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 II

Designing Finite Automata

- เมื่อมีการรับ input ทีละตัว แล้วต้องจำสิ่งใดบ้างเพื่อที่จะได้ตัดสินใจได้อย่างถูกต้อง   สิ่งที่ต้องจำจะเป็น set ของ state
- ตัวอย่างเช่น ต้องการออกแบบ machine สำหรับ recognize ภาษาที่ประกอบไปด้วยทุก string ที่มี 1 เป็นจำนวนคี่
- Input Σ = เซตของอักขระ {0,1}
- สิ่งที่ต้องจำ คือ ต้องรู้ว่าตอนนี้นับ 1 ได้เป็นจำนวนเป็นคี่หรือยัง และจะเก็บการจำนี้อย่างไร

Designing Finite Automata

- ออกแบบ finite automata E1 ที่ recognize ภาษาประกอบด้วยทุก string ที่มี 1 จำนวนคี่ตัว

1) สิ่งที่ต้องจำ คือ ต้องรู้ว่าใน state ปัจจุบันเป็นสถานะของการมี 1 เป็นจำนวนคู่หรือคี่

2) กำหนด transition จากการมองว่าวิธีการเปลี่ยนจากสถานะหนึ่งไปอีกสถานะหนึ่งเมื่อได้รับ symbol

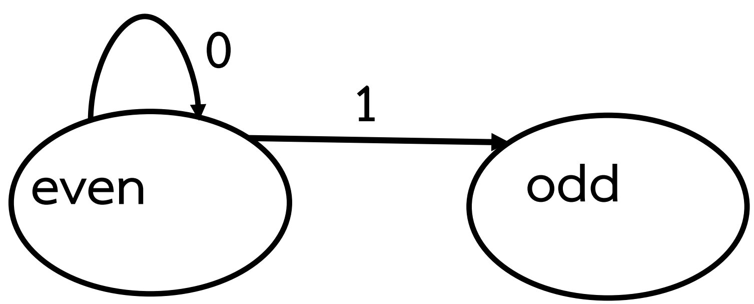
- นั่นคือ หากได้รับ 0 หรือ 1 ควรเปลี่ยนเป็นสถานะของการมี 1 เป็นจำนวนคู่หรือคี่จึงจะถูกต้อง

Designing Finite Automata

- ออกแบบ finite automata E1 ที่ recognize ภาษาประกอบด้วยทุก string ที่มี 1 จำนวนคี่ตัว
- กำหนด transition

2.1) หากสถานะปัจจุบันเป็น state ที่มี 1 เป็นจำนวนคู่(even) มาก่อน ดังนั้น

- หากได้รับ 1 จะเปลี่ยนไปเป็น state ที่มี 1 เป็นจำนวนคี่
- หากได้รับ 0 ยังคงอยู่ที่ state ที่มี 1 เป็นจำนวนคู่อยู่



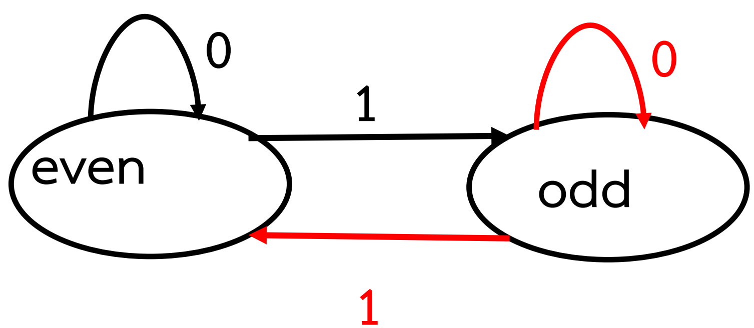
Designing Finite Automata

- ออกแบบ finite automata E1 ที่ recognize ภาษาประกอบด้วยทุก string ที่มี 1 จำนวนคี่ตัว
- กำหนด transition

2.2) หากสถานะปัจจุบันเป็น state ที่มี 1 เป็นจำนวนคี่ (Odd) มาก่อน ดังนั้น

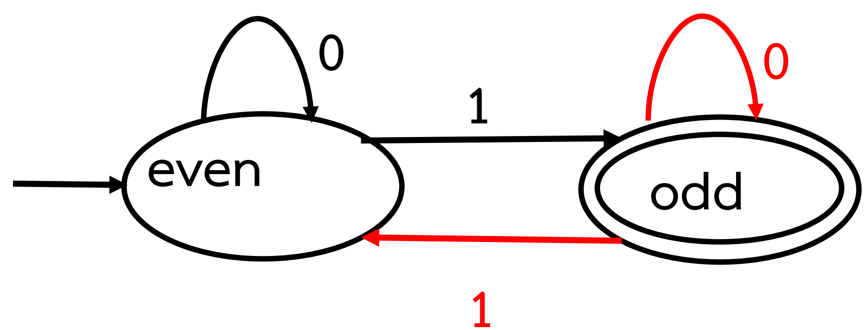
- หากได้รับ 1 จะเปลี่ยนไปเป็น state ที่มี 1 เป็นจำนวนคู่

