รายงานการวิจัย

การเข้ารหัสและถอดรหัสภาษาไทยด้วยออโตมาตาแบบถ่วงน้ำหนัก

Encoding and Decoding Thai Language with Weighted Finite Automata

ผศ. ดร. กรกช ประชุมรักษ์

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบรายได้ ประจำปีงบประมาณ 2550

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลงได้เนื่องจากได้รับทุนสนับสนุนจากเงินรายได้ประจำปี ของคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายละเอียดเกี่ยวกับโครงการ

ชื่อโครงการ การเข้ารหัสและถอดรหัสภาษาไทยด้วยออโตมาตาแบบถ่วงน้ำหนัก

Encoding and Decoding Thai Language with Weighted Finite Automata

ได้รับทุนสนับสนุนงานวิจัยจากคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ประจำปีงบประมาณ 2550 จำนวนเงิน 50,000 บาท
ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ ตุลาคม 2549 ถึง กันยายน 2550

ผู้ดำเนินการวิจัย ผศ. ดร. กรกช ประชุมรักษ์ สังกัด สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง โทร 02-3264341 ต่อ 286

บทคัดย่อ

ผลงานวิจัยชิ้นนี้นำเสนอวิธีการเข้ารหัสและถอดรหัสภาษาไทยด้วยออโตมาตาแบบถ่วงน้ำหนักซึ่งเป็นวิธีการใหม่ในการเข้ารหัส โดยใช้วิธีการหาค่าความสัมพันธ์ของตัวอักษรไทยที่อยู่ในประโยค ซึ่งเมื่อหาค่าความสัมพันธ์ได้แล้ว จะนำค่าความสัมพันธ์ต่างๆ มาสร้างเป็นออโตมาตาแบบถ่วงน้ำหนัก และนำค่าโหนดในออโตมาตามาแทนคำที่อยู่ในประโยค

This research proposes a new method to encode and decode Thai language with Weighted Finite Automata. To encode Thai language, this method is done by looking for the relationship of every character. Then, the relationships of the characters are used for the creation of the Weighted Finite Automata. Each node of the automata is the representation of the words in the sentences.

สารบัญ

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

งานวิจัยนี้เป็นการศึกษาการเข้ารหัสและถอดรหัสข้อความภาษาไทยด้วยออโตมาตาแบบถ่วงน้ำหนัก ซึ่งเป็นวิธีที่จะใช้หาค่าความสัมพันธ์ของตัวอักษรไทยที่อยู่ในประโยค เมื่อหาค่าความสัมพันธ์ได้แล้ว จะนำค่าความสัมพันธ์ต่างๆ มาสร้างเป็นออโตมาตาแบบถ่วงน้ำหนัก และนำค่าโหนดในออโตมาตานี้มาแทนคำที่อยู่ในประโยค

1.2 วัตถุประสงค์ของโครงการ

วิธีการนี้สามารถนำไปใช้ในการบีบอัดข้อความภาษาไทยให้มีขนาดเล็กลง โดยมีข้อดีคือ ข้อความที่มีขนาดเล็กลง ต้องการพื้นที่ในการเก็บน้อยลง จึงไม่สิ้นเปลืองพื้นที่ในหน่วยความจำ ไม่จำเป็นต้องเสียค่าใช้จ่ายเพิ่มมากขึ้นในการซื้ออุปกรณ์ฮาร์ดดิสก์มาเก็บข้อมูล และยังช่วยประหยัดเวลาในการถ่ายโอนข้อมูลจากต้นทางไปยังปลายทางอีกด้วย

1.3 ขอบเขตของการวิจัย

- ศึกษาและค้นคว้าการเข้ารหัสและถอดรหัสข้อความภาษาไทยด้วยออโตมาตาแบบถ่วงน้ำหนัก
- พัฒนาโปรแกรมการเข้ารหัสและถอดรหัสออโตมาตาแบบถ่วงน้ำหนักให้นำมาใช้กับภาษาไทยได้
- ทดลองและเปรียบเทียบผลการวิจัย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

โปรแกรมเข้ารหัสและถอดรหัสออโตมาตาแบบถ่วงน้ำหนักที่พัฒนาขึ้น สามารถนำไปใช้ในการบีบอัดข้อมูลภาษาไทย ทำให้ประหยัดพื้นที่ในการเก็บข้อมูลภาษาไทย ซึ่งจะช่วยในการลดค่าใช้จ่ายในการซื้ออุปกรณ์ฮาร์ดดิสก์เพิ่ม

1.5 ขั้นตอนการวิจัย

- ศึกษาเทคนิคการบีบอัดข้อความด้วยออโตมาตา
- ออกแบบโครงสร้างและขั้นตอนการบีบอัดข้อความด้วยวิธีการบีบอัดที่ศึกษามา
- พัฒนาโปรแกรมที่ใช้บีบอัดข้อความภาษาไทย
- ทดสอบบีบอัดข้อความภาษาไทยด้วยโปรแกรมต้นฉบับ
- ปรับปรุงแก้ไขและจัดทำเอกสารประกอบ

บทที่ 2

ความรู้เบื้องต้นเกี่ยวกับออโตมาตาแบบถ่วงน้ำหนัก

ออโตมาตาแบบถ่วงน้ำหนัก (Weighted Finite Automata) เป็นกราฟระบุทิศทาง (Directed Graph) ชนิดหนึ่ง โดยที่แต่ละพาท (Path) ได้มีการระบุค่าสำหรับการถ่วงน้ำหนักเอาไว้ด้วย การบีบอัดข้อมูลด้วยออโตมาตาแบบถ่วงน้ำหนักเช่น การบีบอัดรูปภาพ จะอาศัยเทคนิคอยู่ 2 เรื่อง ได้แก่ ออโตมาตา และสมการเชิงเส้น

2.1 ออโตมาตา (Automata)

ออโตมาตาเป็นโมเดลทางคณิตศาสตร์ของเครื่องจักรที่มีสถานะจำกัด (Finite state machine) เครื่องจักรที่มีสถานะจำกัดคือเครื่องจักรที่เมื่อรับข้อมูลเข้ามาแล้วจะมีการกระโดดหรือย้ายไปมาระหว่างสถานะต่างๆ ที่ได้กำหนดไว้ในฟังก์ชันการเปลี่ยนแปลงหรือฟังก์ชันทางผ่าน (transition function) ซึ่งเป็นหัวใจสำคัญสำหรับออโตมาตา เพราะเก็บข้อมูลนำเข้าและสถานะถัดไปของแต่ละสถานะเอาไว้ ข้อมูลที่ป้อนเข้าไปในเครื่องจักรจะถูกอ่านทีละตัวอักษรจนกระทั่งอ่านครบทุกตัว ยกตัวอย่างเช่น ถ้าหากข้อมูลที่ถูกป้อนเข้าไปเป็นข้อความ ข้อความนี้จะถูกแบ่งเป็นตัวอักษรแล้วจึงจะนำเข้าไปในออโตมาตาทีละตัวอักษรซึ่งจะอ่านเฉพาะข้อมูลที่ได้รับการยอมรับว่าเป็นชนิดข้อมูลที่ออโตมาตายอมรับ โดยจะอ่านจากทางด้านซ้ายไปยังทางด้านขวาจนกระทั่งครบทุกตัวอักษรที่อยู่ในข้อความ เมื่อข้อมูลถูกอ่านจนจบออโตมาตาก็จะหยุดทำงาน ในสถานะสุดท้ายของออโตมาตานั้น ถ้าเป็นสถานะที่ยอมรับได้ ก็จะกล่าวว่าออโตมาตายอมรับข้อมูลนี้ แต่ถ้าหากอยู่ในสถานะที่ยอมรับไม่ได้ ก็จะถือว่าไม่ยอมรับข้อมูลนี้
คุณลักษณะโดยทั่วไปของออโตมาตาจะมีดังนี้

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

เนื่องจากออโตมาตามีสถานะที่จำกัดดังนั้นจึงอาจเรียกได้ว่า ออโตมาตาจำกัด (Finite Automata, FA)

2.1.1 สายอักขระ (String) และภาษา (Language)

ชุดตัวอักษร คือเซตจำกัดของสัญลักษณ์ซึ่งจะไม่เป็นเซตว่าง โดยจะใช้สัญลักษณ์ Σ แทนชุดตัวอักษรใดๆ เช่น Σ={0, 1} หรือ Σ={a, b, c} เป็นต้น สายอักขระ จากชุดอักษร Σ คือลำดับจำกัดของสัญลักษณ์จาก Σ ซึ่งเขียนติดกันโดยไม่มีช่องว่างและไม่มีเครื่องหมายใดๆ มาคั่น เช่น Σ={0, 1} จะสามารถสร้างสายอักขระ 1100 ได้เป็นต้น

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 $|\varepsilon|=0$

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

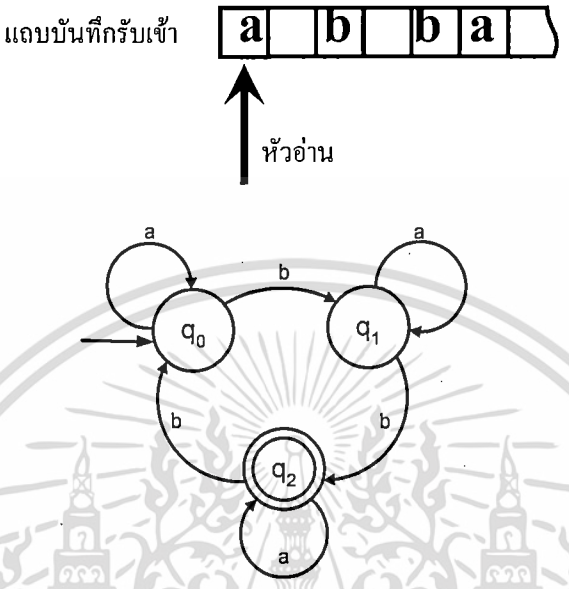
สำหรับชุดตัวอักษร Σ ใดๆ สัญลักษณ์ Σ^* จะหมายถึงเซตของสายอักขระทั้งหมดจาก Σ ซึ่งจะรวมถึงสายอักขระว่างด้วย และสัญลักษณ์ Σ^+ จะหมายถึง $\Sigma^*-\{\varepsilon\}$

