บทที่ 3

แนวทางการออกแบบพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

แนวทางการออกแบบโครงสร้างข้อมูลสำหรับจัดเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยในวิทยานิพนธ์นี้พิจารณาจากวิธีการค้นหาคำศัพท์จากพจนานุกรมและลักษณะคำไทย สำหรับเนื้อหาที่จะกล่าวในบทนี้ประกอบด้วยลักษณะคำไทย โครงสร้างข้อมูลแบบดีเอส-ทรี (DS-TREE) หรือโครงสร้างข้อมูลแบบต้นไม้ที่มีการสืบค้นแบบดิจิตอล โดยแสดงให้เห็นถึงรูปแบบของโครงสร้างข้อมูล ขั้นตอนการเพิ่มคำศัพท์และการสืบค้นคำศัพท์

ลักษณะคำไทย

คำในภาษาไทยนั้นมี 2 ประเภท คือ คำโดด และคำประสม [14] คำโดดหมายถึงคำคำเดียวที่มีความหมายในตัวเอง เช่น เดิน นั่ง วิ่ง เป็นต้น ส่วนคำประสมหมายถึงคำที่เกิดจากการนำคำโดดมารวมกันแล้วเกิดเป็นคำใหม่ที่มีความหมาย เช่น กางใบ เรือใบ กาน้ำ เป็นต้น

นอกจากนี้การแบ่งคำในภาษาไทยยังแตกต่างจากการแบ่งคำในภาษาอังกฤษ เนื่องจากวิธีการเขียนประโยคภาษาไทยนั้นเขียนติดต่อโดยไม่มีช่องว่างดัง เช่นวิธีการเขียนประโยคภาษาอังกฤษซึ่งเขียนโดยมีช่องว่างระหว่างคำ ทำให้ไม่สามารถทราบถึงตำแหน่งสิ้นสุดคำในประโยคภาษาไทยที่แน่นอน เพราะฉะนั้นการเลือกโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมและวิธีปฏิบัติการจะต้องนำลักษณะคำไทยและการแบ่งคำมาเป็นหลักสำหรับการพิจารณาด้วย

การค้นหาคำศัพท์จากพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

พจนานุกรมอิเล็กทรอนิกส์ภาษาไทยเป็นปัจจัยที่สำคัญในงานประมวลผลด้านภาษาศาสตร์ เนื่องจากผู้ใช้สามารถค้นหารายละเอียดต่างๆ จากพจนานุกรมที่จัดเก็บคำศัพท์พร้อมสารสนเทศต่างๆ ได้ทันที นอกจากนี้โครงสร้างข้อมูลที่ใช้จัดเก็บพจนานุกรมจะต้องเป็นโครงสร้างข้อมูลที่เลือกและทดสอบแล้วว่าเหมาะสมและประกอบด้วยวิธีการสืบค้นคำศัพท์

ที่ใช้เวลาสืบค้นที่รวดเร็ว แตกต่างจากการเปิดพจนานุกรมที่หากผู้ใดมีเทคนิคในการใช้พจนานกรมที่ดีและใช้พจนานุกรมบ่อยๆ จะทำให้ระยะเวลาที่ใช้ในการสืบค้นคำศัพท์ที่ต้องการรวดเร็วยิ่งขึ้น สำหรับการสืบค้นคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์นั้นไม่ใช้ประสบการณ์แต่เวลาที่ใช้ในการสืบค้นคำศัพท์ขึ้นกับปัจจัย 2 ประการ ประการแรกคือลักษณะโครงสร้างข้อมูลที่ใช้จัดเก็บคำศัพท์ ประการที่ 2 คือวิธีการสืบค้นคำศัพท์จากโครงสร้างข้อมูลดังกล่าว วิธีการเพิ่มคำศัพท์ในพจนานุกรมนั้น ขั้นแรกจะต้องกำหนดคำสำคัญที่ใช้ สำหรับสืบค้นอาจใช้คำศัพท์ทั้งคำหรือส่วนของคำเป็นคำสำคัญก็ได้ เช่น "การเรียน" อาจใช้ "การ" หรือ "การเรียน" เป็นคำสำคัญก็ได้ นอกจากนี้พจนานุกรมอิเล็กทรอนิกส์ที่ดีจะต้องสามารถแสดงให้เห็นถึงความแตกต่างของคำศัพท์ที่เป็นคำกำกวม เช่น คำว่า "ขัน" ซึ่งมีความหมาย 2 อย่าง คือกรณีที่เป็นคำนามหมายถึงภาชนะ และกรณีที่เป็นคำกริยาหมายถึงกริยาการร้องของไก่ ดังนั้นในพจนานุกรมจะต้องจัดเก็บคำศัพท์พร้อมความหมายทั้ง 2 อย่าง อีกตัวอย่างหนึ่งคือ กรณีที่ต้องการสืบค้นคำ "นา" แต่ในพจนานุกรมอิเล็กทรอนิกส์เก็บคำ "นาฬิกา" ผลลัพธ์จากการสืบค้นจะได้ว่าไม่พบคำว่า "นา" ในพจนานุกรมอิเล็กทรอนิกส์

วิธีการสืบค้นคำศัพท์จากพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยนั้นสามารถใช้วิธีการแบบง่ายๆ ได้ ถ้าผู้ใช้ไม่ได้ให้ความสำคัญของความรวดเร็วในการสืบค้นคำศัพท์ แต่โดยทั่วไปแล้วจำเป็นต้องนำพจนานุกรมอิเล็กทรอนิกส์ไปใช้กับโปรแกรมประยุกต์อื่น ดังนั้นการออกแบบโครงสร้างข้อมูลสำหรับจัดเก็บพจนานุกรมอิเล็กทรอนิกส์และวิธีการสืบค้นคำศัพท์จะต้องคำนึงถึงความเร็วและปริมาณเนื้อที่ที่ใช้สำหรับเก็บพจนานุกรมด้วย เพื่อมิให้ขั้นตอนวิธีสืบค้นคำศัพท์ซึ่งเป็นส่วนที่เพิ่มเข้าไปในโปรแกรมประยุกต์ไปลดประสิทธิภาพการทำงานของโปรแกรมประยุกต์ นอกจากนี้แล้วจะต้องนำลักษณะการเรียบเรียงประโยคภาษาไทยมาร่วมในการพิจารณาการออกแบบโครงสร้างพจนานุกรมอิเล็กทรอนิกส์ภาษาไทยนี้ด้วย

ปัจจุบันโปรแกรมประยุกต์ที่พัฒนาสำหรับใช้ในงานด้านการสืบค้นสารสนเทศมีอยู่มากมาย ผู้พัฒนาโปรแกรมนิยมใช้ขั้นตอนวิธีการสืบค้นสารสนเทศจากโครงสร้างข้อมูลแบบดิจิตอล (Fast Digital Search) หรือใช้วิธีการสืบค้นสารสนเทศจากข้อมูลที่จัดเก็บโดยใช้โครงสร้างแบบทรัย (Trie Search) ซึ่งขั้นตอนวิธีทั้งสองแบบนั้นจะปฏิบัติการกับข้อมูลโดยพิจารณาข้อมูลอินพุทครั้งละ 1 ตัวอักษรไปเรื่อยๆ จนกระทั่งอินพุทตัวสุดท้าย ข้อดีของขั้นตอนวิธีการทั้งสองแบบมีหลายประการ เช่น ผู้ใช้งานสามารถเพิ่มเติมข้อมูลได้ตามความพอใจ และสามารถนำเนื้อที่ที่ว่างอันเกิดจากขั้นตอนวิธีการลบข้อมูลมาใช้ได้ในขณะที่เพิ่มเติมข้อมูลโดยไม่จำเป็นต้องใช้ฟังก์ชันปรับปรุงโครงสร้างข้อมูล (Reorganization) สำหรับจัดระเบียบการใช้เนื้อที่ของโครงสร้างให้เหมาะสม เพราะฉะนั้นวิธีการทั้งสองแบบเหมาะที่จะใช้สำหรับเก็บข้อมูลจำนวนมากๆ และการเพิ่มเติมข้อมูลบ่อยๆ ตัวอย่างของงานการประมวลผลที่นำวิธีการสืบค้นข้อมูลดังกล่าวมาใช้คือ การวิเคราะห์เล็กซิคอน (Lexicon

