กำหนดให้ δ เป็นฟังก์ชันจากเซตของ state และ เซตของ input ที่เป็นไปได้ ไปยังเซตของ state
- ❑ เช่น $\delta(q, 1) = p$
 - ▶ หากอยู่ในสถานะ q เมื่อ input ที่รับเข้ามาเป็นค่า 1 จะย้ายไปสถานะ p

Formal Definition of a Finite Automata

ตัวอย่างการอธิบายการทำงานของ M_1 โดยที่



$M_1 = (Q, \Sigma, \delta, q_1, F)$ เมื่อ

เซตของ state $Q = \{q_1, q_2, q_3\}$

เซตของ input $\Sigma = \{0, 1\}$

δ คือ transition function

q_1 เป็น start state

เซตของ Final state $F = \{q_2\}$



	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

The language of a finite automaton

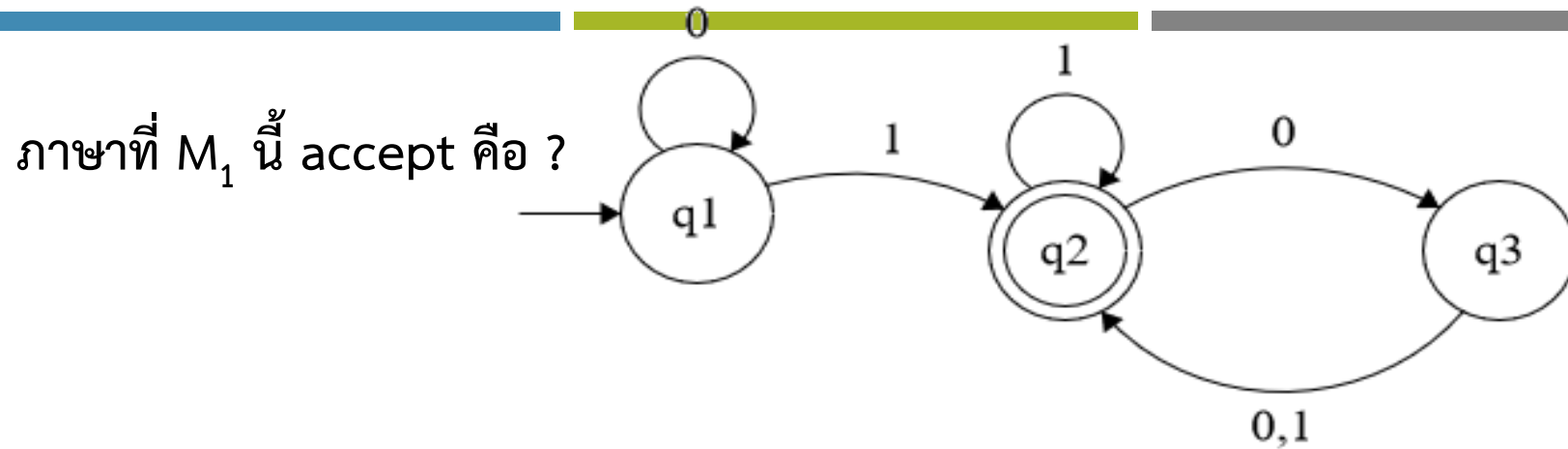
“is the set of strings that it accepts”

ถ้า A เป็นเซตของ string ทั้งหมดที่ machine M accept เราจะเรียกว่า A เป็น language of machine M เขียนได้เป็น

$$L(M) = A$$

- ▶ หรือกล่าวได้ว่า M recognizes A หรือ M accepts A
- Machine สามารถ accept ได้หลาย string
- ▶ แต่จะ recognizes ได้เพียงหนึ่ง language
- ถ้า machine accept no strings มันจะ recognize อยู่หนึ่ง language ที่ชื่อว่า empty language $\rightarrow \emptyset$

Language of machine

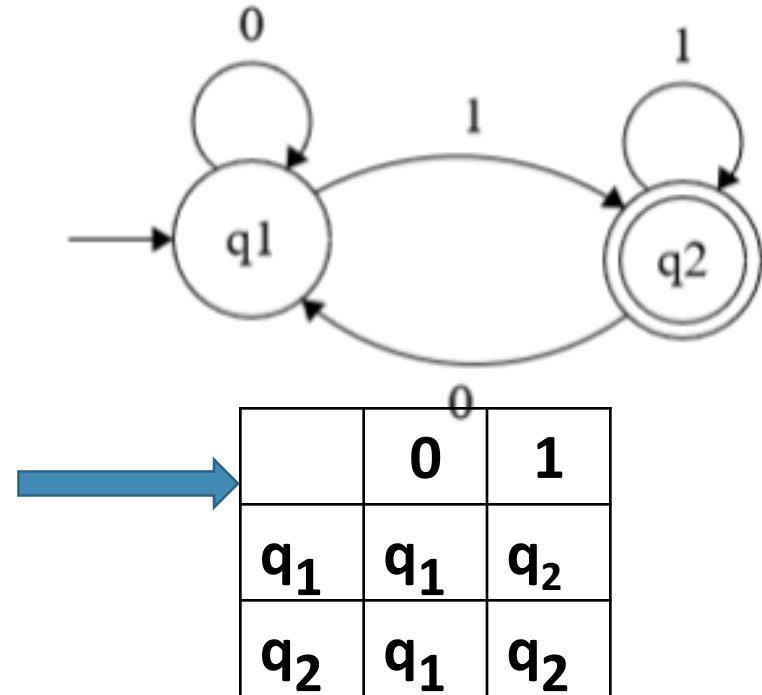


- ❑ $A = \{w \mid w \text{ contains at least one 1 and an even number of 0s follow the last 1}\}$
- ❑ จะได้ว่า $L(M_1) = A$ หรือ M_1 recognizes A