- หากได้รับ 0 ยังคงอยู่ที่ state ที่มี 1 เป็นจำนวนคี่อยู่



Designing Finite Automata

3. ระบุ start state โดยดูว่าถ้าไม่มี symbol หรือเป็น empty string จะอยู่ที่ state ไหน
   - ตัวอย่างนี้ start state ที่สอดคล้อง คือ even
   - และกรณี input string คือ 0 ก็ถือว่า มี 1 เป็นจำนวนคู่
4. ระบุ accept state ซึ่งคือสถานะ odd



Designing Finite Automata

จงออกแบบ finite automata E2 ที่ recognize regular language ของทุก string ที่มี 001 เป็น substring ตัวอย่างเช่น 0001, 1001, 001, 1011101011100101 ทุกตัวที่กล่าวมาอยู่ในภาษา แต่ 11, 000 ไม่ใช่

Regular operations

- Operation สำหรับดำเนินการกับ Language ซึ่งเป็นเครื่องมือสำหรับสร้าง machine ที่ซับซ้อนขึ้น

Let A and B be languages.

We define the regular operations union, concatenation, and star as follows.

- Union: $A \cup B = \{x \mid x \in A$ or $x \in B\}$
- Concatenation: $A \circ B = \{xy \mid x \in A$ and $y \in B\}$
- Star: $A^* = x_1 x_2 \ldots x_k$ $k \geq 0$ and each $x_i \in A$

Regular operations

- ตัวอย่างเช่น

ให้ Σ เป็นตัวอักษรภาษาอังกฤษตัวเล็ก 26 ตัว {a,b,…,z}

ถ้า A = { good, bad} และ B = {boy, girl}

- $A \cup B$ = {good, bad, boy, girl}
- $A \circ B$= {goodboy, goodgirl, badboy, badgirl }
- $A*$= {ϵ, good, bad, goodgood, goodbad, badbad, badgood, …}

คุณสมบัติปิด

- Set จะมีคุณสมบัติปิดภายใต้การดำเนินการบางอย่าง
- ถ้านำสมาชิกของเซตมาดำเนินการภายใต้การดำเนินการนั้นแล้ว ผลลัพธ์ที่ได้ก็ยังอยู่ในเซต
- ตัวอย่าง เช่น เซตของจำนวนธรรมชาติ มีคุณสมบัติปิดภายใต้การคูณ

Irregular Language

เรารู้ว่าภาษาใดจะเป็น irregular language ถ้าไม่มี finite automaton ที่ recognize มันได้

<u>ต้องการทราบว่า</u>

เซตของ Regular language มีคุณสมบัติปิดภายใต้ union? นั่นคือถ้า A_1 และ A_2 เป็น regular แล้ว $A_1 \cup A_2$ เป็น regular?

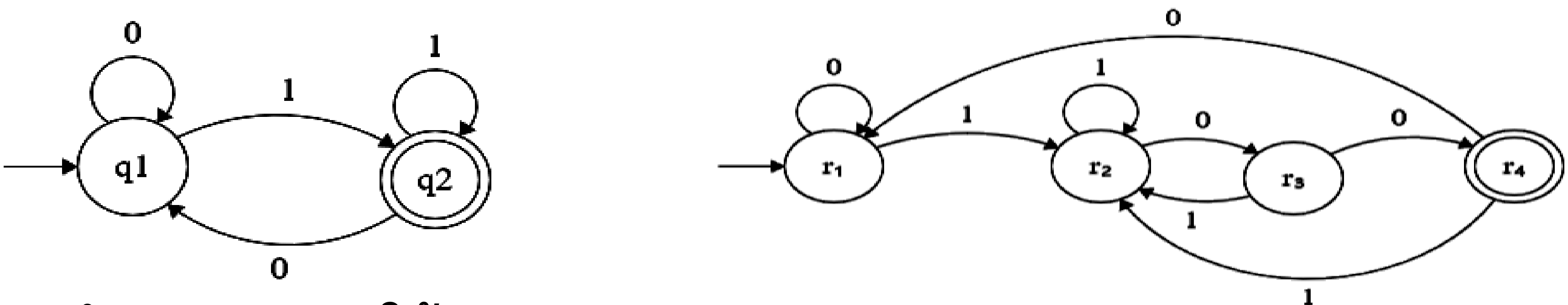
<u>แนวความคิด</u>

หาก A_1และ A_2เป็น regular ก็จะมี M_1 ที่ recognize A_1 และ M_2 ที่ recognize A_2

ดังนั้นแสดงว่าเราต้องการทราบว่าจะมี finite automaton M ที่สามารถ recognize $A_1 \cup A_2$ ได้หรือไม่? นั่นเอง

มี finite automaton M ที่สามารถ recognize $A_1 \cup A_2$

- ให้ M_1 accept string ที่ลงท้ายด้วย 1
- ให้ M_2 accept String ที่ลงท้ายด้วย 100

