

บทที่ 2

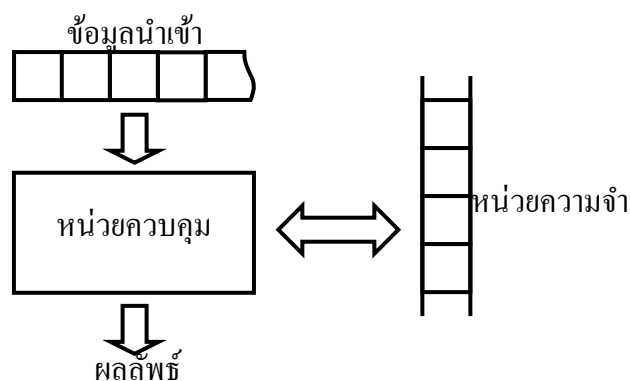
เทคนิคการแบ่งรูปภาพและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดของออโตมาตาแบบต่างๆ ที่นำมาใช้ประยุกต์เกี่ยวกับงานวิจัยนี้เพื่อใช้เป็นพื้นฐานสำหรับการถอดรหัสของรูปภาพดิจิทัล และในส่วนของ การเข้ารหัสรูปภาพดิจิทัลก็จะแบ่งเป็นเทคนิคการแบ่งรูปภาพดิจิทัล และการแก้ปัญหาสมการเชิงเส้นแบบต่างๆ

2.1 ออโตมาตา (Automata)

ออโตมาตาเป็นโมเดลทางคณิตศาสตร์ของเครื่องจักรที่มีสถานะจำกัด (Finite state machine) เครื่องจักรที่มีสถานะจำกัดคือเครื่องจักรที่เมื่อรับข้อมูลเข้ามาแล้วจะมีการกระโดดหรือย้ายไปมาระหว่างสถานะต่างๆ ที่ได้กำหนดไว้ในฟังก์ชันการเปลี่ยนแปลงหรือฟังก์ชันทางผ่าน (transition function) ซึ่งเป็นหัวใจสำคัญสำหรับออโตมาตา เพราะเก็บข้อมูลนำเข้าและสถานะถัดไปของแต่ละสถานะเอาไว้ ข้อมูลที่ป้อนเข้าไปในเครื่องจักรจะถูกอ่านทีละตัวอักษรจนกระทั่งอ่านครบทุกตัว ยกตัวอย่างเช่น ถ้าหากข้อมูลที่ป้อนเข้าไปเป็นข้อความ ข้อความนี้จะถูกแบ่งเป็นตัวอักษร แล้วจึงจะนำเข้าไปในออโตมาตาทีละตัวอักษรซึ่งจะอ่านเฉพาะข้อมูลที่ได้รับการยอมรับว่าเป็นชนิดข้อมูลที่ออโตมาตายอมรับ โดยจะอ่านจากทางด้านซ้ายไปยังทางด้านขวาจนกระทั่งครบทุกตัวอักษรที่อยู่ในข้อความ เมื่อข้อมูลถูกอ่านจนจบออโตมาตาก็จะหยุดทำงาน ในสถานะสุดท้ายของออโตมาตานั้น ถ้าเป็นสถานะที่ยอมรับได้ ก็จะกล่าวว่าออโตมาตายอมรับข้อมูลนี้ แต่ถ้าหากอยู่ในสถานะที่ยอมรับไม่ได้ ก็จะถือว่าไม่ยอมรับข้อมูลนี้

เนื่องจากตัวออโตมาตาเป็นเครื่องจักรดังนั้น ออโตมาตาจึงจำเป็นจะต้องมีอุปกรณ์หน่วยความจำ (storage device) ไว้สำหรับเก็บข้อมูลนำเข้าซึ่งแต่ละหน่วยของอุปกรณ์หน่วยความจำจะถือว่าเก็บข้อมูลอยู่หน่วยละ 1 ตัวอักษร ซึ่งสามารถที่จะอ่านและทำการเปลี่ยนแปลงค่าที่อยู่ในอุปกรณ์หน่วยความจำนี้ได้ ส่วนออโตมาตาจะมีหน่วยควบคุม (control unit) ซึ่งจะเก็บสถานะภายใน (internal state) อยู่ ดังรูป 2.1 ซึ่งแสดงโครงสร้างโดยทั่วไปของออโตมาตาที่เมื่อมีข้อมูลนำเข้ามาแล้วจะถูกหน่วยควบคุมนำไปเก็บไว้ในหน่วยความจำ และเมื่อออโตมาตาในหน่วยควบคุมต้องการที่จะใช้ข้อมูลนำเข้าก็จะดึงข้อมูลนำเข้ามาจากหน่วยความจำทีละตัวอักษรจนกระทั่งอ่านข้อมูลนำเข้าครบทุกตัวอักษรของข้อมูลนำเข้าแล้ว ก็จะพิจารณาสถานะสุดท้ายว่าเป็นสถานะที่ออโตมาตานี้ยอมรับหรือไม่ แล้วจึงส่งค่ายอมรับหรือไม่ยอมรับออกมาเป็นผลลัพธ์



รูปที่ 2.1 โครงสร้างโดยทั่วไปของออโตมาตา

คุณลักษณะโดยทั่วไปของออโตมาตาดังนี้

- ประกอบด้วยสถานะ (states) ฟังก์ชันการเปลี่ยนสถานะ (transition function) สถานะเริ่มต้น (initial states) และสถานะการยอมรับ (accepting states)
- รับอินพุตจากภายนอกระบบเข้าอย่างต่อเนื่อง เรียกอินพุตที่รับเข้ามานี้ว่าตัวอักษร (alphabets)
- ลำดับของตัวอักษรที่เป็นอินพุตซึ่งรับเข้ามาเรื่อยๆ นั้นจะเป็นสายอักขระที่เรียกว่า คำ (words)
- มีการเปลี่ยนสถานะตามที่กำหนดโดยฟังก์ชันการเปลี่ยนสถานะ ซึ่งจะขึ้นอยู่กับตัวอักษรที่รับอินพุตเข้ามา
- เมื่อหยุดการรับอินพุต หากออโตมาตาอยู่ในสถานะการยอมรับถือว่าออโตมาตายอมรับคำที่เป็นอินพุตนั้น แต่ถ้าออโตมาตาอยู่นอกสถานะการยอมรับ ถือว่าออโตมาตาปฏิเสธคำที่เป็นอินพุตนั้น
- เซตของคำทั้งหมดที่ออโตมาตานั้นยอมรับเรียกว่า ภาษา (language) ซึ่งยอมรับโดยออโตมาตา

เนื่องจากออโตมาตามีสถานะที่จำกัดดังนั้นจึงอาจเรียกได้ว่า ออโตมาตาจำกัด (Finite Automata, FA) ออโตมาตาสามารถที่จะแบ่งออกได้เป็น 2 แบบใหญ่ๆ คือ ออโตมาตาจำกัดเชิงกำหนด (deterministic finite automata, DFA) กับออโตมาตาจำกัดเชิงไม่กำหนด (nondeterministic finite automata, NFA) และจะมีการประยุกต์ออโตมาตาเป็นแบบถ่วงน้ำหนักคือ ออโตมาตาแบบถ่วงน้ำหนัก (weighted finite automata, WFA) โดยออโตมาตาจะมีข้อมูลนำเข้าชนิดเดียวกันก็คือ มีลักษณะเป็นสายอักขระ

2.1.1 สายอักขระ (String) และภาษา (Language)

ชุดตัวอักษร คือเซตจำกัดของสัญลักษณ์ซึ่งจะไม่ใช่เซตว่าง โดยจะใช้สัญลักษณ์ Σ แทนชุดตัวอักษรใดๆ เช่น $\Sigma=\{0, 1\}$ หรือ $\Sigma=\{a, b, c\}$ เป็นต้น สายอักขระ จากชุดตัวอักษร Σ คือลำดับจำกัดของสัญลักษณ์จาก Σ ซึ่งเขียนติดกันโดยไม่มีช่องว่างและไม่มีเครื่องหมายใดๆ มากัน เช่น $\Sigma=\{0, 1\}$ จะสามารถสร้างสายอักขระ 1100 ได้ เป็นต้น

ถ้ากำหนดให้ w เป็นสายอักขระใดๆ ความยาว (length) ของ w เขียนแทนด้วย $|w|$ คือจำนวนตัวอักษรทั้งหมดที่ปรากฏอยู่ในสายอักขระ w เช่น ถ้า $w=10011$ แล้ว $|w|=5$

สายอักขระว่าง (empty string) เขียนแทนด้วย ϵ คือสายอักขระที่ไม่มีสัญลักษณ์ใดๆ ปรากฏเลย ดังนั้น $|\epsilon|=0$