Analyzer) การตรวจสอบตัวสะกด (Spell Checker) การหาความถี่ของการใช้คำศัพท์และการวิเคราะห์ระดับคำ (Morphological analyzer) ในงานการประมวลผลภาษาธรรมชาติ เป็นต้น

วิธีการปฏิบัติการกับข้อมูล

เราสามารถแบ่งยุทธวิธีปฏิบัติการกับข้อมูล โดยใช้รูปแบบการเก็บชุดของคำสำคัญออกเป็น 2 ประเภท คือการปฏิบัติการแบบพลวัต และการปฏิบัติการแบบสถิต

1. การปฏิบัติการแบบพลวัต (Dynamic Method)

วิธีการแบบพลวัตเป็นการปฏิบัติการกับข้อมูลที่จัดเก็บในรูปตารางที่มีการเปลี่ยนแปลงได้ขณะปฏิบัติการกับข้อมูล ตัวอย่างของงานในรูปแบบนี้ เช่น การหาเลขที่อยู่แบบแฮช (Hashing) ต้นไม้แบบทวิภาค (Binary Tree) ต้นไม้แบบบีพลัส (B^+-Tree) การหาเลขที่อยู่ที่อยู่ในเนื้อที่ขยายแบบแฮช (Extensible Hashing) การหาเลขที่อยู่แบบแฮชกับโครงสร้างข้อมูลแบบทรัย (Trie Hashing)

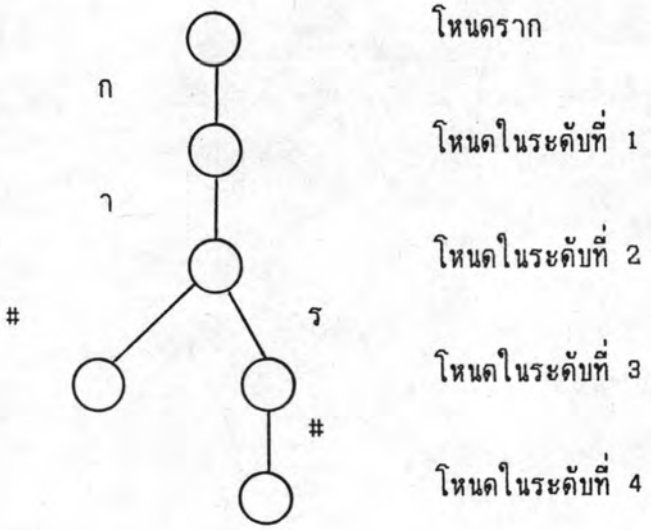
2. การปฏิบัติการแบบสถิต (Static Method)

วิธีการแบบสถิตเป็นการปฏิบัติการกับโครงสร้างข้อมูลที่ไม่มีการเปลี่ยนแปลงหรือตารางตายตัว ตัวอย่างของวิธีการดังกล่าวก็คือ การหาเลขที่อยู่แฮชแบบสมบูรณ์ (Perfect Hashing) ตารางที่ใช้แทนค่าข้อมูลที่เก็บแบบกระจัดกระจาย (Sparse Table Representation) และทรัยแบบลดขนาด (Compressed Trie)

วิธีการปฏิบัติการกับโครงสร้างข้อมูลทั้ง 2 แบบมีข้อดีและข้อด้อยแตกต่างกัน ข้อดีของวิธีการปฏิบัติการแบบสถิตคือโครงสร้างข้อมูลมีขนาดเล็กกระทัดรัดและสามารถสืบค้นข้อมูลได้รวดเร็ว ส่วนข้อดีของวิธีการปฏิบัติแบบพลวัตคือโครงสร้างข้อมูลมีการเปลี่ยนแปลงในขณะมีการปรับทันกาลของข้อมูล (Update) เพราะฉะนั้นจำเป็นต้องจัดเตรียมเนื้อที่ว่างไว้ให้มากพอเพื่อเอื้ออำนวยให้การปฏิบัติการดังกล่าวกระทำในเวลาอันรวดเร็ว สำหรับการทำวิทยานิพนธ์ครั้งนี้ได้นำข้อดีของการปฏิบัติการแบบสถิตมาผสมผสานกับข้อดีของการปฏิบัติการแบบพลวัต [6] แล้วปรับเป็นขั้นตอนที่ใช้ปฏิบัติกับโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

โครงสร้างข้อมูลแบบดีเอส-ทรี (DS-Tree)

ดีเอส-ทรี ย่อมาจาก ดิจิตอลเสริช-ทรี (Digital Search Tree) หมายถึง โครงสร้างข้อมูลแบบต้นไม้ที่มีการสืบค้นแบบดิจิตอล ดังนั้นโครงสร้างแบบดีเอส-ทรีจึงเป็นโครงสร้างข้อมูลที่ประกอบด้วยโหนดต่างๆ และเส้นทางเดินระหว่างโหนด โหนดเริ่มต้นเรียกว่าโหนดราก (Root Node) ส่วนโหนดอื่นๆ ภายในโครงสร้างเรียกว่าโหนดใบ (Leaf Node) แตกต่างจากโครงสร้างข้อมูลแบบต้นไม้ที่เส้นทางเดินระหว่างโหนดของโครงสร้างข้อมูลแบบต้นไม้เป็นการแทนค่าตัวอักษรแต่ละตัวที่ประกอบเป็นคำศัพท์ส่วนเส้นทางเดินระหว่างโหนดของดีเอส-ทรีเป็นผลลัพธ์ที่ได้มาจากการคำนวณระหว่างตัวเลขที่ได้จากการแปลงค่าตัวอักษรที่ประกอบเป็นคำศัพท์หรือพรีฟิกซ์ (Prefix) กับค่าที่เก็บในโหนดก่อนหน้า สำหรับขั้นตอนการเพิ่มหรือการสืบค้นคำศัพท์ของโครงสร้างแบบนี้ได้มาจากการสร้างเส้นทางที่เริ่มต้นจากโหนดราก จนกระทั่งตำแหน่งสุดท้ายของศัพท์ทุกคำโดยจะใช้สัญญลักษณ์ "#" เพื่อบอกความแตกต่างระหว่างตัวพรีฟิกซ์กับคำศัพท์ และมีผลทำให้ทราบว่า "กา" และ "การ" ที่เก็บอยู่ในพจนานุกรมอิเล็กทรอนิกส์มี 2 คำ ไม่ใช่มีคำว่า "การ" คำเพียงคำเดียว แสดงตามรูปที่ 3.1