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

2.1.1.1 ออโตมาตาเชิงกำหนด (Deterministic Finite Automata, DFA)

ลักษณะของออโตมาตาเชิงกำหนด จะรับข้อมูลเข้าเป็นสายอักขระผ่านทาง แถบบันทึกรับเข้า (input tape) ที่จะอ่านข้อมูลเข้าไปทีละตัวจากทางด้ายซ้ายมือโดยหัวอ่านดังรูปที่ 2.1 และเขียนออโตมาตาเชิงกำหนด อยู่ในรูปของกราฟบ่งบอกทิศทาง (directed graph) โดยจะเริ่มจาก สถานะเริ่มต้น(initial state, q_0) โดยในการย้ายสถานะแต่ละครั้ง ออโตมาตาเชิงกำหนด จะอยู่ในสถานะใดสถานหนึ่งเสมอ เช่นในรูปที่ 2.2 แสดงให้เห็นถึงออโตมาตาที่กำลังอยู่ในสถานะที่ q_0 กำลังจะอ่านข้อมูลนำเข้า a เมื่ออ่านเสร็จแล้วหัวอ่านจะย้ายไปทางขวา 1 ช่อง และออโตมาตาเชิงกำหนด จะย้ายสถานะไป q_1 ซึ่งในออโตมาตาเชิงกำหนด ใดๆ อาจจะมีการย้ายสถานะหรือไม่ย้ายก็ได้ขึ้นอยู่กับตัวอักษรที่เป็นข้อมูลนำเข้าและลักษณะของฟังก์ชันทางผ่าน เมื่อหัวอ่านทำการอ่านข้อมูลนำเข้าทีละ 1 ตัวจนหมดสายอักขระ แล้วออโตมาตาเชิงกำหนด จะบ่งบอกว่าสายอักขระนี้ยอมรับโดยออโตมาตาเชิงกำหนด หรือไม่ โดยดูได้จากสถานะสุดท้าย

ของออโตมาตาเชิงกำหนด ว่าเป็นสถานะที่ยอมรับ (accepting state) ได้หรือไม่ ในรูปที่ 2.1 สถานะที่ยอมรับจะถูกแทนด้วยวงกลม 2 วงซ้อนกัน ก็คือสถานะ q_2

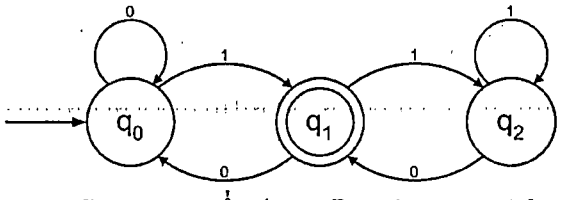


รูปที่ 2.1 แสดงตัวอย่างออโตมาตาเชิงกำหนด

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 จากรูปออโตมาตา M



รูปที่ 1.2 ออโตมาตา M ซึ่งเขียนอยู่ในรูปของกราฟบ่งบอกทิศทาง

จากรูปที่ 2.2 จะแสดงถึงออโตมาตา M ซึ่งเป็นอยู่ในรูปของกราฟบ่งบอกทิศทาง ซึ่งจะกำหนดให้แต่ละจุดยอด (vertex) หรือ โหนดแทนแต่ละสถานะ โดยที่สถานะเริ่มต้นจะมีลูกศรชี้อยู่และสถานะสุดท้ายทุกจุดใน 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 $\delta(q_0,1)=q_1$ แต่เนื่องจากในการรับข้อมูลเข้านั้นไม่ได้รับเพียงแค่สัญลักษณ์นำเข้าเพียงตัวเดียวแต่เป็นสายอักขระ ดังนั้นจึงต้องเขียนให้อยู่ในรูป $\delta^*(q_i,w)$ โดยที่ $q_i \in Q$, $w \in \Sigma^*$ ซึ่งสัญลักษณ์ดาว (star) จะบ่งบอกว่าข้อมูลนำเข้าที่กำลังอ่านอยู่ ยังมีลักษณะเป็นสายอักขระ ซึ่งเมื่อทำการย้ายสถานะเรียบร้อยแล้วก็ให้อ่านข้อมูลนำเข้าตัวต่อไป โดยฟังชันก์การเปลี่ยนแปลงถัดไปจะมีสถานะเป็น $\delta^*(q_j,x)$ เมื่อ $\delta(q_i,a)=qj$ โดยที่ $q_j,q_i \in Q$, $w,a \in \Sigma^*$, $x \in \Sigma$ 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 ดังนั้นสำหรับข้อความนี้จึงไม่ยอมรับโดยออโตมาตา M ข้อความ 01111001

$$\begin{aligned}\delta^*(q_0,01111001) &= \delta^*(q_0,1111001)\\ &= \delta^*(q_1,111001)\\ &= \delta^*(q_0,11001)\\ &= \delta^*(q_1,1001)\\ &= \delta^*(q_2,001)\\ &= \delta^*(q_1,01)\\ &= \delta(q_0,1)\\ &= q_1\end{aligned}$$

จะเห็นข้อความ 01111001 สถานะสุดท้ายจะอยู่ที่โหนด q_1 ซึ่งเป็นสถานะสุดท้าย ดังนั้นออโตมาตา M จึงยอมรับข้อความนี้

2.1.1.2 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 ส่วน เช่นกันคือ $N=(Q,\Sigma,\delta,q_0,F)$ เมื่อ

Q เป็นเซตจำกัดของสถานะภายใน

Σ เป็นเซตจำกัดของสัญลักษณ์นำเข้าที่เรียกว่าตัวอักษรนำเข้า

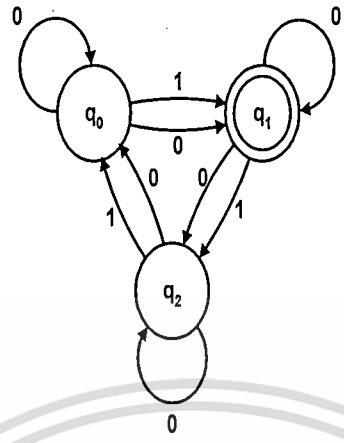
δ เป็นฟังก์ชันจาก $Q\times\Sigma$ ไปยัง P(Q) เขียนแทนด้วย $\delta : Q\times\Sigma\rightarrow P(Q)$

q_0 เป็นสถานะเริ่มต้น

F เป็นเซตของสถานะสุดท้ายที่เป็นสถานะที่มีการยอมรับสายอักขระ

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

ตัวอย่างที่ 2.2 จากรูปที่ 2.3 แสดงถึงเอ็นเอฟเอ N



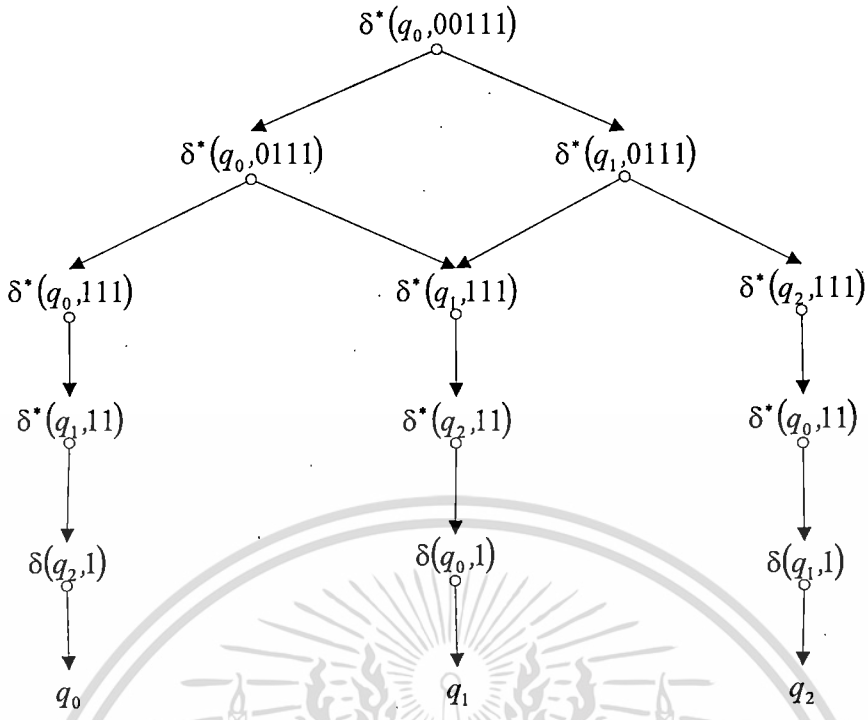
รูปที่ 2.2 แสดงถึงเอ็นเอฟเอ N

การแจกแจงสมาชิกของออโตมาตาจำกัดเชิงไม่กำหนดจะเขียนได้ดังนี้ $N=\{\{q_0,q_1,q_2,q_3\}, \{0,1\},\delta, \{q_0\},\{q_1,q_2\}\}$ 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 2.3



รูปที่ 2.3 แสดงเส้นทางผ่านของเอ็นเอฟเอ N

จะเห็นได้ว่าสายอักขระ 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

2.1.2 ระบบสมการเชิงเส้น (System of Linear Equation)