ตัวอย่าง state diagram ของ machine M_2

□ จงเขียน formal description ของ machine M_2

$M_2 = (Q, \Sigma, \delta, q_1, F)$ เมื่อ
 เซตของ state $Q = \{q_1, q_2\}$
 เซตของ input $\Sigma = \{0, 1\}$
 δ คือ transition function
 q_1 เป็น start state
 เซตของ Final state $F = \{q_2\}$



• ภาษาที่ M_2 นี้ accept คือ ?

$$L(M_2) = \{ w \mid w \text{ ends with } 1 \}$$

“ M_2 accept string ที่ลงท้ายด้วย 1”

ตัวอย่าง state diagram ของ machine M_3

“ M_3 มี 2 accept state คือ
 q_1 และ r_1 ”

$$M_3 = (Q, \Sigma, \delta, s, F)$$

เมื่อ

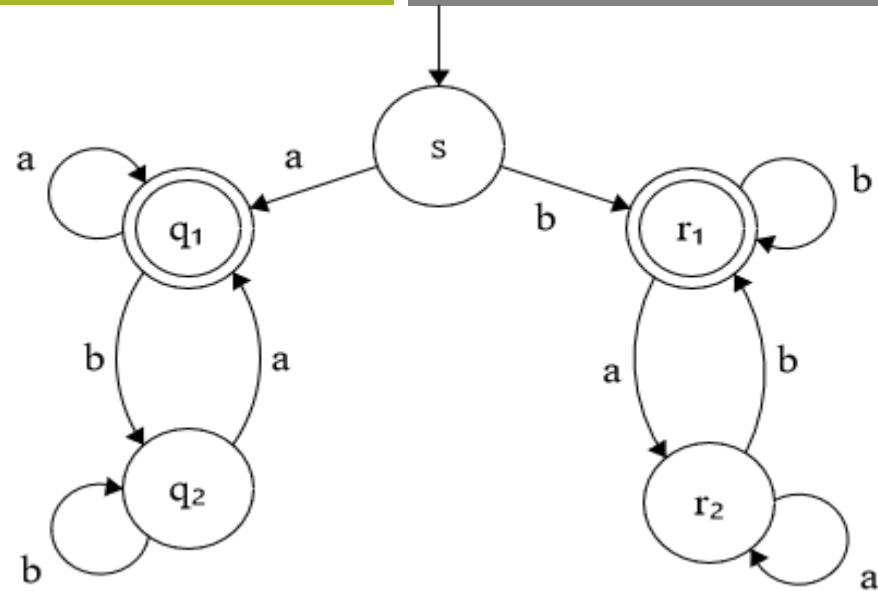
เซตของ state $Q = \{s, q_1, q_2, r_1, r_2\}$

เซตของ input $\Sigma = \{a, b\}$

δ คือ transition function

s เป็น start state

เซตของ Final state $F = \{q_1, r_1\}$



	q1	q2	r1	r2
s	a	-	b	-
q1	a	b	-	-
q2	a	b	-	-
r1	-	-	b	a
r2	-	-	b	a

ตัวอย่าง state diagram ของ machine M_3

- ภาษาที่ M_3 นี้ accept คือ ?

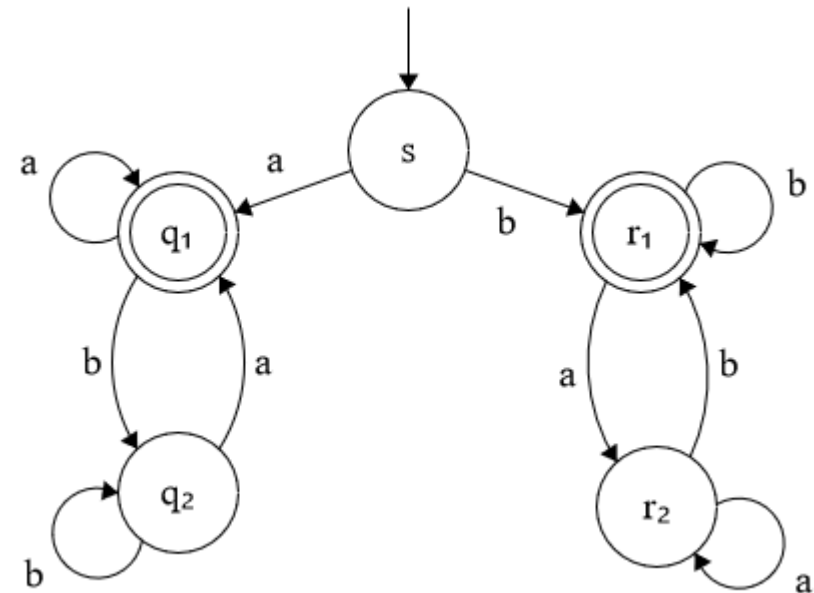
ลองแทนค่าข้อมูลนำเข้า พบว่า

M_3 accept string

a, b, aa, bb, aba

แต่ reject string

ab, ba, baa



$$L(M_3) = \{ w \mid w \text{ starts and ends with same symbol} \}$$

“ accepts ทุก string ที่เริ่มต้นและลงท้ายด้วยอักขระตัวเดียวกัน ”

Assignment#9

Machine ต่อไปนี้จะ recognize ภาษา ?