รูป 3.1 โครงสร้างแบบดีเอส-ทรี

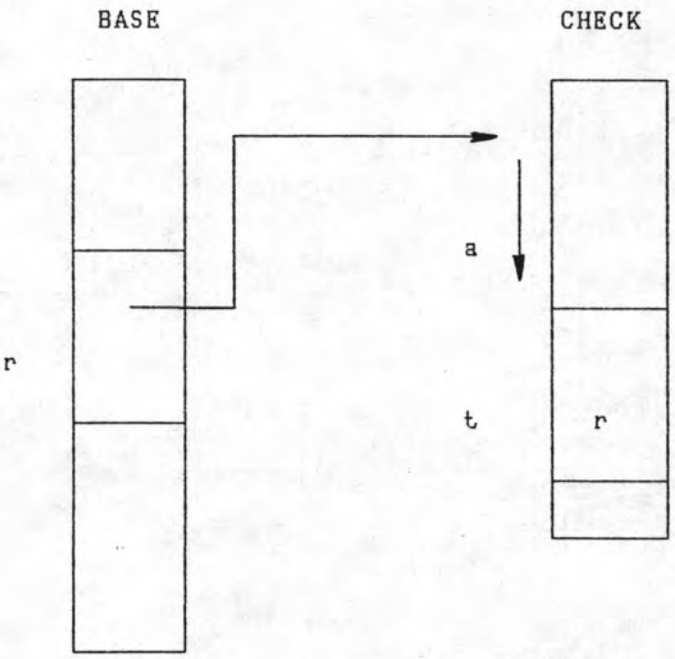
ขั้นตอนวิธีการเพิ่มข้อมูลและการสืบค้นข้อมูลแบบดีเอส-ทรีนั้น มีการอ้างถึงสัญญลักษณ์ต่างๆ มากมาย เพราะฉะนั้นก่อนที่จะอธิบายถึงขั้นตอนวิธีการเพิ่มและสืบค้นคำศัพท์ จะแสดงรายละเอียดของสัญญลักษณ์ต่างๆ ที่ใช้ดังนี้

1. S แทนเซตจำกัดของโหนด
2. K แทนเซตของคำศัพท์
3. I แทนเซตของอินพุท กล่าวคือเซตของตัวอักษร
4. g (goto function) หมายถึงฟังก์ชันจาก S x I ไปยัง S U {fail}
5. s_1 แทนโหนดเริ่มต้นหรือโหนดรากในเซต S
6. A แทนเซตจำกัดของโหนดรับคำ (Accepting Node)
7. s_r เป็นโหนดที่เป็นสมาชิกของเซต A ก็ต่อเมื่อเส้นทางเดินทางโหนด s_1 ไปยังโหนด s_r ให้ผลลัพธ์เป็นตัวสะกดของคำว่า X# ในเซต K ได้
8. อาร์ค a เป็นเส้นทางเดินจากโหนด s_r ไปยัง s_t ได้มาจากฟังก์ชัน $g(s_r,a) = s_t$
   กรณีที่ไม่สามารถสร้างอาร์คระหว่างโหนดจากฟังก์ชัน goto ดังกล่าวได้ แล้วเรียกว่าเฟล (fail)
9. กำหนดให้ a, b, c, d, e $\in$ I U{#}; x, y, z $\in$ (I U{#})* และถ้า ε เป็นสตริงว่าง แล้วจะได้ว่าฟังก์ชัน goto ที่ใช้แทนสตริงในเซต K แสดงได้ตามสมการ
   $g(s_r, \varepsilon) = s_r$, $g(s_r, ax) = g(g(s_r,a)x)$

<u>ลักษณะการเก็บข้อมูลด้วยโครงสร้างข้อมูลแบบดับเบิลอะเรย์</u>

อะเอโอะ (Aoe) นักวิจัยชาวญี่ปุ่นได้ทำการศึกษาและปรับปรุงโครงสร้างข้อมูลรูปแบบหนึ่งสำหรับใช้กับโครงสร้างข้อมูลแบบดีเอส-ทรี โดยให้ชื่อโครงสร้างข้อมูลที่ทำการปรับปรุงว่า "โครงสร้างข้อมูลแบบดับเบิลอะเรย์" [12] [13] โครงสร้างนี้นำหลักการมาจากโครงสร้างข้อมูลแบบทริปเปิ้ลอะเรย์ที่ใช้งานกับทรานซิชั่นเทเบิ้ลสำหรับแยคซ์ (YACC) [5] แล้วทำการปรับปรุงโครงสร้างดังกล่าวโดยมีข้อกำหนดว่าจะนำมาประยุกต์ใช้กับทรานซิชั่นเทเบิลสำหรับสตริงแพทเทินแมชชิงแมชชีน (String Pattern Matching Mechine) โครงสร้างข้อมูลแบบดับเบิลอะเรย์ประกอบด้วยอะเรย์ขนาด 1 มิติ จำนวน 2 อะเรย์ คือ อะเรย์เบส (Base) อะเรย์เช็ค (Check) มีหมายเลขประจำโหนดเป็นดรรชนีเพื่อแสดงความสัมพันธ์ระหว่างเบสและเช็ค ซึ่งรูปแบบของโครงสร้างข้อมูลแบบ

ดับเบิลอะเรย์แสดงตามรูปที่ 3.2



รูปที่ 3.2 ลักษณะโครงสร้างข้อมูลแบบดับเบิลอะเรย์

จากรูป 3.2 จะเห็นได้ว่าโหนด s_r และโหนด s_t ใดๆ ซึ่งเป็นสมาชิกของเซต S ได้มาจากฟังก์ชัน $g(s_r,a) = s_t$ ก็ต่อเมื่อดับเบิลอะเรย์ที่แสดงเซต K หรือของคำศัพท์นั้นเป็นไปตามสมการ BASE[r] + a = t และ CHECK[t] = r

กล่าวอีกนัยหนึ่งก็คือโหนด s_r เป็นสมาชิกของเซต S และมีความสัมพันธ์แบบโดยตรงกับค่าดรรชนี r หรือหมายเลขโหนดของดับเบิลอะเรย์ ซึ่งหมายเลขของโหนดถัดไปและเส้นทางระหว่างโหนดของโครงสร้างข้อมูลแบบดับเบิลอะเรย์นั้น ได้มาจากการคำนวณค่าของ BASE[r] กับค่าตัวเลขของตัวอักษรอินพุทตำแหน่งปัจจุบัน

โครงสร้างแบบดับเบิลอะเรย์นั้นประกอบด้วยโหนดต่างๆ และโหนดเหล่านั้นมีความสัมพันธ์กันอีกด้วย

1. <u>ประเภทของโหนด</u>

โครงสร้างข้อมูลแบบดับเบิลอะเรย์ประกอบด้วยโหนดจำนวนมาก ซึ่งโหนดแต่ละโหนดมีหน้าที่ที่แตกต่างกัน สำหรับโหนดของโครงสร้างข้อมูลแบบดับเบิลอะเรย์มีดังต่อไปนี้

1.1 เซพพาเรทโหนด (Separate Node) ใช้สัญญลักษณ์ S_p แทน หมายถึงโหนดที่ทำให้ทราบว่าคำศัพท์คำหนึ่งแตกต่างจากคำศัพท์คำอื่นที่เป็นสมาชิกในเซต K

นิยาม ถ้า xay เป็นคำศัพท์คำหนึ่งแล้ว s_i จะเป็นเซพพาเรทโหนด เมื่อ สมการ $g(s_i,xa) = s_r$ ถ้า a เป็นสัญญลักษณ์ที่ทำให้เห็นว่าคำศัพท์ xay ต่างจากคำศัพท์คำอื่นที่เป็นสมาชิกของเซต K

1.2 มัลติโหนด (Multi Node) ใช้สัญญลักษณ์ S_m แทน หมายถึงโหนดทุกๆ โหนดที่อยู่ในเส้นทางจากโหนดรากมายังเซพพาเรทโหนด