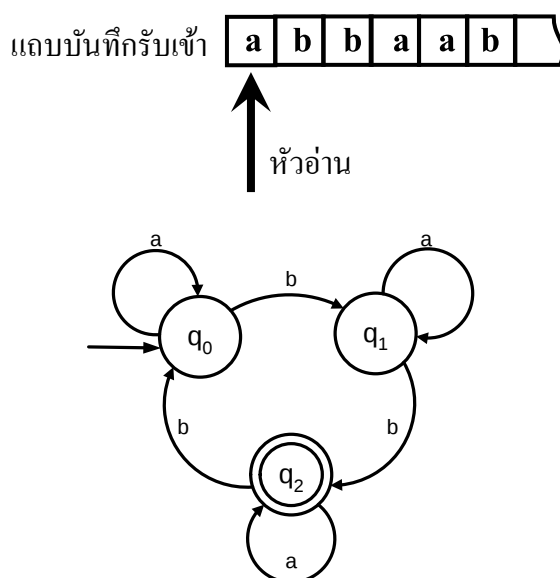
ถ้า w และ v เป็นสายอักขระจากชุดตัวอักษรเดียวกันจะสามารถสร้างสายอักขระใหม่จากการนำ w และ v ได้โดยการนำมาต่อกัน (concatenate) จะได้ wv ดังนั้น $w^5=wwwww$ โดยจะสามารถหาความยาวของสายอักขระจะเท่ากับ $|wv|=|w|+|v|$ และกำหนดให้ $w^0=\epsilon$ และ $w^1=w$

สำหรับชุดตัวอักษร Σ ใดๆ สัญลักษณ์ Σ^* จะหมายถึงเซตของสายอักขระทั้งหมดจาก Σ ซึ่งจะรวมถึงสายอักขระว่างด้วย และสัญลักษณ์ Σ^+ จะหมายถึง $\Sigma^* - \{\epsilon\}$

กำหนดให้ L เป็น ภาษา (Language) จาก Σ ก็ต่อเมื่อ $L \subseteq \Sigma^*$ ตัวอย่างเช่น $\Sigma=\{0, 1\}$ จะสามารถสร้างภาษาได้เช่น $\epsilon, \{0, 1\}, \{00, 1, 01\}$ เป็นต้น และเรียกสมาชิกทุกตัวใน L ว่า ประโยค (sentence) ของ L

2.1.2 ออโตมาตาเชิงกำหนด (Deterministic Finite Automata, DFA)

ลักษณะของออโตมาตาเชิงกำหนด จะรับข้อมูลเข้าเป็นสายอักขระผ่านทาง แถบบันทึกรับเข้า (input tape) ที่จะอ่านข้อมูลเข้าไปทีละตัวจากทางซ้ายมือโดยหัวอ่านดังรูปที่ 2.2 และเขียนออโตมาตาเชิงกำหนด อยู่ในรูปของกราฟบ่งบอกทิศทาง (directed graph) โดยจะเริ่มจากสถานะเริ่มต้น(initial state, q_0) โดยในการย้ายสถานะแต่ละครั้ง ออโตมาตาเชิงกำหนด จะอยู่ในสถานะใดสถานะหนึ่งเสมอ เช่นในรูปที่ 2.2 แสดงให้เห็นถึงออโตมาตาที่กำลังอยู่ในสถานะที่ q_0 กำลังจะอ่านข้อมูลนำเข้า a เมื่ออ่านเสร็จแล้วหัวอ่านจะย้ายไปทางขวา 1 ช่อง และออโตมาตาเชิงกำหนด จะย้ายสถานะไป q_1 ซึ่งในออโตมาตาเชิงกำหนด ใดๆ อาจจะมีการย้ายสถานะหรือไม่ย้ายก็ได้ขึ้นอยู่กับตัวอักษรที่เป็นข้อมูลนำเข้าและลักษณะของฟังก์ชันทางผ่าน เมื่อหัวอ่านทำการอ่านข้อมูลนำเข้าทีละ 1 ตัวจนหมดสายอักขระ แล้วออโตมาตาเชิงกำหนด จะบ่งบอกว่าสายอักขระนี้ยอมรับโดยออโตมาตาเชิงกำหนด หรือไม่ โดยดูได้จากสถานะสุดท้ายของออโตมาตาเชิงกำหนดว่าเป็นสถานะที่ยอมรับ (accepting state) ได้หรือไม่ ในรูปที่ 2.2 สถานะที่ยอมรับจะถูกแทนด้วยวงกลม 2 วงซ้อนกัน ก็คือสถานะ q_2

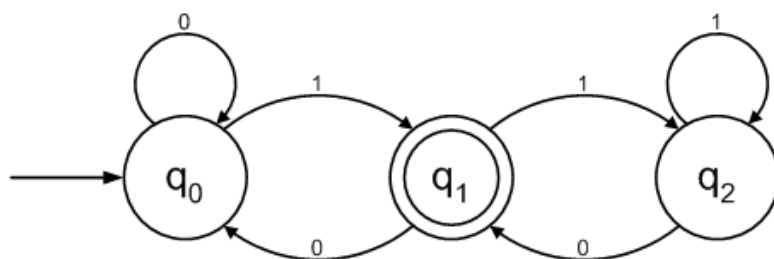


รูปที่ 2.2 แสดงตัวอย่างออโตมาตาเชิงกำหนด

ส่วนประกอบของออโตมาตาเชิงกำหนด M ใดๆ จะประกอบไปด้วยส่วนประกอบ 5 ส่วนคือ $M=(Q,\Sigma,\delta,q_0,F)$ เมื่อ

- Q เป็นเซตจำกัดของสถานะภายใน
- Σ เป็นเซตจำกัดของสัญลักษณ์นำเข้าที่เรียกว่าตัวอักษรนำเข้า (input alphabet)
- δ เป็นฟังก์ชันการเปลี่ยนสถานะของแต่ละสถานะ โดยเป็นฟังก์ชันจาก $Q \times \Sigma$ ไปยัง Q เขียนแทนด้วย $\delta : Q \times \Sigma \rightarrow Q$
- q_0 เป็นสถานะเริ่มต้น
- F เป็นเซตของสถานะสุดท้ายที่เป็นสถานะที่มีการยอมรับสายอักขระ

ตัวอย่างที่ 2.1 จากรูปที่ 2.3 กำหนดให้แทนออโตมาตา M



รูปที่ 2.3 ออโตมาตา M ซึ่งเขียนอยู่ในรูปของกราฟบ่งบอกทิศทาง

จากรูปที่ 2.3 จะแสดงถึงออโตมาตา M ซึ่งเป็นอยู่ในรูปของกราฟบ่งบอกทิศทาง ซึ่งจะกำหนดให้แต่ละจุดยอด (vertex) หรือ โหนดแทนแต่ละสถานะ โดยที่สถานะเริ่มต้นจะมีลูกศรชี้อยู่ และสถานะสุดท้ายทุกจุดใน F จะแทนด้วยวงกลม 2 วงซ้อนกัน แต่ละขอบ (edge) ให้แทนฟังก์ชันการเปลี่ยนสถานะ ซึ่งเชื่อมจากจุด p ไปยังจุด q โดยมีสัญลักษณ์นำเข้าเป็นฉลาก (label) เขียนกำกับ ซึ่งสามารถเขียนแจกแจงรายละเอียดออโตมาตาได้ดังนี้ $M = \{q_0, q_1, q_2\}, \{0, 1\}, \delta, \{q_0\}, \{q_1\}$ ซึ่ง δ หรือ ฟังก์ชันการเปลี่ยนสถานะ สามารถเขียนแสดงเป็นตารางได้ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงฟังก์ชันการเปลี่ยนสถานะของออโตมาตา M