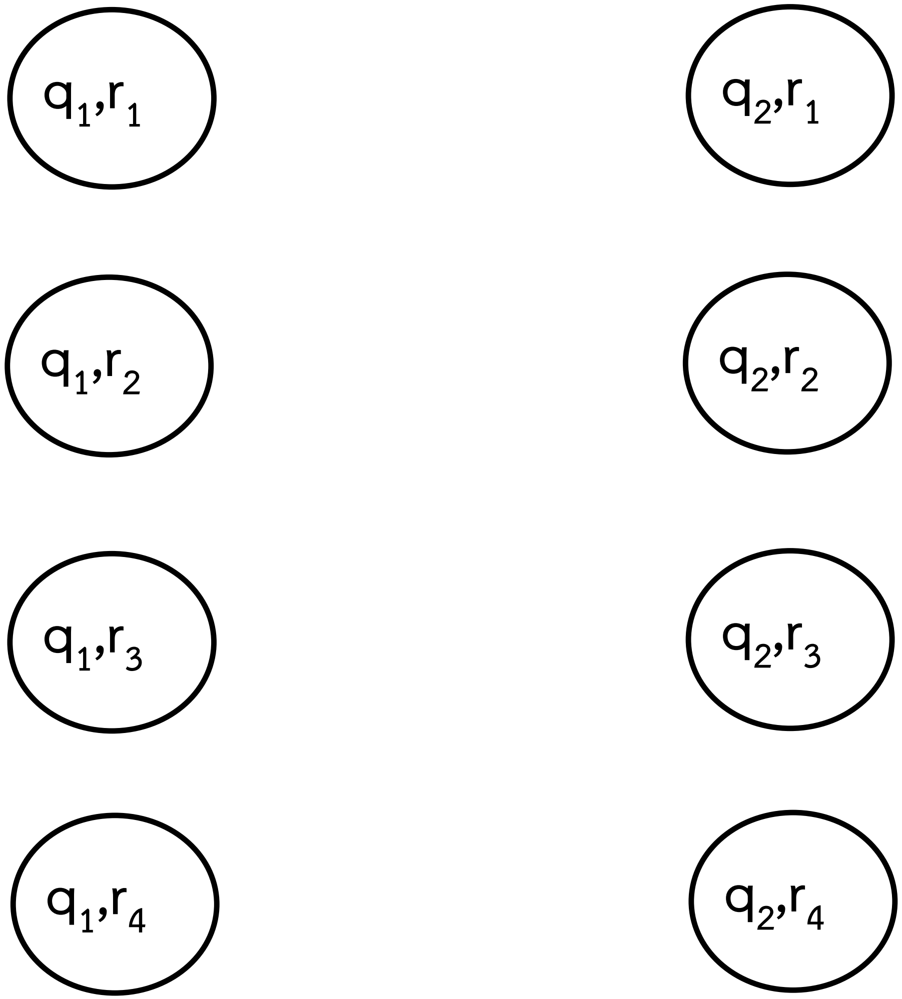


1. กำหนด State ให้ M

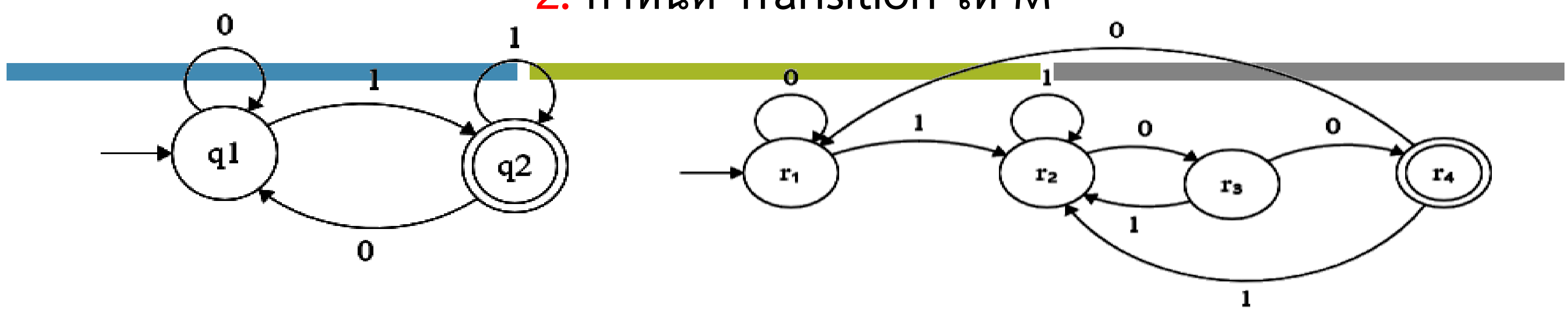
- ตัวแรกอยู่ที่ q_1 ตัวสองอาจจะอยู่ที่ r_1 หรือ r_2 หรือ r_3 หรือ r_4 ก็ได้
- ตัวแรกอยู่ที่ q_2 ตัวสองอาจจะอยู่ที่ r_1 หรือ r_2 หรือ r_3 หรือ r_4 ก็ได้