1.3 ซิงเกิลโหนด (Single Node) ใช้สัญญลักษณ์ S_i แทน หมายถึงโหนดทุกๆ โหนดที่อยู่ระหว่างเส้นทางจากเซพพาเรทโหนดมายังแอคเซพติงโหนด

จากนิยามของโหนดทั้ง 3 ประเภท สามารถสรุปได้ว่า เซพพาเรทโหนดเป็นโหนดที่เป็นผลลัพธ์ของโอเปอร์เรชันอินเตอร์เซ็คชันของมัลติโหนดกับซิงเกิลโหนด หรืออาจเขียนแทนด้วยสัญญลักษณ์ทางคณิตศาสตร์เป็น $S_p = S_m \cap S_i$

2. องค์ประกอบโครงสร้างแบบดับเบิลอะเรย์

นิยาม กำหนดให้สตริง x ซึ่งสอดคล้องกับฟังก์ชัน $g(s_r,x) = s_t$ โดยที่โหนด s_r เป็นเซพพาเรทโหนด และโหนด s_t เป็นสมาชิกของเซต A แล้ว จะถือว่า x เป็นซิงเกิลสตริงของเซพพาเรทโหนด s_r สามารถเขียนแทนด้วย $STR[s_r]$

จากนิยามดังกล่าว พบว่าโครงสร้างข้อมูลแบบดับเบิลอะเรย์นั้นจะถูกแยกเป็น 2 ส่วน คือ เส้นทางเดินของโหนดที่สร้างจากฟังก์ชัน $S_m \times (I \cup (\#))$ ไปยัง S_m เก็บไว้ในดับเบิลอะเรย์ (อะเรย์เบส และอะเรย์เช็ค) ส่วนเส้นทางเดินของโหนดที่สร้างจากฟังก์ชัน $S_i \times (I \cup (\#))$ ไปยัง S_i เป็นซิงเกิลสตริงเก็บไว้อะเรย์เทล นั่นคือสตริงอินพุทแต่ละชุดนั้นจะแยกพิจารณาเป็น 2 ส่วน

ส่วนแรก นำสตริงอินพุทมาสร้างมัลติโหนดและเซพพาเรทโหนด

ส่วนที่สอง ถ้าตัวอักษรที่ตำแหน่งใดในสตริงอินพุทไม่สามารถสร้างเซพพาเรทโหนดได้แล้ว จะสร้างซิงเกิลโหนดพร้อมกับจัดเก็บตำแหน่งที่จะดึงชุดของซิงเกิลโหนด (ซิงเกิลสตริง) จากเทลไว้ในเซพพาเรทโหนด (อะเรย์เบส)

3. ความสัมพันธ์ภายในโครงสร้างแบบดับเบิลอะเรย์

สมมติว่ามีดับเบิลอะเรย์และอะเรย์เทลชุดหนึ่งที่ใช้เก็บชุดของคำศัพท์ที่เก็บในเซต K เราสามารถแสดงความสัมพันธ์ของสมาชิกในดับเบิลอะเรย์ และอะเรย์เทล

ได้ดังนี้

3.1 มัลติโหนด s_r และ s_t เป็นโหนดที่เกิดมาจากฟังก์ชัน $g(s_r,a) = s_t$ ก็ต่อเมื่อค่า BASE[r] + a = t และ CHECK[t] = x

3.2 ถ้า s_r เป็นเซพพาเรทโหนด ที่สามารถสร้างซิงเกิลสตริงได้จากฟังก์ชัน STR[s_r] = $b_1 b_2 \ldots b_m$ (ค่าของ m มากกว่า 0) และค่าของเซพพาเรทโหนด s_r (BASE[r]) น้อยกว่า 0 แล้วจะได้ว่า ตำแหน่งของซิงเกิลโหนดในอะเรย์เทล (p) เท่ากับค่าของเซพพาเรทโหนดที่ไม่คิดเครื่องหมาย (-BASE[r]) นั่นเอง ซึ่งอาจแสดงค่าซิงเกิลโหนดที่เก็บในอะเรย์เทลได้ดังนี้

TAIL[p] = b_1

TAIL[p+1] = b_2

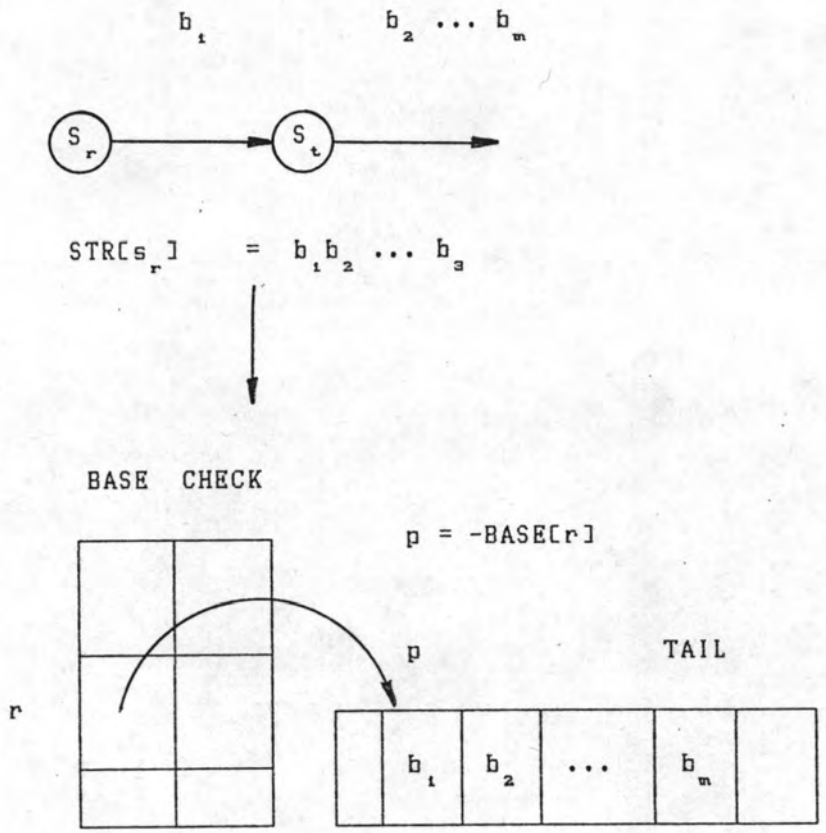
.

.

.

TAIL[p+m-1] = b_m

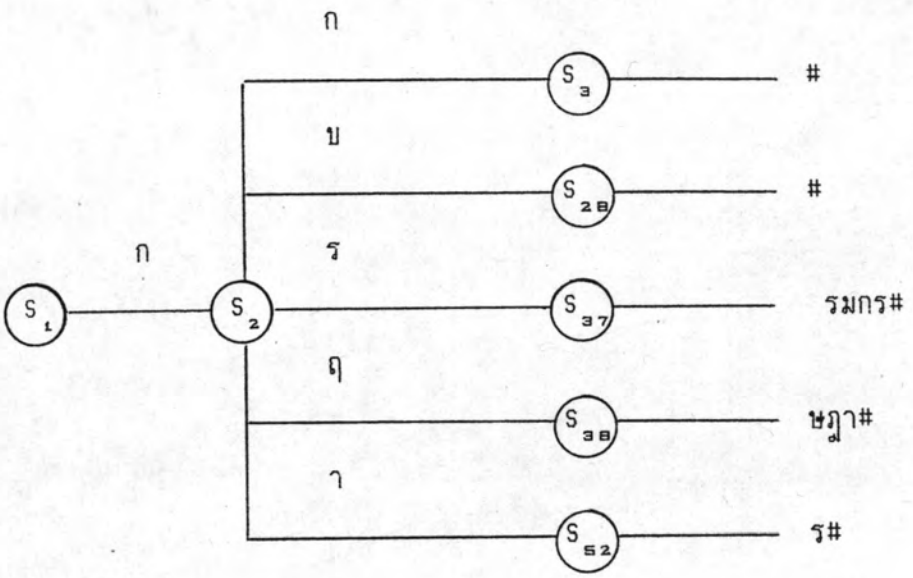
ส่วนความสัมพันธ์ระหว่างดับเบิลอะเรย์และอะเรย์เทลแสดงได้ตามรูปที่ 3.3