δ	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_1	q_2

ในการแสดงฟังก์ชันการเปลี่ยนแปลงของแต่ละสถานะจะอยู่ในรูป

$$\delta(q_i, a) = q_j$$

เมื่อ $q_i, q_j \in Q$; $i=0, 1, 2$ และ $a \in \{0, 1\}$

เช่น $\delta(q_0, 1) = q_1$ แต่เนื่องจากในการรับข้อมูลเข้านั้นไม่ได้รับเพียงแค่สัญลักษณ์นำเข้าเพียงตัวเดียว แต่เป็นสายอักขระ ดังนั้นจึงต้องเขียนให้อยู่ในรูป $\delta^*(q_i, w)$ โดยที่ $q_i \in Q$, $w \in \Sigma^*$ ซึ่งสัญลักษณ์ดาว (star) จะบ่งบอกว่าข้อมูลนำเข้าที่กำลังอ่านอยู่ ยังมีลักษณะเป็นสายอักขระ ซึ่งเมื่อทำการย้ายสถานะเรียบร้อยแล้วก็ให้อ่านข้อมูลนำเข้าตัวต่อไป โดยฟังก์ชันการเปลี่ยนแปลงถัดไปจะมีสถานะเป็น $\delta^*(q_j, x)$ เมื่อ $\delta(q_i, a) = q_j$ โดยที่ $q_i, q_j \in Q$, $w, a \in \Sigma^*$, $x \in \Sigma$ และ $ax = w$ ซึ่งจะเขียนอยู่ในรูปทั่วไปได้โดย

$$\delta^*(q_i, w) = \delta^*(q_j, x)$$

ในการเริ่มตรวจสอบข้อความว่าออโตมาตาจะยอมรับข้อความนี้หรือไม่ จะเริ่มต้นที่โหนดเริ่มต้น (สถานะเริ่มต้น) และจะอ่านข้อมูลเข้ามาทีละตัวอักษรจากทางด้านซ้ายไปเรื่อยๆ จนกระทั่งครบทุกตัวอักษร เช่น ข้อความ 0111

$$\begin{aligned} & \delta^*(q_0, 0111) \\ &= \delta^*(q_0, 111) \\ &= \delta^*(q_1, 11) \\ &= \delta(q_2, 1) \\ &= q_2 \end{aligned}$$

ซึ่งจะเห็นว่าสถานะสุดท้ายเป็นโหนด q_2 แต่โหนด q_2 ไม่ได้เป็นสถานะสุดท้ายที่จะแสดงว่ายอมรับข้อความนี้ ดังนั้นสำหรับข้อความนี้จึงไม่ยอมรับ โดยออโตมาตา M ข้อความ 01111001

$$\begin{aligned} & \delta^*(q_0, 01111001) \\ &= \delta^*(q_0, 1111001) \end{aligned}$$

$$\begin{aligned}
&= \delta^*(q_1, 111001) \\
&= \delta^*(q_0, 11001) \\
&= \delta^*(q_1, 1001) \\
&= \delta^*(q_2, 001) \\
&= \delta^*(q_1, 01) \\
&= \delta(q_0, 1) \\
&= q_1
\end{aligned}$$

จะเห็นข้อความ 01111001 สถานะสุดท้ายจะอยู่ที่ โหนด q_1 ซึ่งเป็นสถานะสุดท้าย ดังนั้นออโตมาตา M จึงยอมรับข้อความนี้

2.1.3 ออโตมาตาจำกัดเชิงไม่กำหนด (Nondeterministic Finite Automata, NFA)

ลักษณะของออโตมาตาจำกัดเชิงไม่กำหนด หรือ เอ็นเอฟเอ จะมีคล้ายกับออโตมาตาเชิงกำหนด จะต่างกันก็เพียงฟังก์ชันทางผ่าน โดยจะสังเกตเห็นว่าออโตมาตาเชิงกำหนด มีข้อมูลออกของฟังก์ชันทางผ่านเป็นสถานะได้เพียงสถานะเดียวเท่านั้น แต่ของเอ็นเอฟเอนั้นจะมีข้อมูลออกของฟังก์ชันทางผ่านเป็นเซตกำลัง (power set) ของ Q หรือ $P(Q)$

ส่วนประกอบของออโตมาตาเชิงไม่กำหนด N ใดๆ จะประกอบไปด้วยส่วนประกอบ 5 ส่วนเช่นกันคือ $N=(Q, \Sigma, \delta, q_0, F)$ เมื่อ

Q เป็นเซตจำกัดของสถานะภายใน

Σ เป็นเซตจำกัดของสัญลักษณ์นำเข้าที่เรียกว่าตัวอักษรนำเข้า

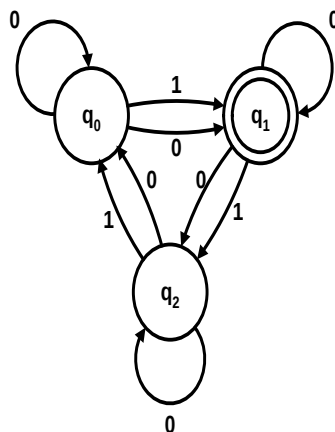
δ เป็นฟังก์ชันจาก $Q \times \Sigma$ ไปยัง $P(Q)$ เขียนแทนด้วย $\delta : Q \times \Sigma \rightarrow P(Q)$

q_0 เป็นสถานะเริ่มต้น

F เป็นเซตของสถานะสุดท้ายที่เป็นสถานะที่มีการยอมรับสายอักขระ

จากฟังก์ชันทางผ่านจะเห็นได้ว่า $\delta(q, x)$ มีได้มากกว่าหนึ่งสถานะ เมื่อ $q \in Q, x \in \Sigma$ ดังนั้นแต่ละข้อมูลนำเข้าของเอ็นเอฟเอ จึงมีเส้นทางผ่านได้หลายเส้นทาง ซึ่งอาจที่จะมีบางเส้นทางที่ทำให้สามารถไปถึงสถานะสุดท้ายที่มีการยอมรับสายอักขระนี้ได้ ก็จะถือว่าเอ็นเอฟเอนี้ยอมรับสายอักขระนี้ ดังตัวอย่างที่ 2.2

ตัวอย่างที่ 2.2 จากรูปที่ 2.4 แสดงถึงเอ็นเอฟเอ N



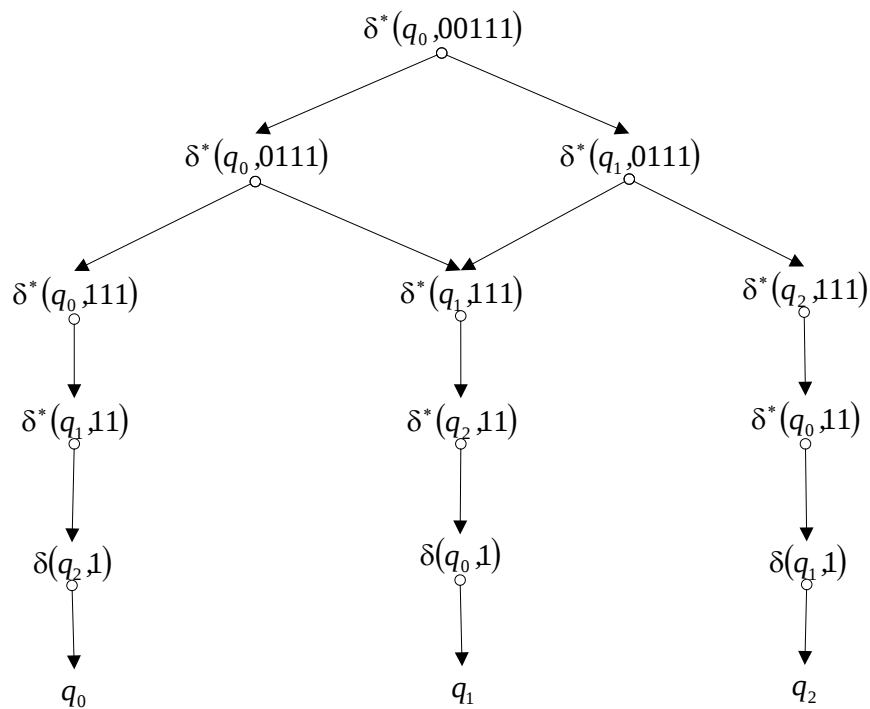
รูปที่ 2.4 แสดงถึงเอ็นเอฟเอ N

การแจกแจงสมาชิกของออโตมาตาคำจำกัดซึ่งไม่กำหนดจะเขียนได้ดังนี้ $N = \{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, \{q_0\}, \{q_1, q_2\}$ โดยที่ฟังก์ชันทางผ่านจะเป็นดังตารางที่ 2.2

ตารางที่ 2.2 แสดงถึงฟังก์ชันทางผ่านของเอ็นเอฟเอ N

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

จะสังเกตได้ว่ามีฟังก์ชันทางผ่านบางคู่ที่ได้ผลลัพธ์เป็นสถานะ 2 สถานะ เช่น $\delta(q_0, 0) = \{q_0, q_1\}$ ดังนั้นแต่ละข้อมูลนำเข้าจึงอาจจะสามารถสร้างทางผ่านได้หลายเส้นทาง แต่ขอเพียงแค่มีทางผ่านทางเดียวที่สามารถไปยังสถานะที่ยอมรับได้ ก็จะถือว่าสายอักขระนั้นยอมรับโดยเอ็นเอฟเอ N เช่น 00111 จะสามารถสร้างทางผ่านได้ดังรูปที่ 2.5