q_1,r_1	q_2,r_1
q_1,r_2	q_2,r_2
q_1,r_3	q_2,r_3
q_1,r_4	q_2,r_4

1. กำหนด State ให้ M



2. กำหนด Transition ให้ M



$\delta(q_1r_1,0) = q_1r_1$

$\delta(q_1r_1,1) = q_2r_2$

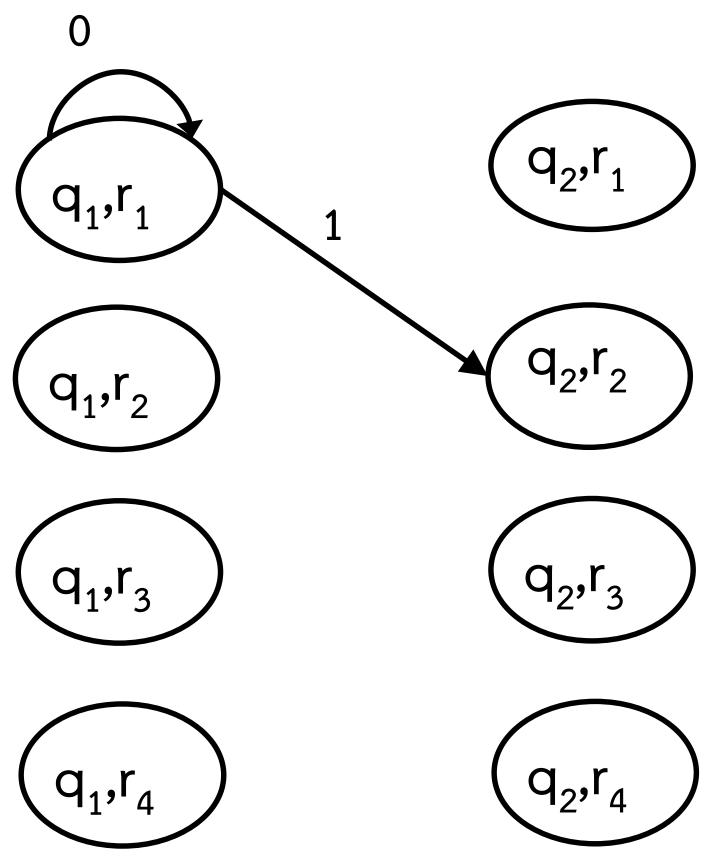
เมื่ออยู่ที่ state q_1 , r_1

รู้ว่า $\delta_1(q_1,0) = q_1$ และ $\delta_2(r_1,0) = r_1$

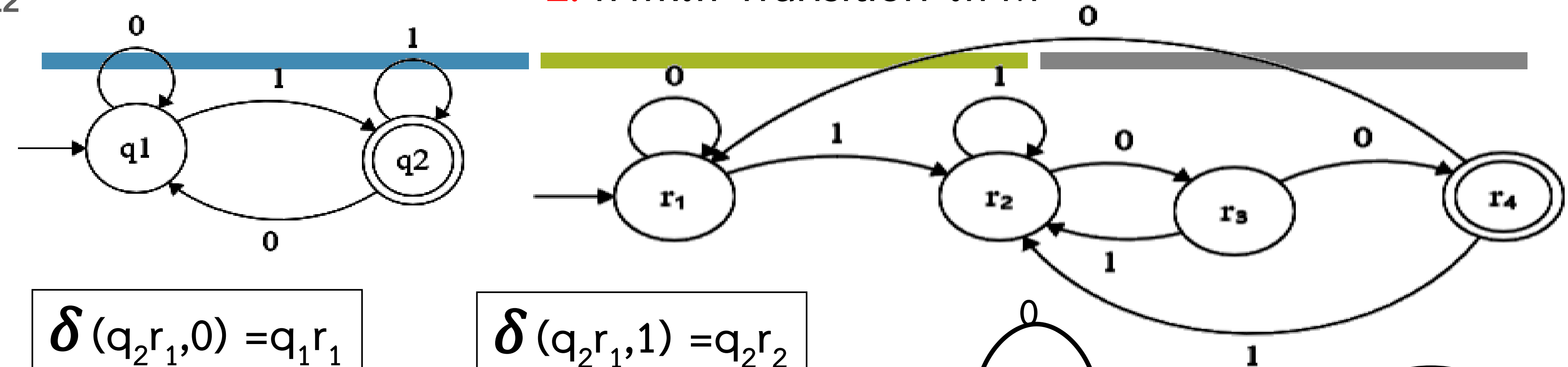
- ดังนั้น input เป็น 0 M_3 จะอยู่ state เดิม (q_1,r_1)