รูปที่ 3.3 ความสัมพันธ์ของโครงสร้างแบบดับเบิลอะเรย์

ค่าที่เก็บในมัลติโหนด หรือค่าของ BASE[r] นั้นมีอยู่ 2 แบบ คือ ถ้ามีค่าเป็นบวกหมายถึงหมายเลขของมัลติโหนดโหนดถัดไป แต่ถ้าค่าเป็นลบก็จะหมายถึงตำแหน่งของชิงเกิลสตริงที่จัดเก็บไว้ในอะเรย์เทล เราเรียกโหนดที่ค่า BASE[r] เป็นลบว่าเซพพาเรทโหนดแทน อันที่จริงแล้วดับเบิลอะเรย์หมายถึง อะเรย์เบส กับอะเรย์เช็ค จะประกอบด้วยตารางเก็บรายการสืบค้นข้อมูลภายใน (internal retrieval table) ส่วนอะเรย์เทลนั้นเก็บชิงเกิลสตริงและหมายเลขระเบียนที่เก็บสารสนเทศอื่นๆ ซึ่งใช้สำหรับการสืบค้นคำศัพท์ เพราะฉะนั้นจึงเพิ่มสัญญลักษณ์ "$" เพื่อใช้สำหรับเก็บสารสนเทศ และสัญญลักษณ์ "?" ซึ่งหมายถึงการ์เบจซิมโบลใช้สำหรับระบุว่าข้อมูลในอะเรย์เทลตำแหน่งนี้ไม่ใช้งาน

สมมุติว่ามีพจนานุกรมอิเล็กทรอนิกส์ที่ใช้เก็บชุดคำศัพท์ชุดต่อไปนี้คือ {กก#, กบ#, การ#, กรรมกร#, กฤษฎา#} ซึ่งโครงสร้างข้อมูลของพจนานุกรมอิเล็กทรอนิกส์นี้แสดงตามรูปที่ 3.4 ส่วนรูปที่ 3.5 เป็นรูปที่แสดงโครงสร้างของดับเบิลอะเรย์และอะเรย์เทล



รูปที่ 3.4 ดีเอส-ทรี ของคำศัพท์ตัวอย่าง

หมายเลขดรรชนี

	1	2	3		28		37	38		52
เบส	1	2	1		4		9	15		6
เช็ค	52	1	2		2		2	2		2

#$?#$ร#$รมกร#$ษฎา#$

รูปที่ 3.5 ดับเบิลอะเรย์และเทล

จากรูปที่ 3.4 และ 3.5 สามารถสรุปได้ว่า

เซพพาเรทโหนด คือ s_3 s_{28} s_{37} s_{38} s_{52}

มัลติโหนด คือ s_1 s_2

ซิงเกิลโหนด คือ s_{53} ... s_{66}

เนื่องจากโหนดและเส้นทางเดินภายในของโครงสร้างข้อมูลแบบดับเบิลอะเรย์นั้นได้มาจากการแทนค่าพยัญชนะด้วยตัวเลข กำหนดให้พยัญชนะและสระคือ ก, ข, ค, ง, จ, ... ฮ, ะ, า, ... แทนด้วยค่า 1, 2, 3, ... 77 ส่วน # แทนด้วยค่า 78

จากตัวอย่างพบว่าเซพพาเรทโหนด s_{28} สร้างมาจากโหนด s_2 กับ "บ" เขียนแทนด้วยฟังก์ชัน $g(s_2,26)$ ส่วนค่าของอะเรย์เบสคำนวณจากสมการข้างล่างนี้

BASE[2] = 2 + 26 = 28 = t

และค่าของ CHECK[28] = 2 โดยที่ 2 เป็นดรรชนีของอะเรย์เบส แสดงว่าโหนด 28 สร้างมาจากโหนดที่ 2 ต่อจากนั้นพิจารณาค่าที่เก็บในโหนด 28 ซึ่งมีค่าเป็น -4 สรุปได้ว่าโหนด 28 เป็นเซพพาเรทโหนดและค่าที่เก็บเป็นตำแหน่งของซิงเกิลสตริงในอะเรย์เทล ซึ่งเมื่อดึงซิงเกิลสตริงจากเทลจะได้ "#"

<u>อัลกอริทึมการสืบค้นข้อมูล</u>

การสืบค้นคำศัพท์จากโครงสร้างข้อมูลแบบดับเบิลอะเรย์นั้นเป็นอัลกอริทึมที่ใช้สำหรับตรวจสอบว่าคำศัพท์ที่ต้องการสืบค้นมีในโครงสร้างหรือไม่มี ซึ่งขั้นตอนการสืบค้นมีดังนี้

1. ขั้นตอนการสืบค้น

กำหนดให้

x แทนคำศัพท์คำหนึ่งโดยที่ $x = a_1 a_2 \ldots a_n a_{n+1}$ a_{n+1} แทน #

r แทนหมายเลขโหนดปัจจุบัน

t แทนหมายเลขโหนดถัดไป

a_n แทนตัวอักษรใดๆ ของคำศัพท์

1. สร้างโหนดถัดไปจากโหนดที่ 1 และตัวอักษรตัวแรกของคำศัพท์ตามสมการ t = BASE[r] + a_n
2. ตรวจสอบว่าโหนดที่สร้างขึ้นมีค่าเกินกว่าขนาดของดับเบิลอะเรย์หรือไม่ ถ้ามากกว่าระบุว่าไม่พบคำศัพท์และออกจากการสืบค้น

3. ตรวจสอบว่าโหนดใหม่เป็นโหนดที่สร้างจากโหนดแรก โดยพิจารณาจากค่าของ CHECK[t] ว่าเท่ากับ r หรือไม่ ถ้าค่าไม่เท่าระบุว่าไม่พบคำศัพท์ออกจากการสืบค้น ถ้าค่าเท่ากันแสดงว่าเป็นโหนดที่สร้างจากโหนดแรกจริง
4. ตรวจสอบว่าโหนดที่สร้างใหม่เป็นโหนดประเภทใดโดยพิจารณาจากค่า BASE[r] ดังนี้

   เท่ากับ 0 แสดงว่าเป็นโหนดที่ยังไม่ได้ใช้งาน ระบุว่าไม่พบคำศัพท์ออกจากการสืบค้น

   มากกว่า 0 แสดงว่าเป็นมัลติโหนด

   น้อยกว่า 0 แสดงว่าเป็นเซพพาเรทโหนด ตรวจสอบว่าคำศัพท์หมดหรือยังถ้าหมดแล้วระบุว่าไม่พบคำศัพท์ออกจากการสืบค้น ถ้ายังไม่หมดให้ดึงชิงเกิลสตริงจากอะเรย์เทลโดยใช้ฟังก์ชัน FetchStr และค่าสัมบูรณ์ของอะเรย์เบสแทนตำแหน่งของชิงเกิลสตริง ต่อจากนั้นเปรียบเทียบคำศัพท์ส่วนที่เหลือหลังจากดึงตัวอักษรตัวแรกออกแล้วกับชิงเกิลสตริงถ้าเหมือนกันแสดงว่าพบคำศัพท์ออกจากการสืบค้น ถ้าแตกต่างกันแสดงว่าไม่พบคำศัพท์ออกจากการสืบค้น
5. กรณีที่เป็นมัลติโหนดสร้างโหนดต่อไปโดยใช้โหนดที่สร้างใหม่แทนโหนดปัจจุบันหรือกำหนด r = t และตัวอักษรตัวถัดไปของคำศัพท์ตามขั้นตอนที่ 2 ถึง 4 จนกระทั่งตัวอักษรทุกตัวของคำศัพท์หมดหรือออกจากการสืบค้นตามเงื่อนไขที่พบ