รูปที่ 2.5 แสดงเส้นทางผ่านของเอ็นเอฟเอ N

จะเห็นว่าสายอักขระ 00111 สามารถสร้างทางผ่านได้หลายเส้นทางแต่มีอยู่เส้นทางบางเส้นทางที่สามารถให้สถานะที่ยอมรับได้คือ

$$\begin{aligned}
 &\delta^*(q_0, 00111) \\
 &= \delta^*(q_1, 0111) \\
 &= \delta^*(q_1, 111) \\
 &= \delta^*(q_2, 11) \\
 &= \delta(q_0, 1) \\
 &= q_1
 \end{aligned}$$

หรือ

$$\begin{aligned}
 &\delta^*(q_0, 00111) \\
 &= \delta^*(q_0, 0111) \\
 &= \delta^*(q_1, 111) \\
 &= \delta^*(q_2, 11) \\
 &= \delta(q_0, 1) \\
 &= q_1
 \end{aligned}$$

ก็จะถือว่าสายอักขระหรือข้อมูลนำเข้า 00111 นี้ยอมรับโดยเอ็นเอฟเอ N

2.2 สมการเชิงเส้น (Linear Equation)

ในส่วนนี้จะกล่าวถึงส่วนที่เกี่ยวข้องกับการหาค่าถ่วงน้ำหนักของออโตมาตาซึ่งจะใช้ความรู้ทางคณิตศาสตร์ในการแก้สมการ

2.2.1 ระบบสมการเชิงเส้น (System of Linear Equation)

เมื่อกล่าวถึงสมการโดยทั่วไปไปที่กำลังของตัวแปร (variable) หรือตัวไม่ทราบมีกำลังเป็น 1 แล้ว สมการจะอยู่ในรูปของสมการเชิงเส้น ดังนี้

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \quad (2.1)$$

เมื่อ $a_i, b \in \mathbb{R}$ และ x_i เป็นตัวแปรที่ยังไม่รู้ค่า ซึ่งจะเรียก $a_1, a_2, \dots, a_n$ ว่าเป็นสัมประสิทธิ์ของ $x_1, x_2, \dots, x_n$ ตามลำดับ และเรียก b ว่าเป็นค่าคงตัวของสมการ ในกรณีที่กำหนดค่าคงที่ให้กับตัวแปรทุกตัวแล้วทำให้สมการที่ 2.1 เป็นจริง หรือจะกล่าวว่า ถ้ากำหนดให้

$$x_1 = k_1, x_2 = k_2, \dots, x_n = k_n \quad \text{เมื่อ } k_i \text{ เป็นค่าคงที่}$$

จะทำให้

$$a_1k_1 + a_2k_2 + \dots + a_nk_n = b$$

จะเรียกเซต (set) ของ $\{k_1, k_2, \dots, k_n\}$ ว่า ผลเฉลย (solution) ของสมการที่ 2.1 ซึ่งผลเฉลยของสมการอาจจะมีได้มากกว่า 1 เซต

ในกรณีที่สมการที่ 2.1 มี m สมการแต่มีตัวแปรชุดเดียวกัน ดังนี้

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n &= b_3 \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned} \quad (2.2)$$

จะเรียกสมการทุกสมการรวมกันว่า ระบบสมการเชิงเส้น (System of Linear Equation) และจะเรียกผลเฉลยของระบบสมการเชิงเส้นว่า เซตผลเฉลย (solution set) หรือ ผลเฉลยทั่วไป (general solution) ในกรณีที่ $b_1, b_2, \dots, b_m$ มีค่าเท่ากับ 0 ทั้งหมดจะเรียกระบบสมการนี้ว่าเป็น เอกพันธ์ (homogeneous) ซึ่งจะเขียนได้อยู่ในรูปของ สมการที่ 2.3

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= 0 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= 0 \\ &\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= 0 \end{aligned} \quad (2.3)$$

ซึ่งระบบสมการเอกพันธ์นี้จะมีเซตของผลเฉลยเสมออย่างน้อย 1 เซต คือ $\{0, 0, \dots, 0\}$ ซึ่งจะมีชื่อเฉพาะสำหรับเรียกผลเฉลยนี้ว่า ผลเฉลยที่เป็นศูนย์ หรือผลเฉลยที่มีความสำคัญน้อย (trivial solution)

ในกรณีที่นำค่าคงที่ c_i คู่กับสมการที่ i ของระบบสมการ 2.2 ซึ่งจะได้ดังนี้

[illegible]

แล้วนำแต่ละสมการมาบวกกันแล้วจัดรูปจะได้ดังสมการที่ 2.4 ดังนี้

$$\begin{aligned} & (c_1 a_{11} + c_2 a_{21} + c_3 a_{31} + \dots + c_m a_{m1}) x_1 + \\ & (c_1 a_{12} + c_2 a_{22} + c_3 a_{32} + \dots + c_m a_{m2}) x_2 + \\ & \dots \dots \dots + (c_1 a_{1n} + c_2 a_{2n} + c_3 a_{3n} + \dots + c_m a_{mn}) x_n \\ & = c_1 b_1 + c_2 b_2 + c_3 b_3 + \dots + c_m b_m \end{aligned} \quad (2.4)$$

จะเรียกสมการในลักษณะนี้ว่า การรวมเชิงเส้น (linear combination) ซึ่งจะสามารถกล่าวได้ ผลเฉลยของระบบสมการเชิงการใดๆ จะเป็นผลเฉลยของการรวมเชิงเส้นของระบบสมการนั้นด้วย หรือจะกล่าวว่าถ้าเซตของ $\{k_1, k_2, \dots, k_n\}$ เป็นผลเฉลยของสมการ 2.2 แล้วจะเป็นผลเฉลยของสมการ 2.4 ด้วย ซึ่งจะแสดงได้ดังนี้

จาก $\{x_1, x_2, \dots, x_n\} = \{k_1, k_2, \dots, k_n\}$ ดังนั้น

[illegible]

หรือ $a_{i1}k_1 + a_{i2}k_2 + \dots + a_{in}k_n = b_i$ เมื่อ $i = 1, 2, 3, \dots, m$
 จะต้องแสดงว่า

$$(c_1 a_{11} + \dots + c_m a_{m1})k_1 + \dots + (c_1 a_{1n} + \dots + c_m a_{mn})k_n = c_1 b_1 + \dots + c_m b_m$$

จัดรูปแบบพจน์ทางด้านซ้ายมือของเครื่องเท่ากับใหม่จะได้ว่า

$$c_1(a_{11}k_1 + \dots + a_{1n}k_n) + c_2(a_{21}k_1 + \dots + a_{2n}k_n) + \dots + c_m(a_{m1}k_1 + \dots + a_{mn}k_n)$$

จากสมการ 2.5 จะได้ว่าเท่ากับ $c_1 b_1 + c_2 b_2 + c_3 b_3 + \dots + c_m b_m$

นอกจากนี้ยังสามารถที่จะรูปแบบสมการ 2.5 ให้อยู่ในรูปเมตริกซ์ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.6)$$

ซึ่งสามารถเขียนอยู่ในรูปย่อคือ $[A] \times [X] = [B]$ โดยที่ $[A]$ เป็นเมตริกซ์ขนาด $m \times n$ และ $[X]$, $[B]$ เป็นเมตริกซ์คอลัมน์ แต่อาจจะเรียกได้อีกอย่างหนึ่งก็คือจัดให้เมตริกซ์ A อยู่ในรูปของเซตของเวกเตอร์คอลัมน์ A หรือ $A = \{A_1, A_2, \dots, A_n\}$ และ เวกเตอร์คอลัมน์ X กับเวกเตอร์คอลัมน์ B

2.2.2 การแยกส่วนประกอบด้วยคิวอาร์ (QR Factorization)

การแยกส่วนประกอบด้วยคิวอาร์ นิยมใช้ในการคำนวณหาค่าไอเกนของเมตริกซ์ (eigenvalues of a matrix) ในการแก้ระบบสมการเชิงเส้น (linear systems)

ถ้า A เป็นเมตริกซ์ที่มีขนาด $m \times n$ ซึ่งแต่ละคอลัมน์ (columns) เป็นอิสระต่อกัน แล้ว A สามารถแยกส่วนประกอบได้เป็น $A = QR$ เมื่อ Q เป็นเมตริกซ์ที่มีขนาดเป็น $m \times n$ โดยแต่ละคอลัมน์มีขนาดเป็น 1 และเป็นอิสระต่อกันและ R เป็นเมตริกซ์สามเหลี่ยมบนที่มีขนาดเท่ากับ $n \times n$

