บทที่ 4

โครงสร้างของพจนานุกรม

ในบทนี้จะขอกล่าวถึงวิธีการจัดเก็บคำศัพท์ที่จะนำมาใช้ในการตัดคำ เนื่องจากการตัดคำด้วยพจนานุกรมจะต้องมีการสืบค้นหาคำศัพท์ในพจนานุกรมเป็นจำนวนมาก ทำให้ต้องมีการพิจารณานำโครงสร้างข้อมูลแบบต่างๆ ที่เหมาะสมเข้ามาใช้ในการจัดเก็บคำศัพท์ในพจนานุกรมเพื่อที่จะสามารถจัดเก็บคำศัพท์ได้อย่างมีประสิทธิภาพในแง่ของความรวดเร็วในการสืบค้น และใช้เนื้อที่หน่วยความจำน้อย

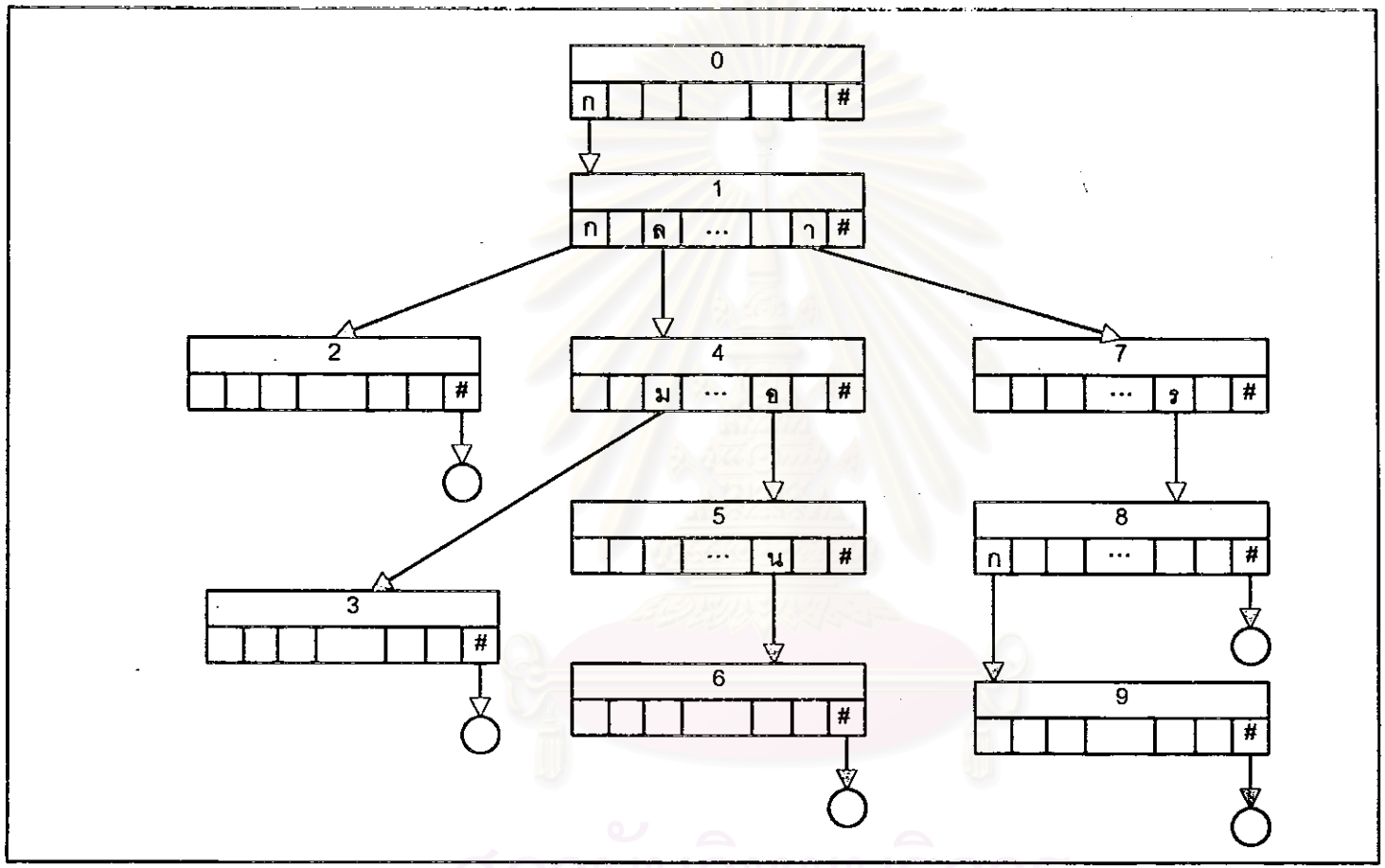
เนื่องจากการตัดคำส่วนมากจะนิยมทำการตัดคำจากซ้ายมาขวา ยกเว้นในกรณีที่นำการตัดคำมาใช้ในการขึ้นบรรทัดใหม่ (Word Wrap) เพียงอย่างเดียวเท่านั้นซึ่งอาจจะทำการตัดคำจากขวามาซ้ายก็ได้ สำหรับการตัดคำแบบเลือกคำที่ยาวที่สุด หรือการตัดคำโดยเลือกแบบที่เหมือนมากที่สุด หรือการตัดคำแบบอื่นๆ ที่ต้องมีการตัดคำที่เป็นไปได้ทุกๆ แบบก่อนแล้วค่อยนำมาประมวลผลต่อไป จะนิยมทำการตัดคำจากซ้ายมาขวามากกว่า ดังนั้นในวิทยานิพนธ์นี้จะกล่าวถึงโครงสร้างของพจนานุกรมที่เหมาะสมที่จะนำมาใช้กับการตัดคำจากซ้ายไปขวาเท่านั้น