ในส่วนนี้จะกล่าวถึงส่วนที่เกี่ยวข้องกับการหาค่าถ่วงน้ำหนักของออโตมาตาซึ่งจะใช้ความรู้ทางคณิตศาสตร์ช่วยในการแก้สมการ เมื่อกล่าวถึงสมการโดยทั่วๆ ไปที่กำลังของตัวแปร (variable) หรือตัวไม่ทราบมีกำลังเป็น 1 แล้ว สมการจะอยู่ในรูปของสมการเชิงเส้น ดังนี้

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \qquad (2.2.1)$$

เมื่อ $a_i, b \in \Re$ 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 2.2.1 เป็นจริง หรือจะกล่าวว่า ถ้ากำหนดให้

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

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 2.2.1 ซึ่งผลเฉลยของสมการอาจจะมีได้มากกว่า 1 เซต

ในกรณีที่สมการที่ 2.2.1 มี m สมการแต่มีตัวแปรชุดเดียวกัน ดังนี้

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n &= b_3 \\
&\dots\dots\dots\dots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m
\end{aligned} \qquad (2.2.2)$$

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 เอกพันธุ์ (hemogeneous) ซึ่งจะเขียนได้อยู่ในรูปของ สมการที่ 2.2.3

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= 0 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= 0 \\
&\dots\dots\dots\dots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= 0
\end{aligned} \qquad (2.2.3)$$

ซึ่งระบบสมการเอกพันธุ์นี้จะมีเซตของผลเฉลยเสมออย่างน้อย 1 เซต คือ {0, 0, … , 0} ซึ่งจะมีชื่อเฉพาะสำหรับเรียกผลเฉลยนี้ว่า ผลเฉลยที่เป็นศูนย์ หรือผลเฉลยที่มีความสำคัญน้อย (trivial solution)

ในกรณีที่นำค่าคงที่ c_i คูณกับสมการที่ i ของระบบสมการ 2.2.2 ซึ่งจะได้ดังนี้

$$\begin{aligned} c_1a_{11}x_1 + c_1a_{12}x_2 + \dots + c_1a_{1n}x_n &= c_1b_1 \\ c_2a_{21}x_1 + c_2a_{22}x_2 + \dots + c_2a_{2n}x_n &= c_2b_2 \\ &\dots\dots\dots\dots \\ c_ma_{m1}x_1 + c_ma_{m2}x_2 + \dots + c_ma_{mn}x_n &= c_mb_m \end{aligned} \qquad (2.2.4)$$

แล้วนำแต่ละสมการมาบวกกันแล้วจัดรูปจะได้ดังสมการที่ 2.2.5 ดังนี้

$$\begin{aligned} &(c_1a_{11} + c_2a_{21} + c_3a_{31} + \dots + c_ma_{m1})x_1 + \\ &(c_1a_{12} + c_2a_{22} + c_3a_{32} + \dots + c_ma_{m2})x_2 + \\ &\dots\dots\dots\dots + (c_1a_{1n} + c_2a_{2n} + c_3a_{3n} + \dots + c_ma_{mn})x_n \\ &= c_1b_1 + c_2b_2 + c_3b_3 + \dots + c_mb_m \end{aligned} \qquad (2.2.5)$$

จะเรียกสมการในลักษณะนี้ว่า การรวมเชิงเส้น (linear combination) ซึ่งจะสามารถกล่าวได้ ผลเฉลยของระบบสมการเชิงการใดๆ จะเป็นผลเฉลยของการรวมเชิงเส้นของระบบสมการนั้นด้วย หรือจะกล่าวว่าถ้าเซตของ $\{ k_1, k_2, \dots , k_n \}$ เป็นผลเฉลยของสมการ 2.2.2 แล้วจะเป็นผลเฉลยของสมการ 2.2.4 ด้วย ซึ่งจะแสดงได้ดังนี้

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

$$\begin{aligned} a_{11}k_1 + a_{12}k_2 + \dots + a_{1n}k_n &= b_1 \\ a_{21}k_1 + a_{22}k_2 + \dots + a_{2n}k_n &= b_2 \\ a_{31}k_1 + a_{32}k_2 + \dots + a_{3n}k_n &= b_3 \\ &\dots\dots\dots\dots \\ a_{m1}k_1 + a_{m2}k_2 + \dots + a_{mn}k_n &= b_m \end{aligned} \qquad (2.2.6)$$

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

จะต้องแสดงว่า

$$(c_1a_{11} + \dots + c_ma_{m1})k_1 + \dots + (c_1a_{1n} + \dots + c_ma_{mn})k_n = c_1b_1 + \dots + c_mb_m$$

จัดรูปแบบพจน์ทางด้านซ้ายมือของเครื่องเท่ากับใหม่จะได้ว่า

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

จากสมการ 2.2.5 จะได้ว่าเท่ากับ $c_1b_1+c_2b_2+c_3b_3+\ldots+c_mb_m$

นอกจากนี้ยังสามารถที่จะจัดรูปแบบสมการ 2.2.5 ให้อยู่ในรูปเมตริกซ์ดังนี้

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad (2.2.7)$$

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

2.1.3 เทคนิคพื้นฐานที่เกี่ยวข้องกับการบีบอัดรูปภาพ

เทคนิคพื้นฐานของการบีบอัดรูปภาพเป็นส่วนที่จะทำให้การบีบอัดรูปภาพได้ประสิทธิภาพยิ่งขึ้น ได้แก่ การแบ่งรูป การหมุน และ การย่อขยาย

ในการบีบอัดรูปภาพดิจิตอลด้วยวิธีออโตมาตาแบบถ่วงน้ำหนัก โดยปกติแล้วจะกำหนดขนาดของรูปภาพเป็น $2^m \times 2^m$ พิกเซล^2 (pixels^2), m = 0, 1, 2, ... ที่เป็นรูปภาพดิจิตอลที่ไล่ระดับสีเทา (grey-scale)) โดยปกติแล้วจะให้มีค่าอยู่ระหว่าง $7 \leq m \leq 11$ ดังนั้นจะสามารถกำหนดนิยามของรูปภาพที่มีขนาดไม่จำกัดเป็นฟังก์ชันได้ดังนี้

$$F:\Sigma^* \rightarrow \Re \qquad (2.2.8)$$

เมื่อ Σ เป็นตำแหน่งที่อยู่ซึ่งได้จากการแบ่งรูปภาพ และ

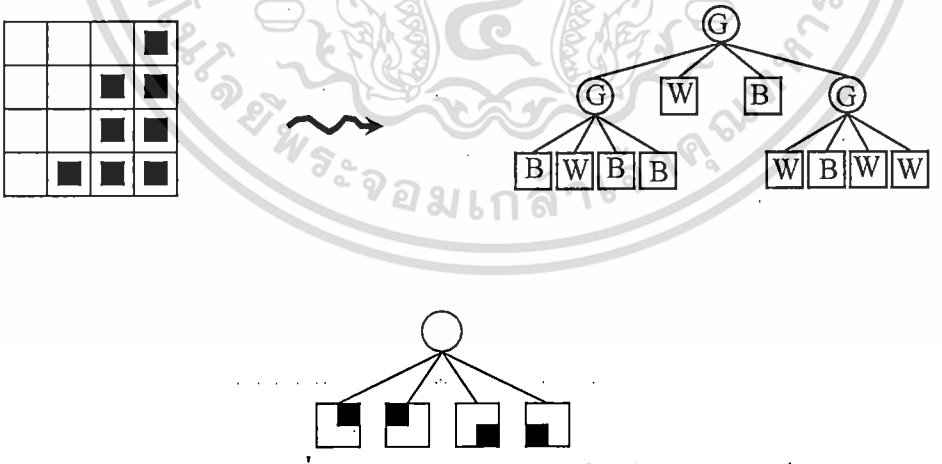
F(x) เป็นค่าเฉลี่ยความเข้มแสงซึ่งจะเป็นจำนวนเต็มที่มีอยู่ระหว่าง 0 ถึง 255 โดยที่ค่าความเข้มแสง 0 จะมีสีดำ และค่าความเข้มแสง 255 จะมีสีขาว

ก่อนทำการบีบอัดจะต้องทำการแบ่งรูปภาพออกเป็นรูปย่อย โดยแต่ละรูปย่อยจะมีที่อยู่ หรือแอดเดรส (Address) เพื่อบ่งบอกตำแหน่งเพื่อที่จะนำแต่ละส่วนมาหาความสัมพันธ์กัน ทำให้มีเทคนิคการแบ่งและการหาความสัมพันธ์กันด้วยการหมุน และการย่อขยาย

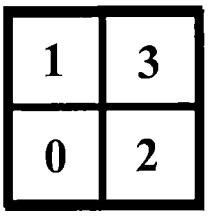
2.1.3.1 ควอดทรี (Quadtrees)

การแบ่งแบบควอดทรีจะทำให้ได้โครงสร้างข้อมูลแบบเป็นลำดับขั้น โดยแต่ละขั้นจะทำแบบการแบ่งรูปภาพออกเป็น 4 ส่วนไปเรื่อยๆ จนกระทั่งได้ส่วนที่เหมาะสม (ในกรณีที่แย่ที่สุดคือแบ่งจนกระทั่งได้ขนาดของรูปย่อยเป็น 1×1 พิกเซล) ขอบเขตของการแบ่งแบบควอดทรี (region quadtree) คือจะแบ่งรูปภาพต้นฉบับเป็นสี่ส่วนซ้ำไปเรื่อยๆ ด้วยอัตราส่วนของขนาดที่เท่ากัน ด้วยเหตุนี้ควอดทรีจะเป็นสามารถสร้างเป็นต้นไม้ (tree) ได้ โดยที่แต่ละโหนดจะสามารถมีโหนดลูก (Child)ได้ 4 โหนด หรือมีใบ (leaves) ได้ 4 ใบ โดยที่แต่ละโหนดจะสามารถแทนค่าได้ด้วยรูปย่อยซึ่งจะแทนค่าเป็นค่าความเข้มแสง (grayness values) เฉลี่ยของแต่ละรูปย่อย