ขั้นตอนในการหาส่วนประกอบด้วยคิวอาร์

ขั้นที่ 1: หาเมตริกซ์ $[w_1 \ w_2 \dots \ w_n]$ ที่มีขนาดเป็น 1 และแต่ละคอลัมน์เป็นอิสระต่อกันของเมตริกซ์ $A = [u_1 \ u_2 \dots \ u_n]$ โดยการใช้วิธีแกรม-ชมิทท์ (Gram-Schmidt) ที่กล่าวว่า เซตตั้งฉากปกติ $[u_1 \ u_2 \dots \ u_n]$ เป็นเซตอิสระเชิงเส้น แล้ว

$$v_i = u_i - \langle w_1, u_i \rangle w_1 - \langle w_2, u_i \rangle w_2 - \dots - \langle w_{i-1}, u_i \rangle w_{i-1},$$

$$w_i = \frac{v_i}{\|v_i\|} ; i = 1, 2, \dots \quad (2.7)$$

โดยให้ $Q = [w_1 \ w_2 \ \dots \ w_n]$

ขั้นที่ 2: คำนวณหา R จาก

$$A = QR$$

$$R = Q^t A$$

ตัวอย่าง 1 หาส่วนประกอบคิวอาร์ของ

$$A = \begin{bmatrix} 1 & -1 & -1 \\ 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

ขั้นที่ 1: จะได้เวกเตอร์ u_i คือ

$$u_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad u_2 = \begin{bmatrix} -1 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \quad u_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

อาศัยวิธีการของเกรมมิตต์ในสมการที่ (2.7) จะได้ว่า

$$v_1 = u_1 = [1 \ 1 \ 1 \ 0]^t \text{ และ } w_1 = \frac{u_1}{\|u_1\|} = \frac{[1 \ 1 \ 1 \ 0]^t}{\sqrt{1^2 + 1^2 + 1^2 + 0^2}} = \left[\frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}} \ 0 \right]^t$$

$$v_2 = u_2 - \langle w_1, u_2 \rangle w_1 = \begin{bmatrix} -1 \\ 0 \\ -1 \\ 0 \end{bmatrix} - \left\langle \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ -1 \\ 0 \end{bmatrix} \right\rangle \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \\ 0 \end{bmatrix} = \begin{bmatrix} -1/3 \\ 2/3 \\ -1/3 \\ 0 \end{bmatrix}$$

$$\text{และ } w_2 = \frac{v_2}{\|v_2\|} = \frac{\begin{bmatrix} -1/3 \\ 2/3 \\ -1/3 \\ 0 \end{bmatrix}}{\sqrt{\left(-\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2 + \left(-\frac{1}{3}\right)^2 + (0)^2}} = \left[-\frac{1}{\sqrt{6}} \ \frac{2}{\sqrt{6}} \ -\frac{1}{\sqrt{6}} \ 0 \right]^t$$

$$v_3 = u_3 - \langle w_1, u_3 \rangle w_1 - \langle w_2, u_3 \rangle w_2$$

$$= \begin{bmatrix} -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} - \left\langle \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \right\rangle \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix} - \left\langle \begin{bmatrix} -1/\sqrt{6} \\ 5/\sqrt{6} \\ -1/\sqrt{6} \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \right\rangle \begin{bmatrix} -1/\sqrt{6} \\ 5/\sqrt{6} \\ -1/\sqrt{6} \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{2} & \frac{3}{2} & \frac{1}{2} & 0 \end{bmatrix}^t$$

$$w_1 = \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \\ 0 \end{bmatrix}, w_2 = \begin{bmatrix} -1/\sqrt{15} \\ 2/\sqrt{15} \\ -1/\sqrt{15} \\ 3/\sqrt{15} \end{bmatrix}, w_3 = \begin{bmatrix} -4/\sqrt{35} \\ 3/\sqrt{35} \\ 1/\sqrt{35} \\ -3/\sqrt{35} \end{bmatrix}.$$

โดย $Q = [w_1 \ w_2 \dots w_n]$

ขั้นที่ 2:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{3} & -2/\sqrt{3} & -1/\sqrt{3} \\ 0 & 5/\sqrt{15} & -2/\sqrt{15} \\ 0 & 0 & 7/\sqrt{35} \end{bmatrix},$$

ดังนั้น

$$r_{11} = u_1 \cdot w_1 = \frac{3}{\sqrt{3}}, r_{12} = u_2 \cdot w_1 = \frac{-2}{\sqrt{3}}, r_{22} = u_2 \cdot w_2 = \frac{5}{\sqrt{15}},$$

$$r_{13} = u_3 \cdot w_1 = \frac{-1}{\sqrt{3}}, r_{23} = u_3 \cdot w_2 = \frac{-2}{\sqrt{15}}, r_{33} = u_3 \cdot w_3 = \frac{7}{\sqrt{35}}$$

ในการแก้สมการจะสามารถแก้ได้ดังต่อไปนี้

$$\text{จาก} \quad [A][X] = [B]$$

ให้ $[A] = [Q][R]$ ดังนั้นจะได้ว่า

$$[Q][R][X] = [B]$$

$$[Q]^t [Q][R][X] = [Q]^t [B]$$

$$[R][X] = [Z]$$

ในขั้นตอนสุดท้าย จะสามารถคำนวณหาค่า x ได้ด้วยวิธีการคำนวณค่าถ้อยหลัง (back substitution)

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ 0 & 0 & \cdots & \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix}$$

ตัวอย่างที่ 2

ให้

$$[A] = \begin{bmatrix} 1 & 2 & -1 & 3 \\ 2 & 1 & 1 & 2 \\ -2 & 3 & 4 & 1 \\ 4 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 1 & -1 & 2 & 0 \end{bmatrix}, [B] = \begin{bmatrix} 1 \\ 5 \\ -2 \\ 1 \\ 3 \\ 5 \end{bmatrix}$$

ด้วยวิธีการของแกรมชมิทท์ จะได้

$$[Q] = [w_1 \ w_2 \ w_3 \ w_4] = \begin{bmatrix} -0.1961 & -0.3851 & 0.5099 & 0.3409 \\ -0.3922 & -0.1311 & -0.1768 & 0.4244 \\ 0.3922 & -0.7210 & -0.4733 & -0.2177 \\ -0.7845 & -0.2622 & -0.1041 & -0.5076 \\ 0 & -0.4260 & 0.0492 & 0.4839 \\ -0.1961 & 0.2540 & -0.6867 & 0.4055 \end{bmatrix} \quad \text{และ}$$

$$[R] = \begin{bmatrix} -5.0990 & -0.9806 & 0.1961 & -0.9806 \\ 0 & -4.6945 & -2.8102 & -3.4164 \\ 0 & 0 & -4.0081 & 0.8504 \\ 0 & 0 & 0 & 3.1054 \end{bmatrix}.$$

แล้ว

$$[Z] = [Q]^t \cdot [B] = \begin{bmatrix} 4.7068 \\ -0.1311 \\ 2.8172 \\ 5.8699 \end{bmatrix}.$$

ขั้นตอนสุดท้ายแก้สมการหาค่า x_i

$$\begin{aligned}
 [R] \cdot [X] &= \begin{bmatrix} -5.0990 & -0.9806 & 0.1961 & -0.9806 \\ 0 & -4.6945 & -2.8102 & -3.4164 \\ 0 & 0 & -4.0081 & 0.8504 \\ 0 & 0 & 0 & 3.1054 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \\
 &= \begin{bmatrix} 4.7068 \\ -0.1311 \\ 2.8172 \\ 5.8699 \end{bmatrix} = [Z] = [Q]^T \cdot [B]
 \end{aligned}$$