สำหรับโครงสร้างของพจนานุกรมที่นำมาใช้ในวิทยานิพนธ์นี้คือโครงสร้างข้อมูลแบบทรัย (Trie) ซึ่งในส่วนถัดไปจะอธิบายถึงโครงสร้างข้อมูลแบบทรัย และสำหรับเหตุผลที่นำโครงสร้างข้อมูลแบบทรัยมาจัดเก็บพจนานุกรมนั้นจะอธิบายในส่วนถัดไป

4.1 โครงสร้างข้อมูลแบบทรัย

โครงสร้างข้อมูลแบบทรัย (Corman, Leiserson and Rivest, 1990: Frakes and Baeza-Yates, 1992) จะมีลักษณะคล้ายกับโครงสร้างข้อมูลแบบต้นไม้ แต่วิธีการจัดเก็บข้อมูลจะแตกต่างกัน โดยที่โครงสร้างข้อมูลแบบทรัยนี้จะจัดเก็บตัวอักษรของคำศัพท์ ซึ่งโครงสร้างข้อมูลแบบต้นไม้นั้นจะจัดเก็บข้อมูลทั้งคำ สำหรับโครงสร้างข้อมูลแบบทรัย แสดงในรูปที่ 4-1

จากรูป 4-1 เป็นตัวอย่างโครงสร้างข้อมูลแบบทรัย ที่ใช้ในการจัดเก็บคำศัพท์ กก กลม กลอน การ และ การก



รูปที่ 4-1 โครงสร้างข้อมูลแบบทรัย

จากรูป 4-1 โครงสร้างของทรัยจะประกอบไปด้วยโหนดต่างๆ โดยที่ข้อมูลภายใน 1 โหนดจะประกอบไปด้วย พอยเตอร์ที่ชี้ไปยังโหนดของตัวอักษรถัดไป ซึ่งมีจำนวนพอยเตอร์เท่ากับจำนวนตัวอักขระที่จะอนุญาตให้มีได้ในพจนานุกรมบวกกับอักขระที่ใช้ระบุเป็นตัวจบคำศัพท์ (Terminator) อีก 1 ตัวอักษร ซึ่งสัญลักษณ์ที่ใช้ในที่นี้คือเครื่องหมาย #