รูปที่ 2.5 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 2.5 แสดงเทคนิคการแบ่งรูปแบบควอดทรี

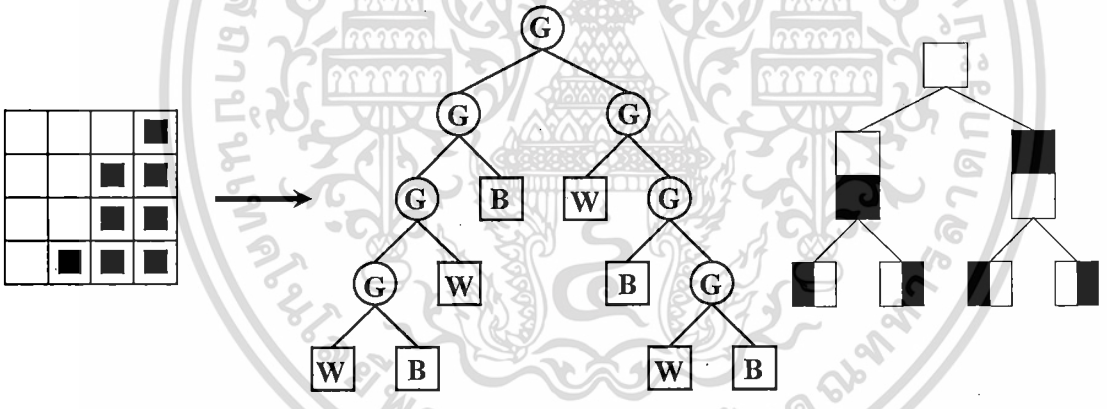


รูปที่ 2.4 แสดงตำแหน่งแอดเดรสของแต่ละรูปย่อย

ควอดทรีเป็นการแบ่งที่ทำซ้ำกันไปเรื่อยๆ โดยที่ปกติแล้วจะหยุดก็ต่อเมื่อรูปย่อยจะมีขนาดเป็น 1×1 พิกเซล โดยการแบ่งแต่ละครั้งแต่ละรูปย่อยจะมีตำแหน่งแอดเดรสดังนี้

2.1.3.2 ไบนารีทรี (Binary Trees)

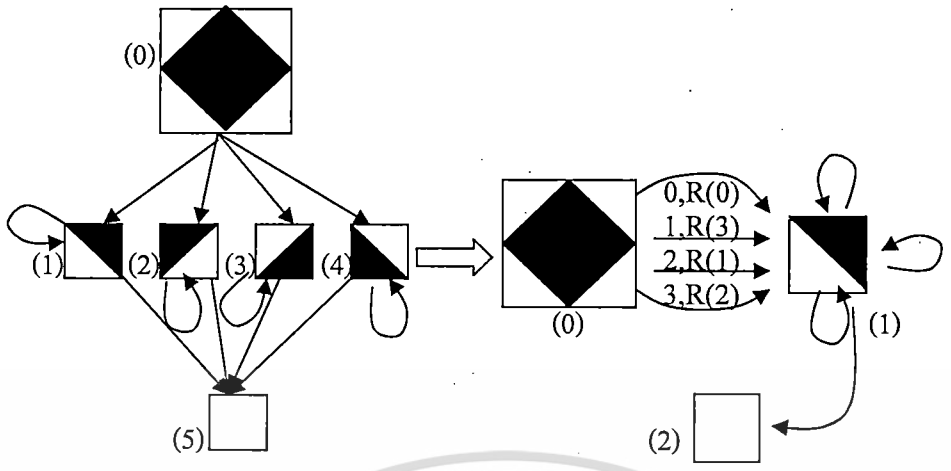
การแบ่งรูปแบบไบนารีทรีเป็นการแบ่งสองส่วนแทนที่จะเป็นสี่ส่วน โดยจะเป็นการแบ่งแนวนอนและแนวตั้ง ดังรูปที่ 2.7 จะแสดงถึงการแบ่งรูปแบบไบนารีทรี



รูปที่ 2.5 แสดงเทคนิคการแบ่งรูปแบบไบนารีทรี

2.1.3.3 การหมุน

เมื่อได้ทำการแบ่งรูปย่อยออกเป็นส่วนๆ แล้ว ก่อนที่จะนำมาสร้างเป็นไฟไนต์ออโตมาตาแบบถ่วงน้ำหนักแล้ว อาจจะนำมาหมุน เพื่อที่จะทำให้แต่ละรูปมีความสัมพันธ์กันมากขึ้น ทำให้สามารถใช้เนื้อที่หน่วยความจำได้น้อยลง ดังรูป



รูปที่ 2.6 แสดงการหมุนรูปเพื่อลดจำนวนโหนด

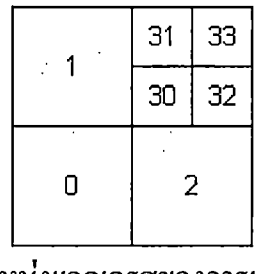
ในรูปที่ 2.8 จากออโตมาตาที่อยู่ทางด้านซ้ายมือ จะพบว่าในโหนดที่ (0) เป็นรูปภาพรูปเดียวกับรูปที่จะทำการบีบอัด ซึ่งเมื่อทำการแบ่งรูปภาพออกเป็นสี่ส่วนตามแอดเดรส 0, 1, 2, 3 แล้วก็จะได้รูปย่อยซึ่งจะนำมาสร้างเป็นโหนดที่ (1), (2) ,(3) และ (4) ตามลำดับ ซึ่งถ้าสังเกตจะเห็นว่าโหนดที่ (1), (2), (3) และ (4) เป็นรูปที่มีลักษณะเหมือนกันเพียงแต่มีการหมุนรูปที่ไม่เท่ากัน

2.1.3.4 การย่อหรือการขยาย

การย่อหรือการขยายก็เป็นพื้นฐานอย่างหนึ่งของกราฟฟิกส์เช่นกัน แต่โดยพื้นฐานของการเข้ารหัสแบบไฟไนต์ออโตมาตาแล้วจะรวมการย่อหรือการขยายเข้าไปอยู่แล้ว ดังนั้น ในพื้นฐานของการย่อยหรือขยายจึงไม่จำเป็นที่จะต้องทำการตรวจสอบ

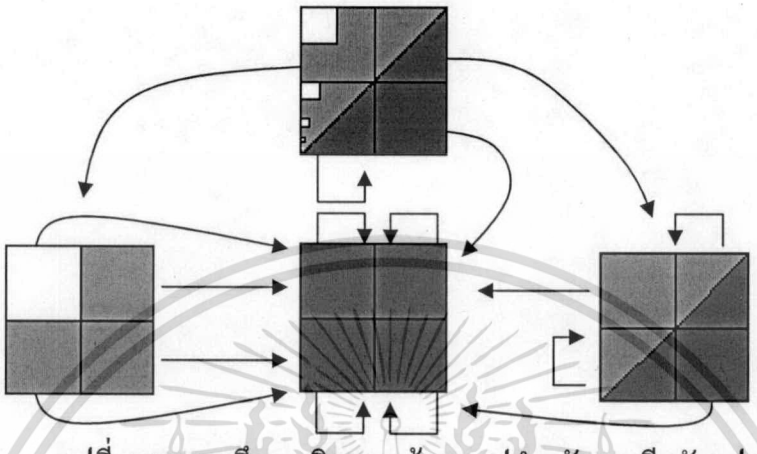
2.1.4 ความรู้เบื้องต้นในการบีบอัดภาพด้วยวิธีออโตมาตาแบบถ่วงน้ำหนัก

ในแต่ละจุดของรูปภาพดิจิตอลที่มีขนาด $2^n \times 2^n$, $n = 0, 1, 2, \ldots$ ที่จะทำการแบ่งเพื่อทำการบีบอัดรูปภาพ จะมีการกำหนดตำแหน่งแอดเดรสที่มีความยาว n ซึ่งจะประกอบด้วยตัวอักษรที่จะบ่งบอกถึงตำแหน่งแอดเดรสของพิกเซลนั้นๆ โดยที่ตัวอักษรแต่ละตัวจะแสดงถึงควอดเดรนท์ของรูปย่อยซึ่งเกิดจากการแบ่งรูปภาพดิจิตอลดังรูป



รูปที่ 2.7 แสดงตำแหน่งแอดเดรสของการแบ่งรูปแบบควอดทรี

โดยเมื่อทำการแบ่งเสร็จสิ้นแล้วก็ทำการสร้างออโตมาตาแบบถ่วงน้ำหนักเพื่อใช้ในเข้ารหัสและถอดรหัสรูปภาพดิจิตอล เมื่อทำการแบ่งรูปภาพเสร็จเรียบร้อยแล้ว ก็นำมาหาความสัมพันธ์กันเพื่อนำมาสร้างเป็นออโตมาตาแบบถ่วงน้ำหนัก ดังรูปที่ 2.10 จะแสดงให้เห็นถึงเทคนิคพื้นฐานในการบีบอัดรูปภาพแบบออโตมาตาแบบถ่วงน้ำหนัก โดยแบ่งแบบควอดทรี