จะได้

$$[X] = \begin{bmatrix} 0.999 \\ -2.0643 \\ 1.1039 \\ 1.8902 \end{bmatrix}$$

2.3 เทคนิคพื้นฐานในการแบ่งรูปภาพดิจิทัล

ในการบีบอัดรูปภาพดิจิทัลด้วยวิธีอัตโนมัติแบบถ่วงน้ำหนัก (Weighted Finite Automata, WFA) ซึ่งโดยปกติแล้วจะกำหนดขนาดของรูปภาพเป็น 2 ลักษณะคือ รูปภาพที่มีขนาดจำกัด (finite-resolution image) จะหมายถึงรูปภาพดิจิทัลที่เป็นรูปไล่ระดับสีเทา (grey-scale) ที่มีขนาดเป็น $2^m \times 2^m$ พิกเซล(pixels) (พิกเซล คือจุดที่ประกอบกันเป็นรูปภาพดิจิทัลขึ้นมา เช่น รูปภาพดิจิทัลขนาด $27 \times 27 = 128 \times 128$ พิกเซล หมายถึง รูปภาพดิจิทัลนี้จะมีจำนวนพิกเซลในแนวนอนเท่ากับ 128 จุด และ จำนวนพิกเซลในแนวตั้ง 128 จุด รวมกันทั้งสิ้น $128 \times 128 = 16,384$) โดยปกติแล้วจะให้ค่าอยู่ระหว่าง $7 \leq m \leq 11$ โดยความหมายของ รูปภาพที่มีขนาดไม่จำกัด (multiresolution image) จะหมายถึง รูปภาพดิจิทัลที่มีขนาดเท่ากับ $2^n \times 2^n$ พิกเซล สำหรับ $n = 0, 1, 2, \dots$ ดังนั้นจะสามารถกำหนดนิยามของรูปภาพที่มีขนาดไม่จำกัดเป็นฟังก์ชันได้ดังนี้

$$F: \Sigma^* \rightarrow \mathcal{R} \quad (2.8)$$

เมื่อ Σ เป็นตำแหน่งที่อยู่ซึ่งได้จากการแบ่งรูปภาพ และ

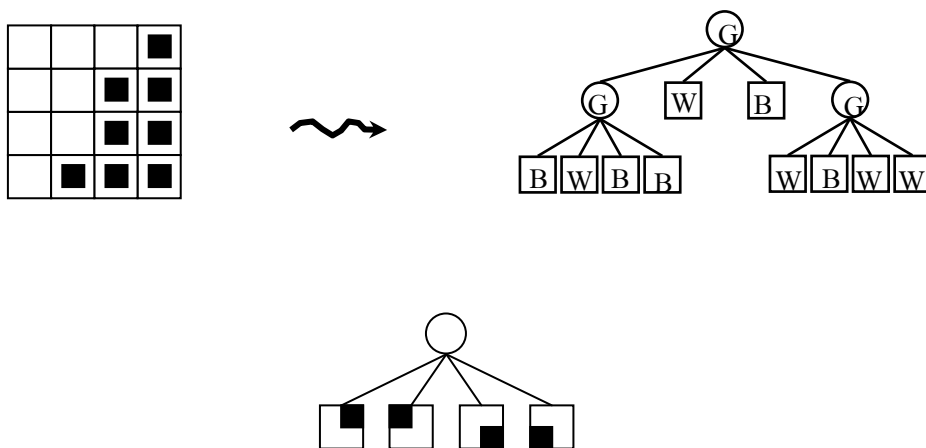
$F(x)$ เป็นค่าเฉลี่ยความเข้มแสงซึ่งจะเป็นจำนวนเต็มที่มีอยู่ระหว่าง 0 ถึง 255 โดยที่ค่าความเข้มแสง 0 จะมีสีดำ และค่าความเข้มแสง 255 จะมีสีขาว

การบีบอัดรูปภาพด้วยวิธีฮอโตมาตาแบบถ่วงน้ำหนักนี้จะเป็นการบีบอัดรูปภาพแบบไม่จำกัดขนาด โดยก่อนทำการบีบอัดจะต้องทำการแบ่งรูปภาพออกเป็นรูปย่อย โดยแต่ละรูปย่อยจะมีที่อยู่ หรือแอดเดรส (Address) เพื่อบ่งบอกตำแหน่งเพื่อที่จะนำแต่ละส่วนมาหาความสัมพันธ์กัน เพื่อที่จะนำรูปย่อยนั้นมาสร้างเป็นโหนด (node) และนำความสัมพันธ์มาสร้างเป็นขอบทางเดิน (edge) ,พาธ (path) หรือ ทรานสิชัน (transition) ของฮอโตมาตาแบบถ่วงน้ำหนัก ซึ่งเทคนิคในการแบ่งรูปภาพมีดังนี้

2.3.1 ควอดทรี (Quadrees)

ควอดทรีเป็นเทคนิคที่ใช้ในการแบ่งรูปภาพเพื่อทำการบีบอัดด้วยเทคนิคฮอโตมาตาแบบถ่วงน้ำหนักซึ่ง K. Culik II และ J. Kari [1] เป็นผู้คิดค้นการบีบอัดรูปดิจิทัลด้วยฮอโตมาตาแบบถ่วงน้ำหนัก ซึ่งการแบ่งแบบควอดทรีจะทำให้ได้โครงสร้างข้อมูลแบบเป็นลำดับขั้น โดยแต่ละขั้นจะทำแบบการแบ่งรูปภาพออกเป็น 4 ส่วนไปเรื่อยๆ จนกระทั่งได้ส่วนที่เหมาะสม (ในกรณีที่ไม่ที่สุดคือแบ่งจนกระทั่งได้ขนาดของรูปย่อยเป็น 1×1 พิกเซล) ขอบเขตของการแบ่งแบบควอดทรี (region quadtree) ก็คือจะแบ่งรูปภาพต้นฉบับเป็นส่วนๆ ไปด้วยอัตราส่วนของขนาดที่เท่ากัน ด้วยเหตุนี้ควอดทรีจะเป็นสามารถสร้างเป็นต้นไม้ (tree) ได้ โดยที่แต่ละโหนดจะสามารถมีโหนดลูก (Child) ได้ 4 โหนด หรือมีใบ (leaves) ได้ 4 ใบ โดยที่แต่ละโหนดจะสามารถแทนค่าได้ด้วยรูปย่อยซึ่งจะแทนค่าเป็นค่าความเข้มแสง (grayness values) เฉลี่ยของแต่ละรูปย่อย

รูปที่ 2.1 จะแสดงให้เห็นถึงเทคนิคเบื้องต้นของการแบ่งรูปภาพแบบควอดทรี โดยการแบ่งจะเริ่มจากมุมบนขวาของรูปภาพ มุมบนซ้าย มุมล่างขวา และมุมล่างซ้าย ตามลำดับ โดยจะมีตัวอักษร B W G แทนด้วยความเข้มแสงที่เป็นสีดำ ขาว เทา ตามลำดับ โดยจะสังเกตได้ว่าโหนดภายใน (inner node) จะมีเฉพาะตัวอักษร G และ ในส่วนที่เป็นใบ (leaves) ของทรีจะมีแต่ตัวอักษร B และ W ที่เป็นเช่นนี้เพราะว่าเนื่องจากโหนดภายในนั้นยังสามารถแบ่งออกไปได้อีก



รูปที่ 2.6 แสดงเทคนิคการแบ่งรูปแบบควอดทรี

ควอดทรีเป็นวิธีที่เหมาะสมสำหรับการบีบอัดรูปภาพในลักษณะที่มีขนาดรูปเป็น $2^n \times 2^n$ เพราะไม่ว่ารูปภาพจะมีขนาดใหญ่จะสามารถแบ่งรูปไปได้เรื่อยๆ จนกระทั่งได้โหนดเดี่ยว (single node) ที่เหมาะสม แต่การแบ่งที่ทำซ้ำกันไปเรื่อยๆ จะหยุดในกรณีพิเศษคือ ถ้าทุกๆ จุดมีความเข้มแสงเป็นสีขาว หรือดำ ทั้งหมด (homogeneity property) โดยที่ปกติแล้วจะหยุดก็ต่อเมื่อรูปย่อยจะมีขนาดเป็น 1×1 พิกเซล ซึ่งจะเห็นได้ดั่งนั้นตัวอย่างรูปที่ 2.1 ซึ่งถ้าเปรียบแต่ละช่องสี่เหลี่ยมเล็กๆ เป็นพิกเซล จะสังเกตเห็นว่าโหนดที่ 2 และ 3 ที่เป็นรูปนั้นจะมีขนาดเป็น 2×2 พิกเซล แต่ก็หยุดทำการแบ่ง เพราะเนื่องจากรูปย่อยทั้งสองรูปเป็นสีขาวและสีดำทั้งหมด แต่โหนดที่ 1 และ 4 ยังคงเป็นสีเทา ดังนั้นจึงยังคงต้องแบ่งอีกจนกระทั่งเป็นพิกเซลเดียว โดยการแบ่งแต่ละครั้งแต่ละรูปย่อยจะมีตำแหน่งแอดเดรสดังนี้