สำหรับการสืบค้นในโครงสร้างข้อมูลแบบทรัยนี้จะทำโดย เริ่มต้นที่โหนด 0 ถ้าต้องการค้นหาคำศัพท์ก็ให้นำอักษรทีละตัวจากคำศัพท์ที่ต้องการ มาดูว่าภายในโหนด 0 นั้นมีพอยเตอร์ของตัวอักษรที่ต้องการชี้ไปโหนดอื่นหรือไม่ ถ้าไม่มีแสดงว่าคำนั้นไม่มีอยู่ในพจนานุกรม แต่ถ้ามีพอยเตอร์ที่ชี้ไปที่โหนดถัดไปก็ให้เดินที่โหนดที่พอยเตอร์นั้นชี้ไป แล้วนำตัวอักษรตัวถัดไปมาทำตามขั้นตอนแบบเดิมจนหมด เมื่อนำตัวอักษรทั้งหมดจากคีย์มาเดินในทรัยแล้ว ให้เดินด้วยอักษร "#" แล้วดูว่าค่าพอยเตอร์มีค่าเท่ากับค่าว่าง (null) หรือไม่ ถ้าเท่าแสดงว่าไม่มีคำศัพท์นั้นในพจนานุกรม แต่ถ้าไม่เท่าก็แสดงว่ามีคำศัพท์นั้นอยู่ในพจนานุกรม โดยพอยเตอร์นี้ส่วนใหญ่จะชี้ไปที่ตำแหน่งของข้อมูลของคำนั้น

ตัวอย่างการสืบค้นคำศัพท์จากโครงสร้างข้อมูลแบบทรัย จากรูปที่ 4-1 ถ้าต้องการสืบค้นคำว่า "กลม" มีขั้นตอนดังนี้คือ โหนด 0 จะเป็นโหนดเริ่มต้น ดังนั้นนำตัวอักษร "ก" เข้ามาเดินภายในทรัยก็จะไปที่โหนด 1 หลังจากนั้นก็นำตัวอักษร "ล" เข้ามาเดินต่อจะไปที่โหนด 4 แล้วก็นำตัวอักษรตัวถัดไปคือ "ม" เข้ามาเดินจะไปที่โหนด 3 สุดท้ายเมื่อทำการค้นหามาถึงตัวอักษรสุดท้ายของคีย์แล้ว ให้เดินด้วย "#" ซึ่งค่าที่ได้ไม่เท่ากับค่าว่างแสดงว่าคำว่า "กลม" มีอยู่ในพจนานุกรม

4.2 ประสิทธิภาพด้านความเร็ว

โครงสร้างของพจนานุกรมที่เหมาะสมที่จะนำมาใช้ในการตัดคำต้องมีความรวดเร็วในการทำงาน ซึ่งเวลาที่ใช้ในการค้นหาคำศัพท์ในพจนานุกรมนี้จะขึ้นอยู่กับจำนวนครั้งทั้งหมดที่ใช้ในการค้นหาคำ และเวลาที่ใช้ในการค้นหาแต่ละครั้ง ดังนั้นโครงสร้างของพจนานุกรมที่ดีจะต้องมีจำนวนครั้งในการค้นหาคำที่มีอยู่ในประโยคที่นำมาตัดคำนั้นเป็นจำนวนน้อย และเวลาที่ใช้ในการค้นหาแต่ละครั้งจะต้องไม่มาก

พจนานุกรมถูกนำมาใช้ในการตัดคำเมื่อต้องการหาบริเวณของกลุ่มตัวอักษรในประโยคที่เป็นคำศัพท์ในพจนานุกรม ดังนั้นถ้าประโยคที่จะนำเข้ามาตัดคำดังแสดงในสมการที่ 4-1

$$S=c_1c_2\ c_3\ c_4\ c_5...c_n \qquad (4\text{-}1)$$

จากสมการที่ 4-1 ให้ S คือประโยคที่นำมาตัดคำซึ่งจะประกอบไปด้วย $c_1c_2\ c_3\ c_4\ c_5...c_n$ โดยที่ c_i คือตัวอักษรภาษาไทย

สำหรับโครงสร้างของพจนานุกรมที่จะต้องมีการนำคีย์ (Key) ทั้งคีย์มาใช้ในการสืบค้นเช่น โครงสร้างข้อมูลแบบตารางแฮช (Hash Table) โครงสร้างข้อมูลแบบไบนารีทรี (Binary Tree) โครงสร้างข้อมูลแบบบีพลัส-ทรี (B^+ -Tree) หรือ โครงสร้างข้อมูลแบบอินเด็กซ์ซีเควนเชียล (Index Sequential) นั้นจำนวนครั้งที่ต้องใช้ในการหาคำศัพท์จากสมการที่ 4-1 จะใช้ทั้งหมดคือ $O(n^2)$ ครั้ง