รู้ว่า $\delta_1(q_1,1) = q_2$ และ $\delta_2(r_1,1) = r_2$

- ดังนั้น input เป็น 1 M จะย้ายจาก state q_1,r_1 ไป q_2,r_2

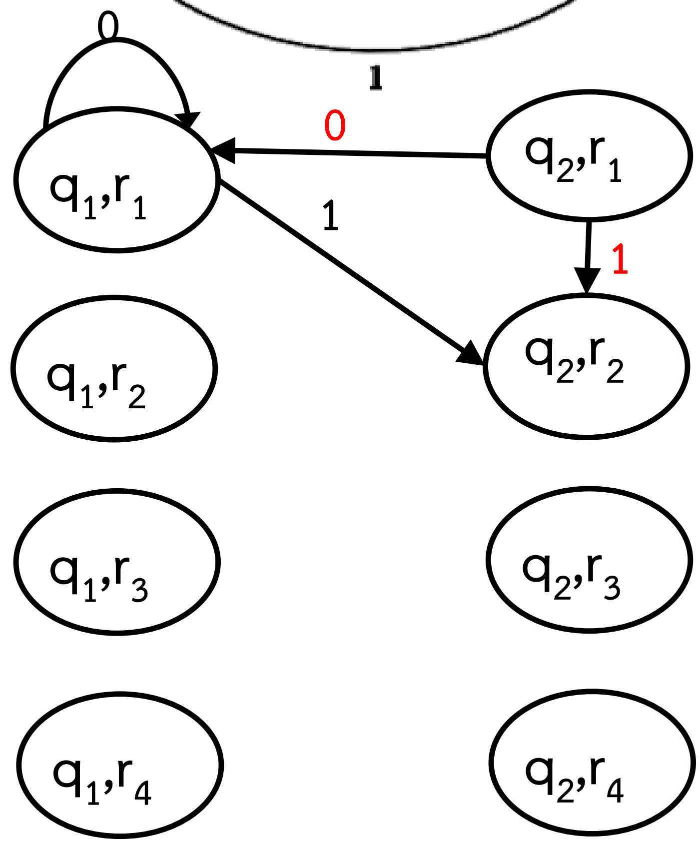


2. กำหนด Transition ให้ M



$\delta(q_2r_1,0) = q_1r_1$

$\delta(q_2r_1,1) = q_2r_2$



เมื่ออยู่ที่ state q_2 หรือ r_1

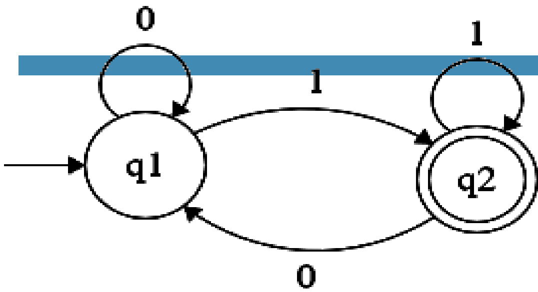
รู้ว่า $\delta_1(q_2,0) = q_1$ และ $\delta_2(r_1,0) = r_1$

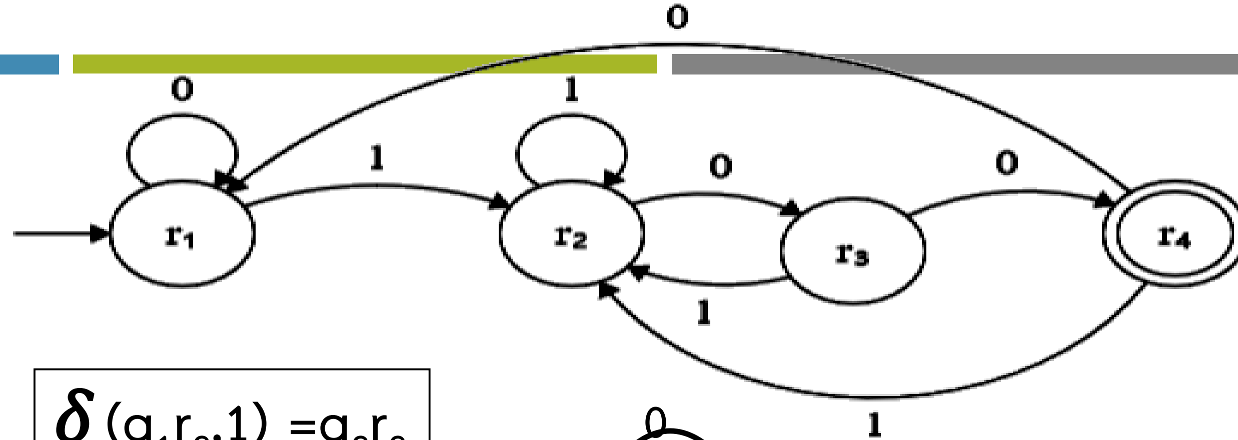
- ดังนั้น input เป็น 0 M_3 จะย้ายจาก state q_2,r_1 ไป state q_1,r_1