1	3
0	2

รูปที่ 2.7 แสดงตำแหน่งแอดเดรสของแต่ละรูปย่อย

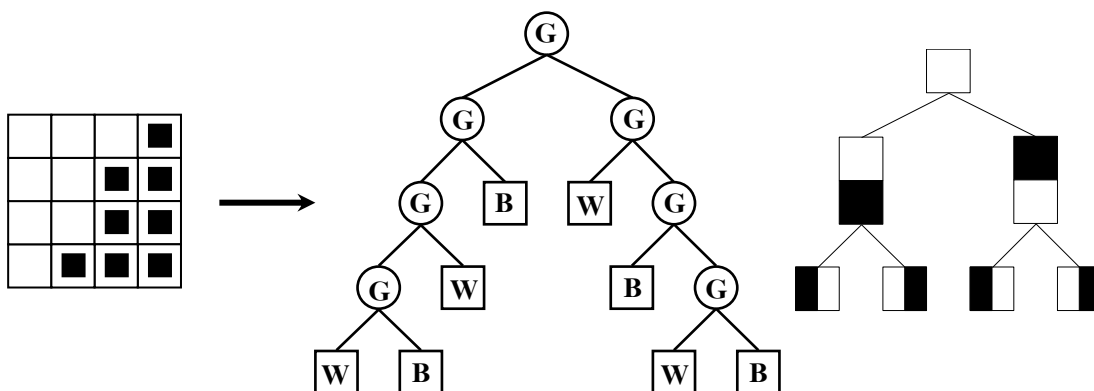
จากนิยามของรูปภาพที่มีขนาดไม่จำกัดจะทำให้สามารถนิยามเอเวอร์เรจพรีเซิร์ฟวิ่งก์ (Average preservint) หรือ เอพีฟังก์ชัน (ap-funtion) ดังนี้

$$F(x) = \frac{1}{4}[F(x_0) + F(x_1) + F(x_2) + F(x_3)] \quad (2.9)$$

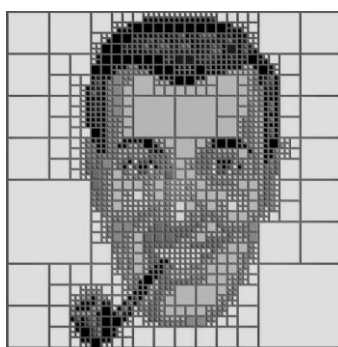
จากนิยามของเอพีฟังก์ชันจะทำให้ทราบได้ว่าเศษหนึ่งส่วนสี่ของผลรวมของค่าเฉลี่ยความเข้มแสงของทุกรูปย่อยจะมีค่าเท่ากับค่าเฉลี่ยความเข้มแสงของรูปภาพนั้นๆ ซึ่งจะพิจารณาได้จาก แต่ละรูปจะหาค่าเฉลี่ยความเข้มแสงได้จากผลรวมค่าความเข้มแสงทั้งหมดของรูปนั้นแล้วหารด้วยจำนวนพิกเซลทั้งหมด ด้วยวิธีการหาค่าเฉลี่ยความเข้มแสงถ้ากำหนดให้รูปภาพใดๆ ที่ประกอบด้วยรูปภาพย่อย 4 รูป โดยที่แต่ละรูปย่อยจะมีค่าเฉลี่ยความเข้มแสงเท่ากับ $F(x_0)$, $F(x_1)$, $F(x_2)$, และ $F(x_3)$ ดังนั้นรูปภาพใดๆ จะสามารถหาค่าเฉลี่ยความเข้มแสงได้จากผลรวมของค่าเฉลี่ยความเข้มแสงของทุกรูปย่อยซึ่งมีทั้งหมด 4 รูป และหารด้วยจำนวนรูปย่อย คือ 4

2.3.2 ไบนารีทรี (Binary Trees)

การแบ่งรูปแบบไบนารีทรีเป็นการประยุกต์การแบ่งมาจากควอดทรี โดยที่แต่ละโหนดของรูปภาพจะถูกแบ่งเป็นสองส่วนแทนที่จะเป็นสี่ส่วน การแบ่งแนวนอนและแนวตั้งเป็นการประยุกต์ในเรื่องของเลเวล (level) และดัชนีของแต่ละโหนดของต้นไม้ ซึ่งจะสามารถเห็นได้ว่าไบนารีทรีจะมีเลเวลของทรีเป็นจำนวนคู่ซึ่งจะคล้ายกับการแบ่งแบบควอดทรี ซึ่งในส่วนนี้จะใช้การแบ่งแบบไบนารีทรีมาประยุกต์ใช้กับการบีบอัดรูปภาพด้วยอโตมาตาแบบถ่วงน้ำหนัก สำหรับรูปที่ 2.2 จะแสดงถึงการแบ่งรูปแบบไบนารีทรี ในการตัดสินใจของการแบ่งว่าจะแบ่งแบบแนวนอนหรือแนวตั้งยังคงไม่มีกฎเกณฑ์ในการตัดสินใจที่แน่นอน ดังจะเห็นได้ในรูปที่ 2.3 ว่าแต่ละส่วนของรูปภาพจะมีการแบ่งแต่ละส่วนไม่เท่ากัน



รูปที่ 2.8 แสดงเทคนิคการแบ่งรูปแบบไบนารีทรี



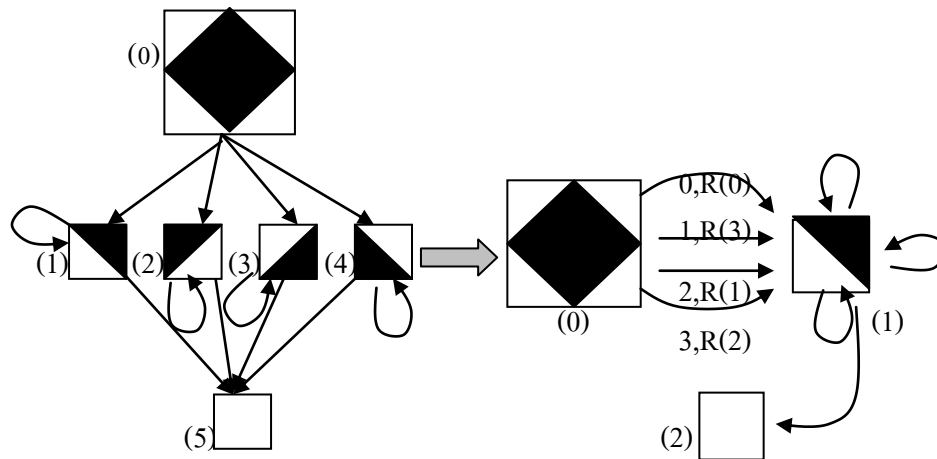
รูปที่ 2.9 แสดงผลลัพธ์การแบ่งรูปภาพด้วยวิธีไบนารีทรี

2.4 การประยุกต์การแบ่งรูป

เมื่อได้ทำการแบ่งรูปออกเป็นส่วนย่อยๆ แล้วและในการหาความสัมพันธ์ระหว่างรูปย่อยด้วยกัน โดยปกติแล้ว เพื่อที่จะให้สามารถหาความสัมพันธ์ระหว่างรูปย่อยได้ง่ายยิ่งขึ้น จึงอาจที่จะเพิ่มขึ้นตอนขึ้นมาอีกหนึ่งชั้น นั่นก็คือการหมุน และการย่อขยาย

2.4.1 การหมุน

เมื่อได้ทำการแบ่งรูปย่อยออกเป็นส่วนๆ แล้ว ก่อนที่จะนำมาสร้างเป็นไฟไนต์ออโตมาตาแบบถ่วงน้ำหนักแล้ว อาจจะทำมาหมุน ตามที่ Zhuhun Jiang , Bruce Litow และ Olivier de Vel [7] ได้ทำการวิจัยไว้แล้ว เพื่อที่จะทำให้สามารถใช้เนื้อที่หน่วยความจำได้น้อยลง จึงได้ทำการหมุนรูปย่อยก่อนเพื่อให้ที่จะได้หาความสัมพันธ์ได้ง่ายขึ้น ดังรูป