แต่สำหรับโครงสร้างข้อมูลแบบทรัยนั้นไม่จำเป็นต้องใช้คีย์ทั้งคีย์มาใช้ในการสืบค้น สามารถทำการสืบค้นโดยนำเข้ามาทีละตัวอักษรได้ ดังนั้นทำให้จำนวนครั้งที่ต้องการในการหาคำศัพท์จากสมการที่ 4-1 ได้เพียง $O(n)$ ครั้ง ซึ่งจะใช้จำนวนครั้งในการสืบค้นน้อยกว่าโครงสร้างของพจนานุกรมที่จะต้องมีการเปรียบเทียบทั้งคีย์ ดังนั้นโครงสร้างข้อมูลแบบทรัยนี้จึงถือว่ามีประสิทธิภาพดีกว่า ในแง่จำนวนครั้งที่ใช้ในการสืบค้น

ส่วนความเร็วในการค้นหาภายในโครงสร้างข้อมูลแบบทรัยนั้นจะไม่ขึ้นอยู่กับจำนวนคำที่มีในพจนานุกรม แต่จะขึ้นอยู่กับความยาวของคีย์ที่ใช้ในการสืบค้น ซึ่งถือได้ว่าเป็นข้อดีสำหรับลักษณะโครงสร้างข้อมูลประเภทนี้ แต่สำหรับโครงสร้างข้อมูลแบบตารางแฮช โครงสร้างข้อมูลแบบไบนารีทรี โครงสร้างข้อมูลแบบบีพลัส-ทรี หรือ โครงสร้างข้อมูลแบบอินเด็กซ์ซีเควนเชียล นั้น เวลาที่ใช้ในการสืบค้นจะขึ้นอยู่กับจำนวนคำที่เก็บในพจนานุกรมด้วย

4.3 ประสิทธิภาพในการใช้หน่วยความจำ

จากลักษณะโครงสร้างการจัดเก็บข้อมูลแบบทรัยนี้ แสดงให้เห็นดังในรูปที่ 4-1 จะเห็นว่าทรัยนั้นต้องมีการจองเนื้อที่เป็นจำนวนมาก ซึ่งเนื้อที่ส่วนใหญ่จะไม่ได้ถูกใช้งาน ดังนั้นจึงได้มีผู้พัฒนาทรัยแบบใหม่ขึ้นมา ซึ่งจะช่วยลดเนื้อที่ในหน่วยความจำลง โดยโครงสร้างแบบใหม่มีชื่อว่า โครงสร้างทรัยแบบสามแถวลำดับ (Triple-Array Trie) ซึ่งพัฒนาโดยจอห์นสัน (Johnson ,1975) โดยที่วิธีการนี้จะมีการใช้แถวลำดับจำนวน 3 ชุดในการเก็บค่า ต่อมาได้มีการพัฒนาให้ใช้หน่วยความจำน้อยลงไปอีก วิธีการที่พัฒนาขึ้นมาใหม่นี้ได้พัฒนาโดยอะเอโอะ (Aoe, 1989) และโครงสร้างทรัยแบบใหม่ที่พัฒนาขึ้นมีชื่อว่า โครงสร้างทรัยแบบแถวลำดับคู่ ซึ่งในวิธีการนี้จะใช้แถวลำดับแค่เพียง 2 ชุดเท่านั้น ทำให้วิธีการนี้สามารถจะลดการใช้เนื้อได้มากขึ้น และจากลักษณะโครงสร้างข้อมูลแบบทรัยนั้น จะมีจุดเด่นคือมีการเก็บข้อความส่วนหน้าเข้าด้วยกัน ซึ่งเป็นสาเหตุให้โครงสร้างของทรัยนั้นมีขนาดเล็กกว่าโครงสร้างข้อมูลแบบอื่น

ส่วนในรายละเอียดของขั้นตอนวิธีการเพิ่มคำ ลดคำและ แก้ไขคำเข้าไปในโครงสร้างทรัยแบบแถวคู่นั้นสามารถศึกษาได้จาก (Aoe, 1989; สมปรารถนา รัทยานนท์, 2535)