รูปที่ 2.8 แสดงถึงเทคนิคการสร้างกราฟสำหรับการบีบอัดรูปภาพ

จากรูปที่ 2.10 รูปที่อยู่บนสุดจะเป็นรูปที่จะทำการเข้ารหัส จะเป็นถูกกำหนดให้เป็นโหนดที่หนึ่ง จากนั้นจึงทำการแบ่งออกเป็นสี่ส่วน จะสังเกตได้ว่าในแอดเดรสที่ 0 ของโหนดที่หนึ่งจะมีลักษณะเหมือนกับโหนดที่หนึ่งเองดังนั้นจึงสร้าง ทางเดิน (Path) วนเข้าหาโหนดที่หนึ่งเอง จากนั้นจึงพิจารณาในส่วนของแอดเดรสที่ 1 ของโหนดที่หนึ่ง ซึ่งจะสังเกตได้ว่าไม่เหมือนกับโหนดที่หนึ่งดังนั้น จึงนำรูปย่อยที่อยู่ในส่วนของแอดเดรสที่ 1 ของโหนดที่หนึ่งมาสร้างเป็นโหนดที่สองและสร้างทางเดินจากโหนดที่หนึ่งมายังโหนดที่สอง จากนั้นจึงพิจารณารูปย่อยในส่วนของแอดเดรสที่ 2 และ 3 ของโหนดที่หนึ่ง ซึ่งจะสังเกตได้ว่ามีลักษณะไม่เหมือนกับโหนดที่หนึ่งและโหนดที่สอง ดังนั้นจึงสร้างเป็นโหนดใหม่ได้แก่ โหนดที่สามและโหนดที่สี่ตามลำดับ จากนั้นจึงสร้างทางเดินมายังโหนดที่สอง เมื่อพิจารณาทุกส่วนของรูปย่อยของโหนดที่หนึ่งเสร็จเรียบร้อยแล้วจึงพิจารณาโหนดที่สองต่อ

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

ในโหนดที่สาม จะสังเกตได้ว่าแต่ละรูปย่อยจะมีลักษณะเหมือนกับโหนดที่สามเองดังนั้นแต่ละรูปย่อยจึงสามารถสร้างทางเดินจากโหนดที่สามวนเข้าหาโหนดที่สามเองได้หลังจากนั้นจึงพิจารณาโหนดสุดท้ายคือโหนดที่สี่ ซึ่งในรูปย่อยที่มีแอดเดรสเป็น 0 กับ 4 จะมีลักษณะเหมือนกับตัวเอง ดังนั้นจึงวนเข้าหาตัวเอง และในรูปย่อยที่มีแอดเดรสเป็น 2 จะมีความเข้มแสงเป็นครึ่งหนึ่งของโหนดสามดังนั้นจึงสร้าง

ทางเดินไปยังโหนดที่สาม ส่วนในรูปย่อยที่มีแอดเดรสเป็น 2 จะมีลักษณะเหมือกับโหนดที่สาม ดังนั้นจึงสร้างทางเดินจากรูปย่อยที่ 2 ของโหนดที่สามไปยังโหนดที่ 2

บทที่ 3

ตัวอักษรภาษาไทยและออโตมาตาแบบถ่วงน้ำหนัก

3.1 อักษรภาษาไทย

อักษรไทยเป็นอักษรที่ใช้เขียนภาษาไทย จะมีรายละเอียดดังนี้

ตัวอักษรหรือ พยัญชนะ ได้แก่ ก ถึง ฮ ทั้งหมดมี 44 ตัว

ตัวสระ ได้แก่ –ะ –า ... ทั้งหมดมี 28 ตัว โดยจะมีรูปแบบทั้งหมด 21 รูปแบบ

วรรณยุกต์ ได้แก่ ่ ้ ๊ ๋ ทั้งหมดมี 4 รูป

ตัวเลข ได้แก่ ๐ ๑ ๒ ... ๙ ทั้งหมด 10 ตัว

สัญลักษณ์อื่นๆ ได้แก่ ๆ “” ฯ ฯ , . () _ ?

พยัญชนะไทยจะเรียงตัวไปตามแนวนอน จากซ้ายไปขวา ส่วนสระจะอยู่หน้า บน ล่าง และหลังพยัญชนะประกอบคำแล้วแต่ชนิดของสระแต่ละตัว

อักษรไทยไม่มีการแยกตัวอักษรไทยตัวใหญ่ หรือไทยตัวเล็ก และไม่มีการเว้นวรรคระหว่างคำ ทำให้เมื่อจบประโยคจะลงท้ายด้วยเว้นวรรค หรือเป็นเครื่องหมายวรรคตอนซึ่งอยู่ในกลุ่มของสัญลักษณ์อื่นๆ จึงทำให้ในการแบ่งคำด้วยระบบคอมพิวเตอร์จะค่อนข้างลำบากแต่ในงานวิจัยชิ้นนี้ไม่ได้เกี่ยวข้องกับการแบ่งคำ จึงจะถือว่าการเว้นวรรค และสัญลักษณ์อื่นๆ เป็นตัวอักษรตัวหนึ่งด้วย

ในการนำตัวอักษรไทยมาใช้ในระบบคอมพิวเตอร์ จะมีการแปลงตัวอักษรไทยไปเป็นรหัสคอมพิวเตอร์ที่เรียกว่ารหัสแอสกี เพื่อความสะดวกในการอ้างอิงว่าตัวอักษรแต่ละตัวควรใช้เลขฐาน 2 เป็นอะไร ซึ่งรหัสแอสกีนี้จะสามารถนำไปใช้นำในการคำนวณค่าความสัมพันธ์ต่างๆ ได้เช่นเดียวกัน

				0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0				0	@	P		p				ฐ	ก	ะ	เ	๐
0	0	0	1			!	1	A	Q	a	q			ก	ฑ	ม	ั	แ	๑
0	0	1	0			"	2	B	R	b	r			ข	ฒ	ย	า	โ	๒
0	0	1	1			#	3	C	S	c	s			ข	ณ	ร	ำ	ใ	๓
0	1	0	0			$	4	D	T	d	t			ก	ด	ฤ	ิ	ไ	๔
0	1	0	1			%	5	E	U	e	u			ง	ต	ล	ี	ๅ	๕
0	1	1	0			&	6	F	V	f	v			ง	ถ	ฦ	ึ	ๆ	๖
0	1	1	1			'	7	G	W	g	w			ง	ท	ว	ื	็	๗
1	0	0	0			(	8	H	X	h	x			จ	ธ	ศ	ุ	่	๘
1	0	0	1			)	9	I	Y	i	y			ฉ	น	ษ	ู	้	๙
1	0	1	0			*	:	J	Z	j	z			ช	บ	ส	ฺ	๊	๑
1	0	1	1			+	;	K	[	k	?			ช	ป	ห	฻	+	๓
1	1	0	0			,	<	L	\	l	:			ฌ	ผ	ฬ	฼	๋	
1	1	0	1			-	=	M	]	m				ญ	ฝ	อ	฽	์	
1	1	1	0			.	>	N	^	n				ฎ	พ	ฮ	฾	๎	
1	1	1	1			/	?	O	-	o				ฏ	ฟ	ฯ	฿	๏	

รูปที่ 3.1 แสดงรหัสแอสกี

รหัสแอสกีที่อ่านได้จากในตารางจะเป็นเลขฐาน 2 ซึ่งประกอบด้วยเลข 0 และ 1 ทั้งหมด 8 ตัว โดยแถวบนจะเป็นตัวเลข 4 ตัวแรก และด้านซ้ายมือจะเป็นตัวเลข 4 ตัวสุดท้าย ซึ่งทำให้สามารถนำมาแปลงเป็นตัวเลขฐาน 10 หรือฐาน 16 ได้ตามความเหมาะสมหรือความต้องการ เช่น

ตัวอักษร ก จะมีรหัสแอสกีที่เป็นเลขฐาน 2 ได้แก่ 1010 0001 เมื่อแปลงเป็นเลขฐาน 10 ก็ได้เป็น 161

ตัวอักษร ฮ จะมีรหัสแอสกีที่เป็นเลขฐาน 2 ได้แก่ 1100 1110 เมื่อแปลงเป็นเลขฐาน 10 ก็ได้เป็น 206

เครื่องหมาย (จะมีรหัสแอสกีที่เป็นเลขฐาน 2 ได้แก่ 0010 1000 เมื่อแปลงเป็นเลขฐาน 10 ก็ได้เป็น 40

เครื่องหมาย) จะมีรหัสแอสกีที่เป็นเลขฐาน 2 ได้แก่ 0010 1001 เมื่อแปลงเป็นเลขฐาน 10 ก็ได้เป็น 41

เนื่องจากสัญลักษณ์บางตัวนั้นในชุดรหัสแอสกีของภาษาไทยนั้นไม่มี จึงต้องนำมาจากชุดรหัสแอสกีของภาษาอังกฤษมาใช้บ้าง เพื่อที่จะให้ชุดของอักษรไทยมีครบ

3.2 ออโตมาตาแบบถ่วงน้ำหนักสำหรับการเข้ารหัสภาษาไทย

จากหัวข้อที่ผ่านมาทำให้ทราบว่าในการเข้ารหัสด้วยออโตมาตาแบบถ่วงน้ำหนักนั้น โดยพื้นฐานจะอาศัยการแบ่งรูปแล้วนำมาหาค่าความสัมพันธ์กัน ดังนั้นสำหรับข้อความภาษาไทยใดๆที่จะทำการเข้ารหัสด้วยวิธีออโตมาตาแบบถ่วงน้ำหนัก ซึ่งกำหนดให้มีขนาดความยาวเท่ากับ 2^n ตัวอักษร เมื่อ n = 0, 1, 2 ... , n โดยรวมทั้งสระและ ตัวอักษรพิเศษ หรือสัญลักษณ์ต่างๆ ด้วยแล้ว จะได้ว่าแต่ละตัวอักษรในประโยคจะมีค่าที่บ่งบอกตำแหน่งหรือที่อยู่ มีความยาวเท่ากับ n โดยที่ n จะประกอบไปด้วยตัวเลขในเซต Σ = {0, 1} โดยตำแหน่งของแต่ละตัวอักษรจะสามารถหาได้คล้ายกับการแบ่งรูปแบบไบนารี่ทรีคือ การแบ่งด้านซ้ายและด้านขวาที่ละครึ่งไปเรื่อยๆ โดยกำหนดให้ จุดเริ่มต้น หรือประโยคนั้นมีที่อยู่หรือตำแหน่งเป็น ε และเมื่อแบ่งครั้งต่อไป ก็จะเป็น 0 และ 1 ซึ่งมีขนาดความยาวของที่อยู่เป็น 1 โดยตัวเลข 0 จะเป็นตำแหน่งที่อยู่ของประโยคย่อยที่ถูกตัดทางด้านซ้าย และตัวเลข 1 จะเป็นตำแหน่งเป็นประโยคย่อยที่ถูกตัดทางด้านขวา เมื่อทำการตัดคำในประโยคย่อยแต่ละประโยคอีกครั้งหนึ่ง ก็จะในตำแหน่งที่เพิ่มขึ้นต่อท้ายตำแหน่งที่อยู่เดิม และมีขนาดของความยาวที่อยู่เพิ่มขึ้นอีก 1 ด้วย ดังตัวอย่างที่ 1