รู้ว่า $\delta_1(q_2,1) = q_2$ และ $\delta_2(r_1,1) = r_2$

- ดังนั้น input เป็น 1 M จะย้ายจาก state q_2,r_1 ไป q_2,r_2

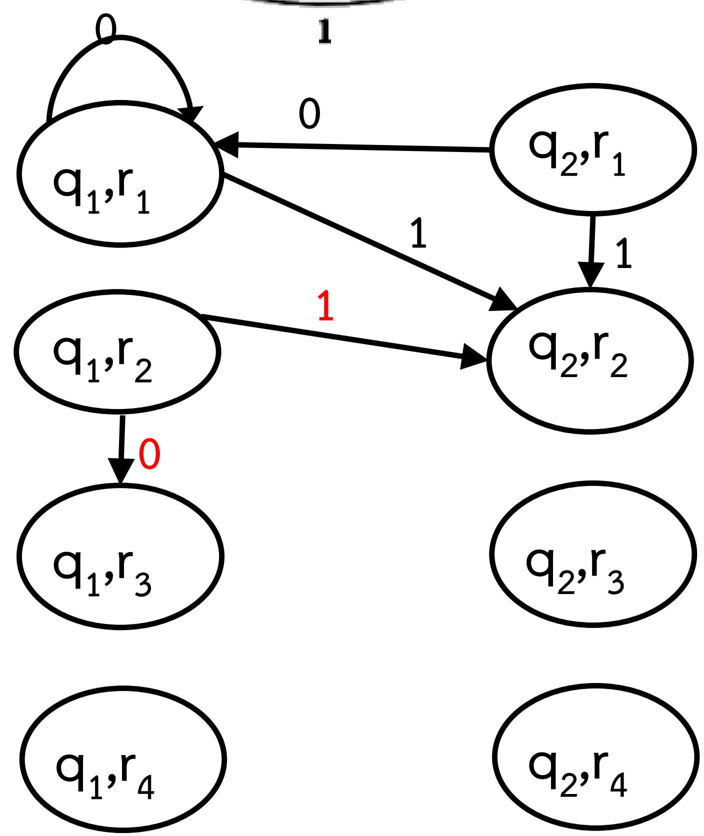
2. กำหนด Transition ให้ M





$\delta(q_1r_2,0) = q_1r_3$

$\delta(q_1r_2,1) = q_2r_2$



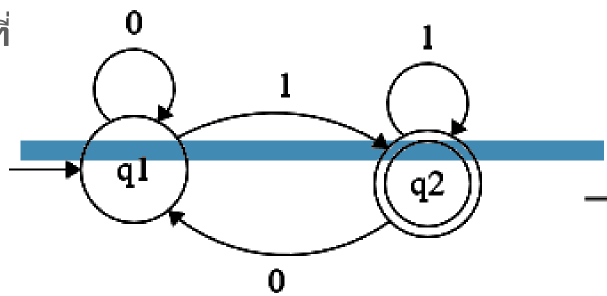
เมื่อยู่ที่ state q_1 , r_2

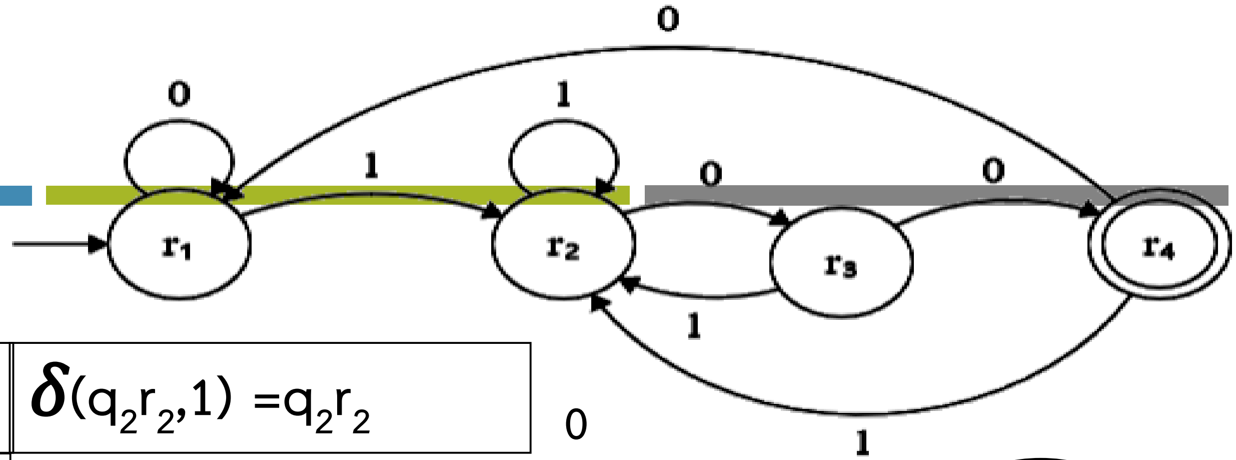
รู้ว่า $\delta_1(q_1,0) = q_1$ และ $\delta_2(r_2,0) = r_3$