การสืบค้นคำศัพท์เป็นขั้นตอนที่ใช้ตรวจสอบว่าคำศัพท์ที่ต้องการสืบค้นปรากฏในโครงสร้างหรือไม่ ซึ่งขั้นตอนการสืบค้นคำศัพท์สรุปได้ดังนี้ คือทำการสร้างมัลติโหนดจากโหนดต่างๆ และตัวอักษรของคำศัพท์จนกระทั่งถึงเซพพาเรทโหนดแล้วดึงชิงเกิลสตริงขึ้นมาแล้วเปรียบเทียบกับคำศัพท์ส่วนที่เหลือจากการสร้างมัลติโหนดแล้วผลของการเปรียบเทียบเหมือนกันแสดงว่ามีคำศัพท์ในโครงสร้าง การสืบค้นตามวิธีการนี้ไม่พบคำศัพท์ 2 กรณี คือ สร้างมัลติโหนดไม่สำเร็จ และสร้างมัลติโหนดจนถึงเซพพาเรทโหนดแล้วนำชิงเกิลสตริงจากอะเรย์เทลมาเปรียบเทียบกับคำศัพท์ส่วนที่เหลือหลังจากสร้างมัลติโหนดแล้วแตกต่างกัน

การสืบค้นคำศัพท์ประกอบด้วยฟังก์ชันสนับสนุนอีก 2 ฟังก์ชัน คือ ฟังก์ชัน FetchStr(p) และ ฟังก์ชัน StrCmp(x,y) หน้าที่ของฟังก์ชันทั้ง 2 มีดังนี้ คือ

ฟังก์ชัน FetchStr(p) เป็นฟังก์ชันที่ใช้สำหรับดึงชุดของชิงเกิลสตริง มาจากอะเรย์เทลจากตำแหน่ง p จนถึงตำแหน่ง p + k โดยที่ TAIL[p+k] = # และ k > 0

ฟังก์ชัน StrCmp(x,y) เป็นฟังก์ชันที่ใช้เปรียบเทียบสตริง x กับสตริง y ว่าสตริงทั้ง 2 เหมือนกันหรือแตกต่างกัน ถ้าเหมือนกันจะส่งค่า -1 กลับ ถ้าไม่เหมือนกันจะส่งค่าจำนวนตัวอักษรในสตริง x และ y ที่เหมือนกันกลับมาให้

2. <u>ตัวอย่างแสดงขั้นตอนการสืบค้นคำ</u>

สมมติว่าต้องการทราบว่า คำ "กรรมกร" เก็บไว้ในพจนานุกรมอิเล็กทรอนิกส์ที่จัดเก็บไว้ในโครงสร้างข้อมูลแบบดับเบิลอะเรย์หรือไม่ รายละเอียดวิธีการสืบค้นแสดงตามตาราง 3.1

	อินพุท	การปฏิบัติ	เอาท์พุท
1	โหนดที่ 1 ตัว 'ก' ขนาดอะเรย์ 52	คำนวณหมายเลขโหนดถัดไป ตรวจสอบว่าหมายเลขโหนดถัดไปเกินขนาดอะเรย์หรือไม่ ปรากฏว่าไม่เกิน	โหนดถัดไป = 2
2	CHECK[2] = 1	ตรวจสอบว่าโหนดที่ได้เป็นโหนดที่สร้างมาจากโหนดที่ 1 จริง ซึ่งเป็นจริง	
3	BASE[2] = 1	ตรวจสอบว่าค่าของโหนดใหม่มีค่ากว่า 0 เพราะฉะนั้นสามารถสร้างมัลติโหนดโหนดถัดไปได้	
4	BASE[2] = 1 ตัว 'ร' ขนาดอะเรย์ 52	สร้างโหนดถัดไป จากโหนดที่ 2 แล้วตรวจสอบว่าหมายเลขของโหนดถัดไป ซึ่งคือ 37 น้อยกว่า 52 จริง	โหนดถัดไป = 37
5	BASE[37] = -9	ตรวจสอบค่าของโหนดใหม่ปรากฏว่าน้อยกว่า 0 แสดงว่าโหนดนี้เป็นเซพพาเรทโหนด	
6		ดึงคำศัพท์ส่วนที่เหลือจากอะเรย์เทล	รมกร#
7	Str1 = รมกร# Str2 = รมกร#	เปรียบเทียบคำศัพท์ที่ดึงจากอะเรย์เทลกับสตริงอินพุทที่เหลือ ปรากฏว่าเหมือนกัน นั่นคือ คำว่า กรรมกรอยู่ในพจนานุกรมอิเล็กทรอนิกส์	

ตารางที่ 3.1 ขั้นตอนการสืบค้นคำศัพท์

ขั้นตอนวิธีการปรับทันกาลของโครงสร้างข้อมูลแบบดับเบิลอะเรย์ (Algorithm of Updating the Double-Array)

1. ขั้นตอนวิธีการเพิ่มข้อมูลของโครงสร้างแบบดับเบิลอะเรย์

ขั้นตอนวิธีการเพิ่มข้อมูลของโครงสร้างแบบดับเบิลอะเรย์นั้น จะอธิบายวิธีการทำงาน และยกตัวอย่างประกอบการอธิบาย เพื่อให้เห็นขั้นตอนวิธีการเพิ่มข้อมูลจริงๆ ว่าเป็นเช่นไร การที่จะเพิ่มคำศัพท์ในพจนานุกรมอิเล็กทรอนิกส์จะต้องตรวจสอบว่ายังไม่มีคำศัพท์คำนั้นในพจนานุกรมหากมีอยู่แล้วจะไม่เพิ่มในพจนานุกรม เพราะฉะนั้นการเพิ่มคำศัพท์ในพจนานุกรมจะเริ่มจากการสืบค้นคำศัพท์เมื่อไม่พบคำศัพท์จึงเพิ่มคำศัพท์ และตามขั้นตอนการสืบค้นคำศัพท์ให้ผลว่าไม่พบคำศัพท์ 2 กรณีตามที่กล่าวไว้ในตอนต้นแล้ว ดังนั้นจะแทรกขั้นตอนการเพิ่มคำศัพท์ 2 แบบทันทีที่ขั้นตอนการสืบค้นให้ผลลัพธ์ว่าไม่พบคำศัพท์ ขั้นตอนการเพิ่มคำศัพท์ทั้ง 2 แบบมีรายละเอียดดังนี้

ฟังก์ชันการเพิ่มข้อมูลแบบที่ 1 AInsert(r,a_h a_{h+1} ... a_n a_{n+1})