ตัวอย่างที่ 1 กำหนดให้ $C_0C_1C_2C_3C_4C_5C_6C_7$ เป็นข้อความภาษาไทยใดๆ ที่มีความยาวเท่ากับ 8 ตัวอักษร (รวมทั้งสระ วรรณยุกต์ และตัวอักษรพิเศษต่างๆ)

กำหนดให้ ε เป็นที่อยู่ของภาษาไทยทั้งประโยค ดังนั้น จะได้ว่า

เมื่อเริ่มต้น	ที่อยู่	ε	เท่ากับ	$C_0C_1C_2C_3C_4C_5C_6C_7$
เมื่อทำการแบ่งครั้งที่ 1 จะได้ว่า	ที่อยู่	0	เท่ากับ	$C_0C_1C_2C_3$
	ที่อยู่	1	เท่ากับ	$C_4C_5C_6C_7$
เมื่อทำการแบ่งครั้งที่ 2 จะได้ว่า	ที่อยู่	00	เท่ากับ	C_0C_1
	ที่อยู่	01	เท่ากับ	C_6C_7
	ที่อยู่	10	เท่ากับ	C_4C_5
	ที่อยู่	11	เท่ากับ	C_6C_7
เมื่อทำการแบ่งครั้งที่ 2 จะได้ว่า	ที่อยู่	000	เท่ากับ	C_0
	ที่อยู่	001	เท่ากับ	C_1
	ที่อยู่	010	เท่ากับ	C_6
	ที่อยู่	011	เท่ากับ	C_7
	ที่อยู่	100	เท่ากับ	C_4
	ที่อยู่	101	เท่ากับ	C_5
	ที่อยู่	110	เท่ากับ	C_6
	ที่อยู่	111	เท่ากับ	C_7

ซึ่งก็จะได้ว่าความยาวของที่อยู่ของแต่ละตัวอักษรเท่ากับ 3 ซึ่ง ประโยคนี้มีความยาวเท่ากับ 2^3 ตัวอักษร

ตัวอย่างที่ 2 กำหนดประโยค “เราไปเที่ยว กทม.”

จากประโยค “เราไปเที่ยว กทม.” จะได้ว่ามีความยาวเท่ากับ $16 = 2^4$ และจะกำหนดตำแหน่งที่อยู่ได้ดังนี้

เมื่อเริ่มต้น	ที่อยู่	ε	เท่ากับ	เราไปเที่ยว กทม.
เมื่อทำการแบ่งครั้งที่ 1 จะได้ว่า	ที่อยู่	0	เท่ากัน	เราไปเที
	ที่อยู่	1	เท่ากับ	่ยว กทม.
เมื่อทำการแบ่งครั้งที่ 2 จะได้ว่า	ที่อยู่	00	เท่ากัน	เราไ
	ที่อยู่	01	เท่ากัน	ปเที
	ที่อยู่	10	เท่ากับ	่ยว_ (ขอให้สัญลักษณ์ _ แทนการเว้นวรรค)
	ที่อยู่	11	เท่ากับ	กทม.
เมื่อทำการแบ่งครั้งที่ 3 จะได้ว่า	ที่อยู่	000	เท่ากัน	เร
	ที่อยู่	001	เท่ากัน	าไ
	ที่อยู่	010	เท่ากัน	ปเ
	ที่อยู่	011	เท่ากัน	ที
	ที่อยู่	100	เท่ากับ	่ย
	ที่อยู่	101	เท่ากับ	ว_
	ที่อยู่	110	เท่ากับ	กท
	ที่อยู่	111	เท่ากับ	ม.
เมื่อทำการแบ่งครั้งที่ 4 จะได้ว่า	ที่อยู่	0000	เท่ากัน	เ
	ที่อยู่	0001	เท่ากัน	ร
	ที่อยู่	0010	เท่ากัน	า
	ที่อยู่	0011	เท่ากัน	ไ
	ที่อยู่	0100	เท่ากัน	ป
	ที่อยู่	0101	เท่ากัน	เ
	ที่อยู่	0110	เท่ากัน	ท
	ที่อยู่	0111	เท่ากัน	ี
	ที่อยู่	1000	เท่ากับ	่
	ที่อยู่	1001	เท่ากับ	ย
	ที่อยู่	1010	เท่ากับ	ว
	ที่อยู่	1011	เท่ากับ	_
	ที่อยู่	1100	เท่ากับ	ก

ที่อยู่ 1101 เท่ากับ ท

ที่อยู่ 1110 เท่ากับ ม

ที่อยู่ 1111 เท่ากับ .

ดังนั้นเราจะได้ว่าประโยค “เราไปเที่ยว กทม.” ซึ่งมีความยาวเท่ากับ $16 = 2^4$ แต่ละตัวอักษรจะมีความยาวของตำแหน่งที่อยู่เท่ากับ 4

สำหรับข้อความภาษาไทยใดๆ จะสามารถกำหนดฟังก์ชันจำนวนจริง $f: \Sigma^* \longrightarrow R$ ได้ว่า

$$f(w) = \frac{1}{|w|}\sum_{w\in\Sigma} f(wa)$$

โดยฟังก์ชันนี้จะได้ค่าออกมาเป็นค่าเฉลี่ยของข้อความ ซึ่งได้จากการนำรหัสแอสกีมาคำนวณ

ออโตมาตาแบบถ่วงน้ำหนักจะมีส่วนประกอบทั้งหมด 5 ส่วน ได้แก่

1. Q ซึ่งจะเป็นเซตของโหนดของออโตมาตา โดยโหนดของออโตมาตาแต่ละโหนดจะแทนด้วยข้อความหรือตัวอักษรของภาษาไทย
2. $\Sigma = \{0, 1\}$
3. $W_a : QxQ \longrightarrow R$ เป็นฟังก์ชันถ่วงน้ำหนัก สำหรับ W(p, a, q) ซึ่งเป็นค่าถ่วงน้ำหนักของทางเดินจากโหนด p ไปยังโหนด q ด้วยทางเดิน a สำหรับทุกๆ $p, q \in Q$ และ $a \in \Sigma$
4. $I : Q \longrightarrow R$ เป็นค่าเริ่มต้นของแต่ละโหนด
5. $F : Q \longrightarrow R$ เป็นค่าสิ้นสุดของแต่ละโหนด

ซึ่งอาจจะกล่าวได้ว่า ออโตมาตา A ใดๆ จะประกอบด้วย $\{Q, \Sigma, W, I, F\}$ หรือเขียนได้ว่า $A = \{Q, \Sigma, W, I, F\}$

3.3 การเข้ารหัสข้อความภาษาไทย

สำหรับตัวอักษรไทยใดๆ ที่อยู่ในประโยค ซึ่งจะสามารถหาได้จากฟังก์ชัน

$$S : a \longrightarrow R$$

โดยค่านำเข้า a คือตำแหน่งของตัวอักษรภาษาไทยที่อยู่ในประโยค

ในกระบวนการสร้างออโตมาตาแบบถ่วงน้ำหนักจะกำหนดตัวแปรดังนี้

$F(q)$	เป็นค่าสิ้นสุดของโหนด q
$I(q)$	เป็นค่าเริ่มต้นของโหนด q
N	เป็นโหนดสุดท้ายที่ถูกสร้าง
q	เป็นตำแหน่งของโหนดที่กำลังพิจารณาอยู่

$\gamma : Q \rightarrow \Sigma^*$ เป็นฟังก์ชันซึ่งมีค่านำเข้าเป็นโหนด แล้วจะได้ผลลัพธ์เป็นที่อยู่ของข้อความภาษาไทยที่โหนดนั้นเป็นตัวแทน

ค่านำเข้า : ข้อความภาษาไทย S

ค่าส่งออก : ออโตมาตาแบบถ่วงน้ำหนัก T ที่แทนข้อความภาษาไทย S

1. กำหนดให้ $N = 0$, $i = 0$, $F(q_0) = f(\varepsilon)$, $\gamma(q_0) = \varepsilon$

2. พิจารณาโหนด q_i สำหรับ $w = \gamma(q_i)$ และทุกๆ $a \in \Sigma = \{0, 1\}$

   ก. ถ้าสามารถหาค่าสัมประสิทธิ์ $c_0, c_1, c_2, \ldots, c_N$ ของสมการ

$$f_{wa} = c_0 M_0 + c_1 M_1 + \ldots + c_N M_N$$

เมื่อ $M_j = f_{\gamma(q_i)}$ สำหรับ $j = 0, 1, \ldots, N$ แล้วกำหนดให้ $W_a(q_i, q_j) = c_j$ สำหรับ $j = 0, 1, \ldots, N$

   ข. ในกรณีที่ไม่สามารถหาได้ กำหนดให้ $\gamma(q_{N+1}) = wa$, $F(q_{N+1}) = f_{avg}(wa)$ และ $W(q_i, a, q_N) + 1 = 1, N = N + 1$

3. กำหนดให้ $i = i + 1$ และ ถ้า $i \leq N$ ให้กลับไปทำข้อ 2.