รูปที่ 2.10 แสดงการหมุนรูปเพื่อลดจำนวนโนนด

ในรูปที่ 2.5 จะสังเกตเห็นออโตมาตาที่อยู่ทางด้านซ้ายมือ จะเห็นว่าในโนนดที่ (0) เป็นรูปภาพรูปเดียวกับรูปที่จะทำการบีบอัด ซึ่งเมื่อทำการแบ่งรูปภาพออกเป็นสี่ส่วนตามแอดเดรส 0, 1, 2, 3 แล้วก็จะได้รูปย่อยซึ่งจะนำมาสร้างเป็นโนนดที่ (1), (2), (3) และ (4) ตามลำดับ ซึ่งถ้าสังเกตจะเห็นว่าโนนดที่ (1), (2), (3) และ (4) เป็นรูปที่มีลักษณะเหมือนกันเพียงแต่มีการหมุนรูปที่ไม่เท่ากัน กล่าวคือรูปของโนนดที่ (2) เกิดจากการหมุนรูปของโนนดที่ (1) ในทิศตามเข็มนาฬิกา ครั้งละ 90 องศา 3 ครั้ง รูปของโนนดที่ (3) ก็เกิดจากรูปของโนนดที่ (1) หมุนไปในทิศทางเดียวกัน 1 ครั้ง และรูปของโนนดที่ (4) เกิดจากการหมุนรูปของโนนดที่ (1) ไปในทิศทางเดียวกัน 2 ครั้ง ซึ่งจะได้ส่วนขยายของทางเดินของโนนดที่ (1), (2), (3) และ (4) เป็น $R(0)$, $R(3)$, $R(1)$ และ $R(2)$ เมื่อเทียบกับรูปของโนนดที่ (1) ตามลำดับ ในการหมุนรูปเพื่อลดจำนวนโนนด ทำให้สามารถใช้เนื้อจัดเก็บได้น้อยลง แต่จะต้องเสียเวลาในการคำนวณมากขึ้นทั้งขั้นตอนในการเข้ารหัสและถอดรหัส เนื่องจากในขั้นตอนการหมุนแต่ละครั้ง เนื่องจากต้องหมุนรูปแล้วจึงหาความสัมพันธ์ จนครบ 4 รอบ โดยการหมุนแต่ละครั้ง นิยมที่จะหมุนครั้งละ 90 องศา

2.4.2 การย่อหรือการขยาย

การย่อหรือการขยายก็เป็นพื้นฐานอย่างหนึ่งของกราฟฟิกเช่นกัน แต่โดยพื้นฐานของการเข้ารหัสแบบไฟไนต์ออโตมาตาแล้วจะรวมการย่อหรือการขยายเข้าไปด้วยแล้ว ดังนั้น ในพื้นฐานของการย่อหรือขยายจึงไม่จำเป็นที่จะต้องทำการตรวจสอบ

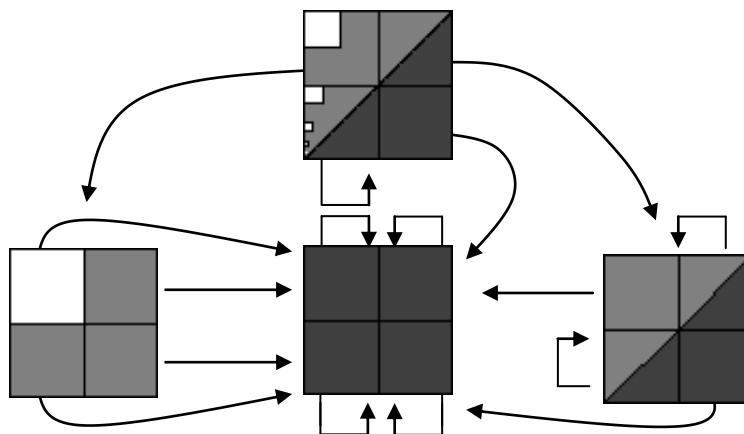
2.5 ความรู้เบื้องต้นในการบีบอัดภาพด้วยวิธีออโตมาตาแบบถ่วงน้ำหนัก

ในแต่ละจุดของรูปภาพดิจิทัลที่มีขนาด $2^n \times 2^n$, $n = 0, 1, 2, \dots$ ที่จะทำให้การแบ่งเพื่อทำการบีบอัดรูปภาพ จะมีการกำหนดตำแหน่งแอดเดรสที่มีความยาว n ซึ่งจะประกอบด้วยตัวอักษรที่จะบ่งบอกถึงตำแหน่งแอดเดรสของพิกเซลนั้นๆ โดยที่ตัวอักษรแต่ละตัวจะแสดงถึงควอดแรนต์ของรูปย่อยซึ่งเกิดจากการแบ่งรูปภาพดิจิทัลดังรูป

1	31	33
	30	32
0	2	

รูปที่ 2.11 แสดงตำแหน่งแอดเดรสของการแบ่งรูปแบบควอดทรี

โดยเมื่อทำการแบ่งเสร็จสิ้นแล้วก็ทำการสร้างออโตมาตาแบบถ่วงน้ำหนักเพื่อใช้ในการเข้ารหัสและถอดรหัสรูปภาพดิจิทัล เมื่อทำการแบ่งรูปภาพเสร็จเรียบร้อยแล้วก็นำมาหาความสัมพันธ์กันเพื่อนำมาสร้างเป็นออโตมาตาแบบถ่วงน้ำหนัก ดังรูปที่ 2.7 จะแสดงให้เห็นถึงเทคนิคพื้นฐานในการบีบอัดรูปภาพแบบออโตมาตาแบบถ่วงน้ำหนัก โดยแบ่งแบบควอดทรี



รูปที่ 2.12 แสดงถึงเทคนิคการสร้างกราฟสำหรับการบีบอัดรูปภาพ

จากรูปที่ 2.7 รูปที่อยู่บนสุดจะเป็นรูปที่จะทำการเข้ารหัส จะเป็นถูกกำหนดให้เป็นโหนดที่หนึ่งจากนั้นจึงทำการแบ่งออกเป็นสี่ส่วน จะสังเกตได้ว่าในแอดเดรสที่ 0 ของโหนดที่หนึ่งจะมีลักษณะเหมือนกับโหนดที่หนึ่งเองดังนั้นจึงสร้าง ทางเดิน (Path) วนเข้าหาโหนดที่หนึ่งเอง จากนั้นจึงพิจารณาในส่วนของแอดเดรสที่ 1 ของโหนดที่หนึ่ง ซึ่งจะสังเกตได้ว่าไม่เหมือนกับโหนดที่หนึ่ง ดังนั้น จึงนำรูปย่อยที่อยู่ในส่วนของแอดเดรสที่ 1 ของโหนดที่หนึ่งมาสร้างเป็นโหนดที่สองและสร้างทางเดินจากโหนดที่หนึ่งมายังโหนดที่สอง จากนั้นจึงพิจารณารูปย่อยในส่วนของแอดเดรสที่ 2 และ 3 ของโหนดที่หนึ่ง ซึ่งจะสังเกตได้ว่ามีลักษณะไม่เหมือนกับโหนดที่หนึ่งและโหนดที่สอง ดังนั้นจึงสร้างเป็นโหนดใหม่ได้แก่โหนดที่สามและโหนดที่สี่ตามลำดับ จากนั้นจึงสร้างทางเดินมายังโหนดที่สอง เมื่อพิจารณาทุกส่วนของรูปย่อยของโหนดที่หนึ่งเสร็จเรียบร้อยแล้วจึงพิจารณาโหนดที่สองต่อ

ในโหนดที่สอง จะสังเกตเห็นได้ว่ารูปย่อยที่มีแอดเดรส 0, 2 และ 3 จะมีความเข้มแสงเป็นครึ่งหนึ่งของโหนดที่สาม ดังนั้นจึงสามารถสร้างทางเดินของรูปย่อยที่มีแอดเดรสจากโหนดที่สองไปยังโหนดที่สามได้ ส่วนรูปย่อยที่มีแอดเดรสเป็น 1 จะมีลักษณะเหมือนกับโหนดที่สามดังนั้นจึงสามารถสร้างทางเดินไปยังโหนดที่สามได้เลย เมื่อพิจารณาโหนดที่สองเสร็จครบทุกรูปย่อยแล้วจึงพิจารณาโหนดที่สามต่อ

ในโหนดที่สาม จะสังเกตได้ว่าแต่ละรูปย่อยจะมีลักษณะเหมือนกับโหนดที่สามเองดังนั้นแต่ละรูปย่อยจึงสามารถสร้างทางเดินจากโหนดที่สามวนเข้าหาโหนดที่สามเองได้หลังจากนั้นจึงพิจารณาโหนดสุดท้ายคือโหนดที่สี่ ซึ่งในรูปย่อยที่มีแอดเดรสเป็น 0 กับ 4 จะมีลักษณะเหมือนกับตัวเอง ดังนั้นจึงวนเข้าหาตัวเอง และในรูปย่อยที่มีแอดเดรสเป็น 2 จะมีความเข้มแสงเป็นครึ่งหนึ่งของโหนดสามดังนั้นจึงสร้างทางเดินไปยังโหนดที่สาม ส่วนในรูปย่อยที่มีแอดเดรสเป็น 2 จะมีลักษณะเหมือนกับโหนดที่สาม ดังนั้นจึงสร้างทางเดินจากรูปย่อยที่ 2 ของโหนดที่สามไปยังโหนดที่