- ดังนั้น input เป็น 0 M_3 จะย้ายจาก state q_1,r_2 ไป state q_1,r_3

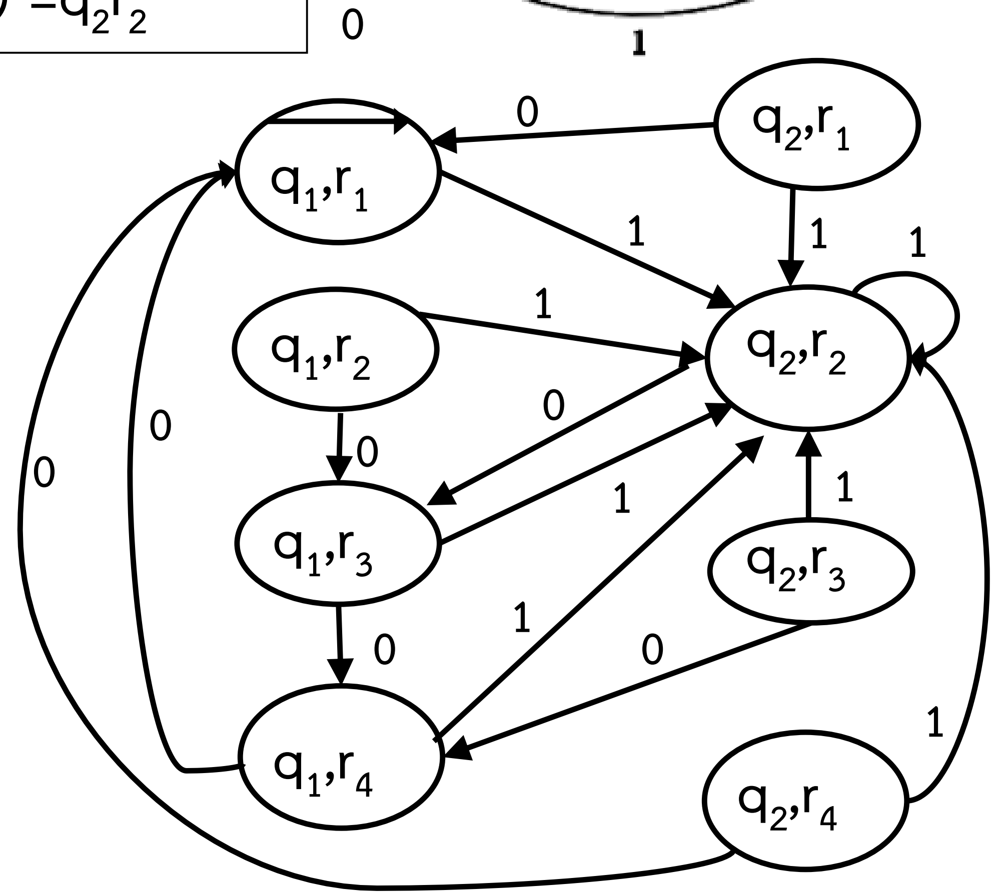
รู้ว่า $\delta_1(q_1,1) = q_2$ และ $\delta_2(r_2,1) = r_2$

- ดังนั้น input เป็น 1 M จะย้ายจาก state q_1,r_2 ไป q_2,r_2

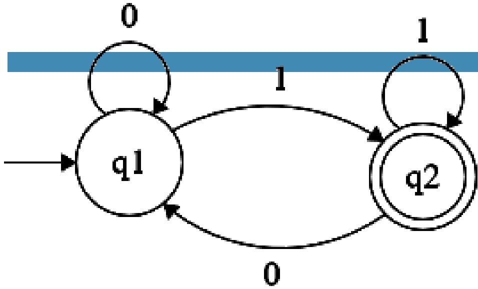


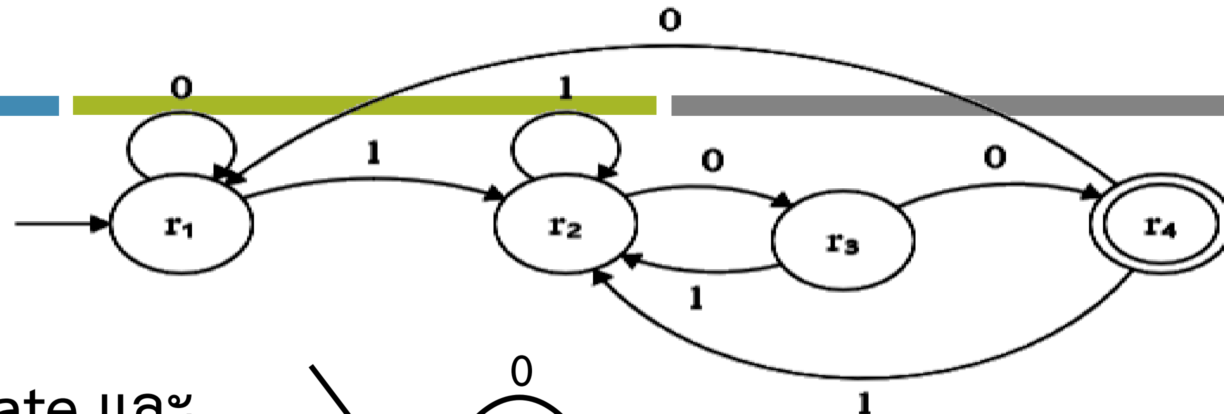


$\delta(q_2r_2,0) = q_1r_3$	$\delta(q_2r_2,1) = q_2r_2$
$\delta(q_1r_3,0) = q_1r_4$	
$\delta(q_1r_3,1) = q_2r_2$	
$\delta(q_2r_3,0) = q_1r_4$	
$\delta(q_2r_3,1) = q_2r_2$	
$\delta(q_1r_4,0) = q_1r_1$	
$\delta(q_1r_4,1) = q_2r_2$	
$\delta(q_2r_4,0) = q_1r_1$	
$\delta(q_2r_4,1) = q_2r_2$	