4. กำหนดให้ $I(q_0) = 1$ และ $I(q_j) = 0$ สำหรับ $j = 1, 2, \ldots, N$ เมื่อ I เป็นค่าเริ่มต้นของแต่ละโหนด

ในกระบวนการขั้นตอนที่ 2.ก เป็นขั้นตอนสำหรับการสร้างสมการหาค่าความสัมพันธ์ของตัวอักษรต่างๆ ซึ่งจะอยู่ในรูปของสมการเชิงเส้น ซึ่งถ้ากำหนดให้ S_i เป็นข้อความภาษาไทยซึ่งมีตำแหน่งเป็น i จะได้ว่า

$$S_i = \frac{1}{2}(S_{i0} + S_{i1})$$

จากคุณสมบัติของตัวมันเองจะแยกได้อีกว่า

$$S_{i0} = \frac{1}{2}(S_{i00} + S_{i01})$$

$$S_{i1} = \frac{1}{2}(S_{i10} + S_{i11})$$

ซึ่งจะได้เป็นระบบสมการเชิงเส้น

3.4 การถอดรหัสข้อความภาษาไทย

สำหรับในหัวข้อนี้จะกล่าวถึงการถอดรหัสข้อความภาษาไทยจากออโตมาตาแบบถ่วงน้ำหนักให้กลับมาเป็นภาษาไทย

ค่านำเข้า : ออโตมาตาแบบถ่วงน้ำหนัก T

ค่าส่งออกก : ข้อความภาษาไทย S

1. กำหนด $\sigma_p(\varepsilon) = F(p)$ สำหรับทุกๆ $p \in Q$
2. ให้ทำขั้นตอนที่ 3 สำหรับทุกๆ $i = 1, 2, \ldots, n$
3. สำหรับทุกๆ $p = Q$, $w = \Sigma^{i-1}$ และ $a \in \Sigma$ ให้คำนวณค่าของ

$$\sigma_p(aw) = \sum_{q \in Q} w(p,a,q)\sigma_q(w)$$

4. สำหรับทุกๆ $w \varepsilon \Sigma^n$ ให้คำนวณค่าดังต่อไปนี้

$$S(w) = \sum_{q \in Q} I(q)\xi_q(w)$$

บทที่ 4

การเข้ารหัสและถอดรหัสภาษาไทย

ในส่วนนี้จะแสดงตัวอย่างในการเข้ารหัสและถอดรหัสภาษาไทย ซึ่งจะขอยกประโยคภาษาไทยขึ้นมาเพียง 1 ประโยคคือ ประโยคคำว่า "กาแฟสดมีประโยชน์" โดยก่อนที่จะทำการเข้ารหัสจะต้องทำการเปลี่ยนตัวอักษรต่างๆ ให้เป็นรหัสแอสกีตามรูปที่ 1 เสียก่อน ซึ่งจะได้รหัสแอสกีในรูปของเลขฐาน 10 ได้ดังนี้

161 210 225 191 202 181 193 213 187 195 208 226 194 171 185 236

โดยในขั้นแรกจะกำหนดให้ประโยคเริ่มต้นนี้เป็นโหนดที่ 0 ดังรูป 2 โดยที่มีค่าสิ้นสุดเท่ากับ 184.875 แล้วจึงทำการแบ่งข้อความออกเป็น 2 ส่วน ซึ่งจะสามารถสร้างสมการสำหรับข้อความที่ 1 เพื่อการคำนวณได้ดังนี้

$$172.875 = c_0\ 184.875$$

และสามารถที่จะแยกออกมาเป็นระบบสมการได้ดังนี้

$$161 = c_0 185.5$$
$$210 = c_0 208$$
$$225 = c_0 191.5$$
$$191 = c_0 203$$
$$202 = c_0 191$$
$$181 = c_0 217$$
$$193 = c_0 182.5$$
$$213 = c_0 210.5$$

ซึ่งไม่สามารถที่จะหาค่า c_0 ที่จะทำให้ทุกสมการเป็นจริงได้ดังนั้นจึงกำหนดให้ข้อความแรกเป็นโหนดที่ 1 ซึ่งจะมีค่าสิ้นสุดของโหนดเป็น 172.875 โดยที่มีทางเดินเป็น 0 และมีค่าถ่วงน้ำหนักเป็น 1 ต่อไปจึงพิจารณา ข้อความที่ 2 โดยจะมีสมการดั้งนี้

$$200.25 = c_0\ 184.875$$

เมื่อแตกสมการแล้วจะได้

$$187 = c_0 185.5$$
$$195 = c_0 208$$
$$208 = c_0 191.5$$
$$226 = c_0 203$$

$$194 = c_0 191$$

$$171 = c_0 217$$

$$185 = c_0 182.5$$

$$236 = c_0 210.5$$

ซึ่งก็ไม่สามารถที่จะหาค่าของ c_0 ที่จะทำให้ทุกสมการเป็นจริงได้ดังนั้นจึงกำหนดให้ข้อความที่ 2 นี้เป็นโหนดที่ 2 ซึ่งจะมีค่าสิ้นสุดของโหนดเป็น 200.25 โดยมีทางเดินเป็น 1 และมีค่าถ่วงน้ำหนักเป็น 1

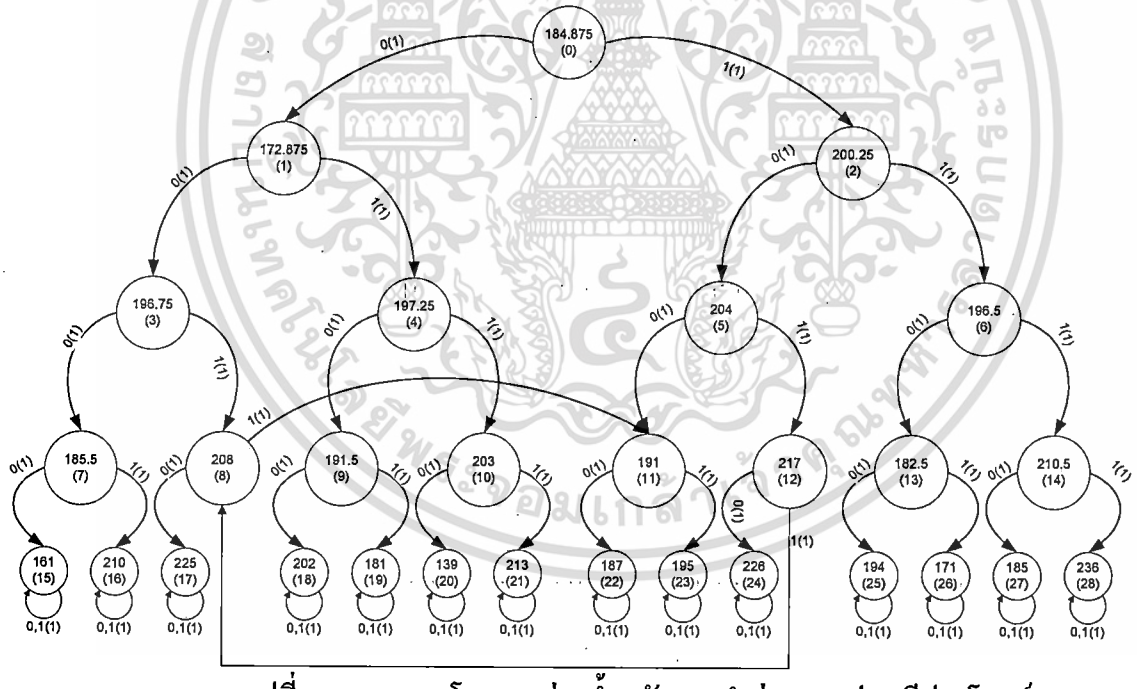
เมื่อกลับมาพิจารณาโหนดที่ 1 ก็จะสามารถสร้างสมการสำหรับข้อความที่ 1 และข้อความที่ 2 ได้ดังนี้

$$196.75 = c_0 184.875 + c_1 172.875 + c_2 200.25$$

$$197.25 = c_0 184.875 + c_1 172.875 + c_2 200.25$$

ตามลำดับ ก็จะไม่สามารที่จะหาค่าสัมประสิทธิ์ c_0, c_1 และ c_2 ได้ จึงต้องสร้างโหนดใหม่เป็นโหนดที่ 3 และ 4 โดยที่มีค่าทั้งเดินเป็น 1 และมีค่าถ่วงน้ำหนักเป็น 0 และมีค่าสิ้นสุดเป็น 196.75 และ 197.25 ตามลำดับ

เมื่อพิจารณาทุกๆ ข้อความเสร็จแล้วจะได้ออโตมาตาแบบถ่วงน้ำหนักสำหรับประโยค “กาแฟสดมีประโยชน์” ดังรูป



รูปที่ 4.1 แสดงออโตมาตาถ่วงน้ำหนักของคำว่า “กาแฟสดมีประโยชน์”