เป็นขั้นตอนที่ใช้หลังจากทำการสืบค้นคำศัพท์แล้วไม่พบคำศัพท์ กรณีที่สร้างมัลติโหนดไม่สำเร็จ โดยที่พารามิเตอร์ r ใช้แทนหมายเลขโหนดปัจจุบัน ส่วนพารามิเตอร์ $a_h, a_{h+1} \ldots a_n a_{n+1}$ ใช้แทนส่วนของคำศัพท์ที่เหลือหลังจากการสร้างมัลติโหนดแล้ว ขั้นตอนการทำงานของฟังก์ชัน AInsert ประกอบด้วยขั้นตอนการสร้างมัลติโหนด s_t จากฟังก์ชัน $g(s_r, a_k) = s_t$ เพื่อเก็บชิงเกิลสตริง STR[s_t] ($a_{h+1} \ldots a_n a_{n+1}$) ไว้ในอะเรย์เทล ซึ่งลักษณะการเก็บข้อมูลของดับเบิลอะเรย์ และเทล แสดงตามรูป 3.5

ฟังก์ชันการเพิ่มข้อมูลแบบที่ 2

BInsert(r,a_{h+1} ... a_{h+k}, a_{h+k+1} ... a_n a_{n+1}, b_1 ... b_m)

เป็นขั้นตอนที่ใช้หลังจากทำการสืบค้นแล้วไม่พบคำศัพท์ กรณีที่เปรียบเทียบชิงเกิลสตริงกับคำศัพท์ส่วนที่เหลือหลังจากการสร้างมัลติโหนดแล้ว โดยที่พารามิเตอร์ $a_{h+1} a_{h+2} \ldots a_{h+k} b_1 \ldots b_m$ ($0 \leq k \leq n-k+1, 1 \leq m$) แทนชิงเกิลสตริง และ $a_{h+1} \ldots a_{h+k}$ a_{h+k+1} ... a_n a_{n+1} แทนคำศัพท์ส่วนที่เหลือ ส่วน $a_{h+1} \ldots a_{h+k}$ แทนสตริงที่เหมือนกันของชิงเกิลสตริงและคำศัพท์ส่วนที่เหลือ สำหรับพารามิเตอร์ที่ใช้ในฟังก์ชันนี้ประกอบด้วย

r แทนหมายเลขโหนดปัจจุบัน

$a_{h+k+1}...a_n a_{n+1}$ แทนคำศัพท์ส่วนที่เหลือหลังจากดึงส่วนที่ซ้ำกับสตริงที่ดึงมาจากเทลแล้ว

$b_1...b_n$ แทนส่วนของซิงเกิลสตริงส่วนที่เหลือหลังจากดึงส่วนที่ซ้ำกับคำศัพท์

ฟังก์ชัน BInsert เป็นฟังก์ชันที่สร้างมัลติโหนดจากสตริง $a_{h+1}...a_{h+k}$ และเก็บสตริง $a_{h+k+1}...a_n a_{n+1}$ กับ $b_1...b_n$ ในอะเรย์เทล ต่อไปจะกล่าวถึงขั้นตอนการทำงานของฟังก์ชัน AInsert BInsert และฟังก์ชันอื่นๆ ที่เป็นฟังก์ชันสนับสนุนการทำงานของฟังก์ชันทั้ง 2 พร้อมทั้งยกตัวอย่างประกอบด้วย

1.1 ฟังก์ชัน AInsert(s$_r$, $a_h a_{h+1}...a_n a_{n+1}$)

1. กำหนดค่าของหมายเลขโหนด t เท่ากับ BASE[r] + a_h
2. ตรวจสอบว่าค่าของ CHECK[t] = 0 หรือไม่
   เงื่อนไขเป็นจริง ทำงานในข้อ 4
   เท็จ ทำงานข้อ 3
3. เปลี่ยนค่าของ BASE[r] และ BASE[k] ซึ่ง
   k = CHECK[t] ตามขั้นต่อไปนี้
   3.1 ใช้ฟังก์ชัน SetList(r) เพื่อหาค่า RList
   SetList(k) เพื่อหาค่า KList
   3.2 ตรวจสอบจำนวนสมาชิกของ RList+1 ว่ามากกว่าจำนวนสมาชิกของ KList หรือไม่
   เงื่อนไขเป็นจริง ทำงานในข้อ 3.3
   เท็จ ทำงานในข้อ 3.5
   3.3 ใช้ฟังก์ชัน Modify(r,r,{a_h},RList) เพื่อหาค่า BASE[k] โดยที่อยู่ภายใต้เงื่อนไข
   CHECK[BASE[r]+b] = 0 และ
   b เป็นสมาชิกใน RList U {a_h}
   ทำงานข้อ 4

3.4 ใช้ฟังก์ชัน Modify(r,r, ,KList) เพื่อหาค่า
BASE[k] และ r โดยที่อยู่ภายใต้เงื่อนไข
CHECK[BASE[r]+b] ไม่เท่ากับ CHECK[BASE[r]+b]
โดยที่ b เป็นสมาชิกใน KList
ทำงานข้อ 4

4. ใช้ฟังก์ชัน InsStr(r,a_h,a_{h+1}...a_{n+1},POS)
เพื่อสร้างโหนดในดับเบิลอะเรย์จากสมการ $g(s_r,a_h)$
และจัดเก็บ a_{h+1}...a_{n+1} ในอะเรย์เทล

1.2 ฟังก์ชัน Modify(currents,a,ADD,ORG)

เป็นฟังก์ชันที่ใช้สำหรับค้นหาตรรชนีของอะเรย์เบสค่าใหม่ เนื่องจากมีการใช้อะเรย์เบสซ้ำ และจะทำการปรับค่าของอะเรย์เบสและเช็คใหม่ด้วย สำหรับขั้นตอนวิธีการมีดังนี้

1. กำหนดค่า oldbase = BASE[h] และ
ใช้ฟังก์ชัน XChechk(ADD U ORG) เพื่อหาค่า BASE[h]
2. ให้ทำงานซ้ำตั้งแต่ข้อ 3 ถึงข้อ 6 จนกระทั่ง c ใน ORG หมด
3. กำหนดค่า t = oldbase + t
กำหนดค่า t' = BASE[h] + c
BASE[t'] = BASE[t] และ CHECK[t'] = h
4. ตรวจสอบว่า BASE[t] มากกว่า 0 หรือไม่
เงื่อนไขเป็นจริง ทำงานข้อ 5
เท็จ ทำงานข้อ 6
5. ให้ค่าของ CHECK[q] เท่ากับ t' โดยที่ q เป็นค่าที่ได้มาจากการตรวจสอบเงื่อนไขจากสมการ
CHECK[BASE[t]+b] + t และ q = BASE[t] + b
และให้ค่าของ t' = currents ถ้า t = currents
6. ให้ค่า BASE[t] = 0 และ CHECK[t] = 0
7. ออกจากฟังก์ชัน และส่งค่า currents กลับ

1.3 ฟังก์ชัน InsStr(h,$e_1e_2...e_q$,dpos)

เป็นฟังก์ชันที่ใช้สำหรับเก็บคำศัพท์ในอะเรย์เทล ณ. ตำแหน่ง ที่ระบุและหากว่าตำแหน่งที่ระบุเป็นตำแหน่งสุดท้ายที่มีข้อมูลของอะเรย์เทลแล้ว จะส่งตำแหน่งสุดท้ายของอะเรย์เทลตำแหน่งใหม่กลับให้ฟังก์ชันที่เรียกฟังก์ชันนี้ สำหรับขั้นตอนวิธีการมีดังนี้