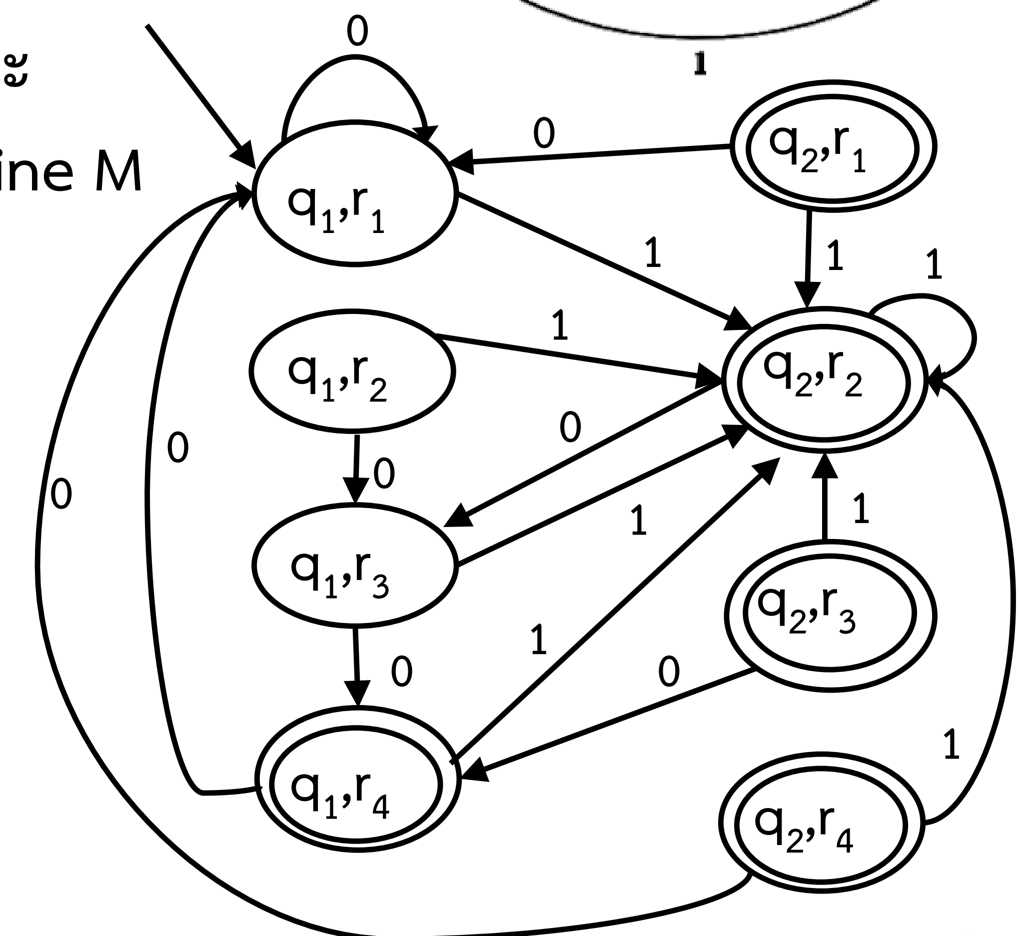


2. กำหนด Transition ให้ machine M





3. กำหนด starting state และ accept state ให้machine M



Regular operations

กำหนดให้ A_1 และ A_1 เป็น Regular languages โดยมี

- Machine M_1= $(Q_1,\Sigma,\delta_1,q_1,F_1)$ ที่ recognize A_1
- Machine M_2= $(Q_2,\Sigma,\delta_2,q_2,F_2)$ ที่ recognize A_2

สามารถจำลอง Machine M ที่ recognize A_1 U A_2 ดังนี้

- $M=(Q,\Sigma,\delta,q_0,F)$ โดย

$Q=Q_1 \times Q_2$

Σ ยังคงเหมือนเดิม

δ นิยามโดย $\delta((r_1,r_2),a) =(\delta_1(r_1,a),\delta_2(r_2,a))$

$q_0= (q_1,q_2)$

$F=\{(r_1,r_2)|r_1 \in F_1 \text{ or } r_2 \in F_2\}$

ดังนั้น A_1 U A_2 เป็น Regular languages

Regular operations

- จากข้างต้น สามารถจำลอง finite automaton (M) ได้จาก 2 finite automata (M1 และ M2)
- ดังนั้น set ของ Regular language นั้นมีคุณสมบัติปิดภายใต้ union