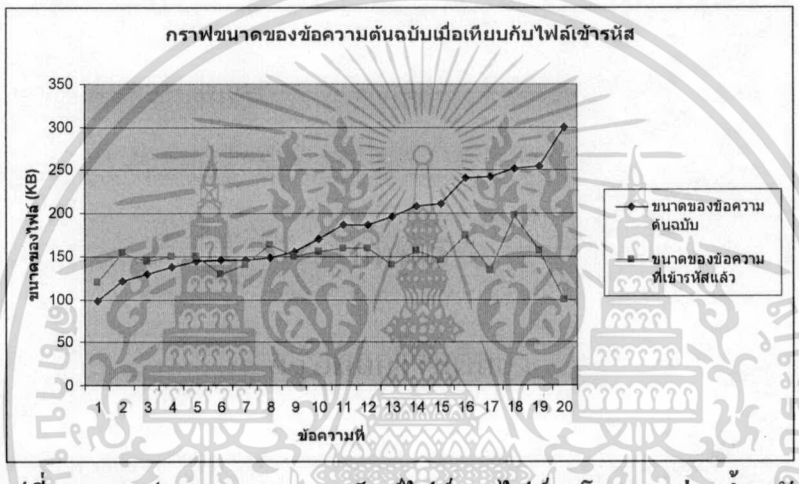
ในการถอดรหัสภาษาไทยออกจากออโตมาตาแบบถ่วงน้ำหนักนั้น จะหาได้จากผลคูณของค่าเริ่มต้นของโหนดเริ่มต้น ค่าถ่วงน้ำหนักของเส้นทางที่ผ่านและค่าสิ้นสุดของโหนดสุดท้าย โดยสิ่งที่ได้จากการเดินผ่านโหนดต่างๆ อีกอย่างหนึ่งก็คือ ตำแหน่งของตัวอักษรที่เดินผ่าน ยกตัวอย่างเช่น ในกรณีที่เดินทางผ่านโหนด 0, 1, 3, 7 และ 15 จะสามารถคำนวณหาค่ารหัสแอสกีได้ดังนี้ I(0) × W(0,0,1) × W(1,0,3) × W(3,0,7) × W(7,0,15) × F(15) = 1 × 1 × 1 × 1 × 1 × 161 = 161 ซึ่งเมื่อทำเป็นเลขฐาน 2 แล้วจะได้

ค่าเท่ากับ 10100001 ซึ่งจะเท่ากับตัวอักษร ก ซึ่งเมื่อทำการถอดรหัสทุกๆ ทางเดินแล้วก็จะได้ประโยคที่ว่า “กาแฟสดมีประโยชน์” ออกมา

บทที่ 5

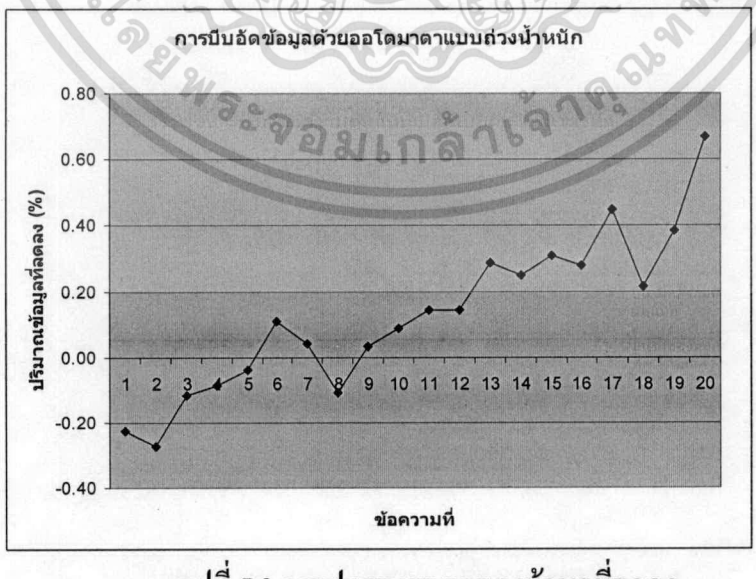
สรุปผลการทดลอง

ตัวอย่างที่ใช้ในการทดสอบการบีบอัดข้อมูลจะได้มาจากข่าวสารทั่วไปในตามเว็บไซต์ต่างๆ ซึ่งจะทำให้ได้ลักษณะของข้อความและคำที่ใช้กันโดยทั่วไปด้วยการสุ่ม โดยตัวอย่างจะถูกเก็บอยู่ในรูปของเท็กซ์ไฟล์ (Text File) ธรรมดาที่ไม่มีการตั้งค่ารูปแบบใดๆ ซึ่งผลลัพธ์ที่ได้จะเป็นกราฟซึ่งเรียงตามขนาดของเท็กซ์ไฟล์ดังรูปที่ 5.1



รูปที่ 5.1 กราฟแสดงขนาดของเท็กซ์ไฟล์และไฟล์ออโตมาตาถ่วงน้ำหนัก

โดยจะสามารถคำนวณปริมาณข้อมูลที่ลดลงเป็นเปอร์เซ็นได้ดังกราฟรูปที่ 5.2



รูปที่ 5.2 กราฟแสดงขนาดของข้อมูลที่ลดลง

จากกราฟทั้ง 2 รูปจะสังเกตเห็นว่าขนาดของข้อความในช่วงแรกๆ นั้น เมื่อทำการบีบอัดข้อมูลแล้ว ขนาดของข้อมูลที่ทำการบีบอัดกลับมีขนาดของไฟล์ที่ใหญ่ขึ้น เพราะถ้าขนาดของข้อมูลมีขนาดเล็กหรือข้อความสั้น จะทำให้จำนวนตัวอักษรที่จะซ้ำกันก็มีโอกาสน้อยลงไปด้วย ทุกๆ ครั้งที่มีการแบ่งคำถ้าไม่สามารถหาโหนดที่มาแทนข้อความนั้นได้ จะต้องสร้างโหนดใหม่ โดยที่โหนดที่สร้างขึ้นนี้ ในบางครั้งนี้อาจจะไม่สามารถใช้เป็นโหนดอ้างอิงสำหรับโหนดอื่นๆ ได้เลย ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บโหนดนี้

แต่ในทางตรงกันข้ามเมื่อขนาดของข้อมูลมีขนาดที่ใหญ่ขึ้นไปเรื่อยๆ จะสังเกตได้ว่าขนาดของไฟล์ออโตมาตาแบบถ่วงน้ำหนักจะมีขนาดที่ลดลง ที่เป็นเช่นนี้ ก็เนื่องมาจากลักษณะของข้อมูลที่เมื่อมีขนาดใหญ่ขึ้น ก็จะมีตัวอักษรหรือข้อความที่ซ้ำกับเพิ่มมากขึ้นเรื่อยๆ ทำให้จำนวนโหนดของออโตมาตาแบบถ่วงน้ำหนักมีจำนวนที่ลดลง จึงสรุปได้ว่าการเข้ารหัสด้วยวิธีออโตมาตาแบบถ่วงน้ำหนักนั้น ใช้ได้ผลดีกับการเข้ารหัสภาษาไทย โดยเฉพาะไฟล์ข้อมูลขนาดใหญ่

เอกสารอ้างอิง

1 K. Culik II and J. Kari, "Image Compression Using Weighted Finite Automata". Computers and Graphics, vol. 17, no. 3, May/June 1993, pp. 305-313

2 K. Culik, J. Kari, and V. Valenta, "Compression of Silhouette-like images based on WFA", Data Compression Conference, 1997.

3 S. Mallat and Z. Zhang, "Mathching Pursuit with Time-Frequency Dictinaries", IEEE Transaction on Signal Processing, 1993.

4 U. Hafner, Lehrstuhl fur Inf., Wurzburg Univ., Germany "Refining Image Compression with Weighted Finite Automata", presented at the IEEE Data Compression Conference, March 1996, pp. 359-368.

5 F. Katritzke, "Refinements of Data Compression Using Weighted Fintie Automata", Ph.D. Dissertation, graph. Darst. – Siegen, Vniv., Diss., 2001.

6 K. Culik II and V. Valenta, "Finite Automata Compression of Bi-level and Simple Color Images", presented at The Data Compression Conference, Snowbird, Utah, 1996.

7 Y. Lin and H. Yen, "An ω-Automata Approach to the Representation of Bilevel Images", IEEE Transactions on Systems, Man, and Cybermetics-Part B:Cybernetics, 2003.

8 M. Giraud and D. Lavenier, "Linear Encoding Scheme for Weighted Finite Automata", CIAA 2004, Lncs 3317, 2005, pp. 146-155.

ภาคผนวก

ข้อความที่นำมาใช้ในการทดสอบ จะเป็นข้อความทั่วๆ ไปที่มีให้อ่านตามอินเตอร์เน็ต ซึ่งเรียงลำดับตามขนาดของไฟล์ ดังต่อไปนี้

ข้อความที่ (ชื่อไฟล์)	เนื้อหา
Text_1	ความคาดหวังต่อครูภาษาไทย ...
Text_2	เรื่องสั้น ฉ.๒๔๑๙...
Text_3	บทที่ 4 อนาคต....
Text_4	๒๙ กรกฎาคม วันภาษาไทยแห่งชาติ...
Text_5	เรื่องนี้เกิดขึ้นเมื่อ...
Text_6	เรื่องสั้น ฉ.๒๔๘๒...
Text_7	เรื่องสั้น ฉ.๒๔๒๓...
Text_8	เส้นทางเดินทัพ พระเจ้าตาก เลียบทะเลตะวันออก...
Text_9	รัศมี ชูทรงเดช คณะโบราณคดี มหาวิทยาลัยศิลปากร...
Text_10	เรื่องสั้น ฉ.๒๔๒๐...
Text_11	ม.ร.ว.คึกฤทธิ์ ปราโมช ที่ข้าพเจ้ารู้จัก...
Text_12	บทความเกี่ยวกับการสอนภาษาไทย...
Text_13	เปิดห้องพระสูตร...
Text_14	ศานติ ภักดีคำ ภาควิชาภาษาไทยและภาษาตะวันออก...
Text_15	พุทธศาสนิกในโลกร่วมสมัย...
Text_16	หัวใจของผมเพื่อเธอ ตอนที่ 1 เส้นทางชีวิตเด็กเทคนิค...
Text_17	บทที่ 1 ๖ ตุลา จากมุมมองนักวิชาการและนักเขียน...
Text_18	พุทธศาสนาในสยาม จากหายนะสู่วัฒนะ (ตอนที่ ๑)...
Text_19	บทที่ 3 สรุปข้อมูลเกี่ยวกับเหตุการณ์นองเลือด ๖ ตุลาคม ๒๕๑๙...
Text_20	บทที่ 2 เหตุการณ์ ๖ ตุลาเกิดขึ้นได้อย่างไร...