1. กำหนดหมายเลขโหนด t เท่ากับ BASE[h] + e_1
2. กำหนดค่า BASE[t] = -dpos และ
   CHECK[t] = h
3. ใช้ฟังก์ชัน StrTail(dpos,$e_2e_3...e_q$\$)
   เพื่อเก็บสตริง $e_2e_3...e_q$ ในอะเรย์เทล ซึ่งฟังก์ชันดังกล่าวจะส่งค่า POS ซึ่งเป็นค่าดรรชนีของอะเรย์เทลค่าสูงสุดที่เก็บข้อมูลหลังจากที่เพิ่มสตริงแล้ว

1.4 ฟังก์ชัน BInsert(r,$a_{h+1}...a_{h+k}$,$a_{h+k+1}...a_n a_{n+1}$,
$b_1...b_m$)

1. กำหนดค่า oldpos = -BASE[r]
2. สร้างเส้นทางเดินของโหนดโดยใช้ตัวอักษรจากสตริงอินพุท ตามขั้นตอนต่อไปนี้
   2.1 หาค่า BASE[r] จากฟังก์ชัน XCheck({a_{h+i}})
   โดยที่ $1 \leq i \leq k$
   2.2 คำนวณค่าโหนด r ใหม่ จาก BASE[r] + a_{h+i}
   ให้สร้างเส้นทางเดินของโหนดจนกระทั่ง i = k
3. หาค่า BASE[r] จากฟังก์ชัน XCheck({a_{h+k+1},b_1})
4. นำสตริงส่วนที่ได้จากอะเรย์เทลเดิมหลังจากที่ดึงสตริงที่เหมือนกันกับคำศัพท์ที่เหลือเก็บในเทล ณ. ตำแหน่งเดิม โดยใช้ฟังก์ชัน InsStr(r,$b_2...b_m$,oldpos)
5. เก็บคำศัพท์ส่วนที่เหลือจากการสร้างโหนดและซิงเกิลสตริง เก็บในอะเรย์เทล โดยใช้ฟังก์ชัน
   InsStr(r,$a_{h+k+2}...a_n a_{n+1}$,POS)

ในตอนเริ่มต้นการทำงานนั้นอะเรย์เบสและอะเรย์เช็คมีรายการในแต่ละอะเรย์เพียงรายการเดียวเท่านั้น คือ BASE[1] = 1 และ CHECK[1] = 1 เมื่อมีการเพิ่มคำศัพท์เข้าไปในโครงสร้างข้อมูลนี้โครงสร้างข้อมูลจะเพิ่มขนาดขึ้นเรื่อยๆ นอกจากฟังก์ชันหลักของการเพิ่มคำศัพท์ คือ ฟังก์ชันการสืบค้น ฟังก์ชัน AInsert และ ฟังก์ชัน BInsert แล้ว ยังมีฟังก์ชันที่ช่วยสนับสนุนการทำงานอีกหลายฟังก์ชันดังนี้

1.5 ฟังก์ชัน SetList(r)

เป็นฟังก์ชันที่ใช้สำหรับหาชุดของพยัญชนะ a_i ใด ๆ ที่สอดคล้องกับสมการ CHECK[BASE[r] + a_i] = r

1.6 ฟังก์ชัน N(LIST)

เป็นฟังก์ชันที่ใช้นับจำนวนรายการภายในเซต สำหรับฟังก์ชันที่จะกล่าวถึงต่อไปมีพารามิเตอร์ที่เกี่ยวข้องด้วยดังต่อไปนี้

RList , KList , LIST , ADD, ORG เป็นสัญญลักษณ์ที่ใช้แทนเซตย่อยของที่เกิดมาจาก I U {#}

POS เป็นตัวแปรชนิดโกลบอลที่เก็บค่าดรรชนีสูงสุดของอะเรย์เทลที่โดยที่ค่าเริ่มต้นของ POS คือ 1

1.7 ฟังก์ชัน XCheck(LIST)

เป็นฟังก์ชันที่หาค่า q ซึ่ง q เป็นค่าของดรรชนีของโหนดที่มีค่าน้อยที่สุดที่สร้างมาจากสมการ CHECK[q + c] = 0 โดยที่ q > 0 และ c เป็นตัวอักษรที่อยู่ในเซต LIST

1.8 ฟังก์ชัน StrTail(p,y)

เป็นฟังก์ชันที่ใช้สำหรับจัดเก็บสตริง y ในอะเรย์เทล ณ. ตำแหน่ง p เมื่อจัดเก็บสตริงแล้วจะส่งค่า POS กลับ โดยที่ค่า POS จะเพิ่มค่าความยาวของสตริงแล้ว ถ้าเป็นกรณีที่ p มีค่าเดียวกับ POS แต่ถ้ากรณีที่ p มีค่าน้อยกว่า POS แล้ว ค่าของ POS จะไม่เปลี่ยนแปลง

2. ขั้นตอนวิธีการลบข้อมูลของโครงสร้างแบบดับเบิลอะเรย์

ขั้นตอนวิธีการลบข้อมูลของโครงสร้างแบบดับเบิลอะเรย์ จะกระทำต่อเมื่อคำศัพท์ที่ต้องการลบปรากฏในโครงสร้าง เพราะฉะนั้นในขั้นแรกจะต้องนำเอาขั้นตอนการสืบค้นคำศัพท์มาใช้สืบค้นคำศัพท์เสียก่อน เมื่อพบแล้วจึงลบคำศัพท์นั้นออกไปการพัฒนาขั้นตอนนี้สามารถปรับปรุงจากขั้นตอนการสืบค้นในขณะที่ขั้นตอนดังกล่าวให้ผลลัพธ์เป็นจริงแล้วเพิ่มขั้นตอนการลบหลังจากนั้นทันที ซึ่งมีขั้นตอนวิธีการดังนี้

2.1 ฟังก์ชันการลบคำศัพท์

1. สืบค้นคำศัพท์โดยใช้ฟังก์ชันสืบค้นคำศัพท์
   สืบค้นคำศัพท์พบ ทำงานข้อ 3
   สืบค้นคำศัพท์ไม่พบ ลบคำศัพท์ไม่ได้ออกจากฟังก์ชันโดยส่งค่าเท็จ
2. เก็บ ? ในอะเรย์เทล ณ. ตำแหน่ง -BASE[r] ซึ่ง ค่าต่างๆ ได้มาจากฟังก์ชันการสืบค้นนั่นเอง
3. กำหนดค่า BASE[r] = 0 และ
   CHECK[r] = 0
4. ออกจากฟังก์ชันโดยส่งค่าจริงกลับ

ขั้นตอนการลบข้อมูลนั้นไม่ยุ่งยากสามารถกระทำโดยตัดความสัมพันธ์ของดับเบิลอะเรย์กับอะเรย์เทล โดยแทนค่าของอะเรย์ที่ตำแหน่งของเซพพาเรทโหนดด้วยศูนย์แล้วแทนที่ส่วนของซิงเกิลสตริงหรือ STR[r] ในอะเรย์เทลด้วยการ์เบจซิมโบล หรือสัญญลักษณ์ "?" นั่นเอง

สรุปแนวทางการออกแบบโครงสร้างข้อมูลสำหรับเก็บพจนานุกรมอิเล็กทรอนิกส์ภาษาไทย

การออกแบบโครงสร้างข้อมูลอิเล็กทรอนิกส์ภาษาไทยที่จะกล่าวในบทที่ 4 นั้น ได้นำแนวทางการจัดเก็บข้อมูล วิธีการปฏิบัติการกับข้อมูลแบบพลวัติและแบบสถิตมาใช้ในการออกแบบ นอกจากนั้นยังนำเอาลักษณะเด่นของคำไทยคือคำโดดมาใช้ ซึ่งรายละเอียดเนื้อหาต่าง ๆ จะอธิบายไว้ในบทที่ 4