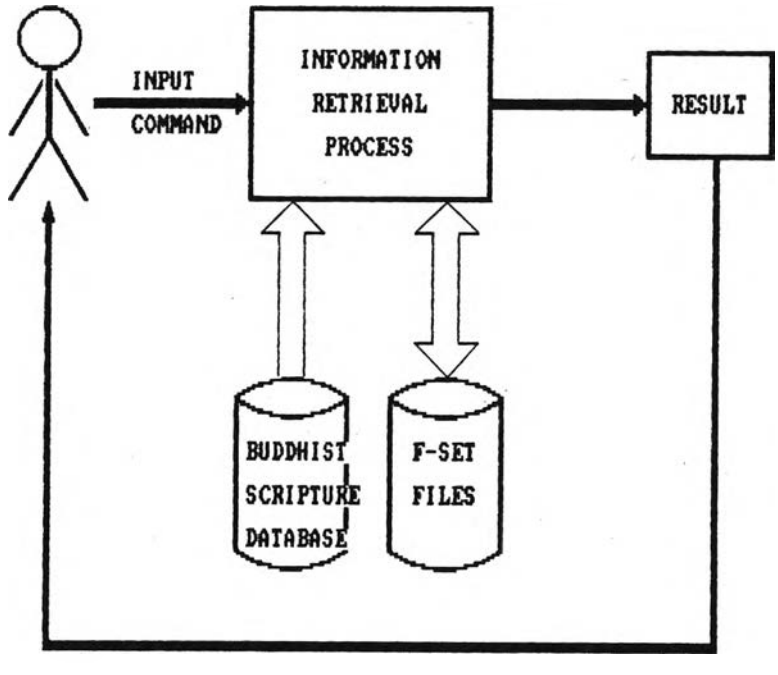
บทที่ 4

การออกแบบระบบสืบค้นสารสนเทศพระไตรปิฎกระบบใหม่

4.1 หลักการของระบบสืบค้นสารสนเทศพระไตรปิฎกระบบใหม่

ระบบสืบค้นสารสนเทศพระไตรปิฎกระบบใหม่ หรือ BUDSIR-III มีหลักการใหญ่ว่า ผู้ใช้เป็นผู้กำหนดทิศทางการค้น โดยผ่านทางคำสั่งค้นที่ป้อนเข้า และขบวนการสืบค้นสารสนเทศจะถูกควบคุมให้ดำเนินการค้นไปตามคำสั่งนั้น โดยมีการดึงข้อมูลต่าง ๆ จากฐานข้อมูลพระไตรปิฎก และกลุ่มแฟ้มข้อมูล F-Set มาใช้งาน เมื่อเสร็จการค้นแต่ละครั้งจะได้ผลลัพธ์แสดงให้แก่ผู้ใช้ทางจอภาพ และมีการเก็บบันทึกสารสนเทศผลลัพธ์เข้าแฟ้มข้อมูล F-Set ไว้ในรูปแบบพิเศษ ส่วนผู้ใช้เมื่อได้รับผลลัพธ์ตอบกลับมา ก็จะใช้ผลลัพธ์นั้น ประกอบการตัดสินใจในการป้อนคำสั่งถัดไป แผนผังข้างล่าง จำลองให้เห็นหลักการของระบบสืบค้นสารสนเทศนี้



รูปที่ 4.1 แผนผังแสดงหลักการของ BUDSIR-III

4.2 โครงสร้างฐานข้อมูล

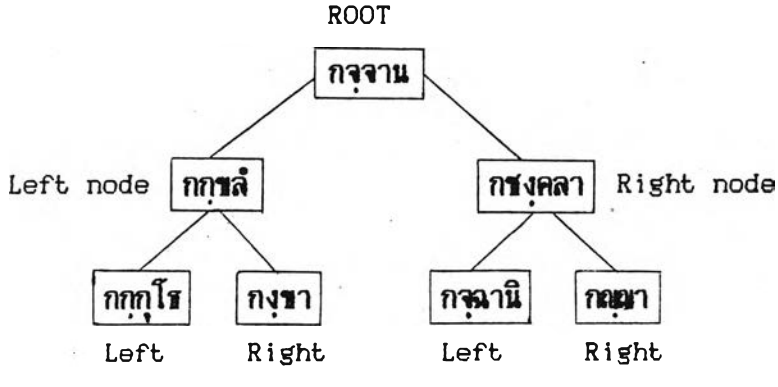
ฐานข้อมูลของ BUDSIR-III ประกอบด้วย ฐานข้อมูลพระไตรปิฎก และ กลุ่มแฟ้มข้อมูล F-Set สำหรับฐานข้อมูลแรก เป็นฐานข้อมูลเดียวกันกับที่ใช้ในบุดเซอร์ ส่วนแฟ้มข้อมูล F-Set ได้ออกแบบเพิ่มขึ้นมาในระบบใหม่นี้ เพื่อใช้เก็บสารสนเทศผลลัพธ์ที่ได้จากการค้นแต่ละครั้ง ทั้งสองส่วนมีรายละเอียด ดังต่อไปนี้

4.2.1 โครงสร้างฐานข้อมูลพระไตรปิฎก

ฐานข้อมูลพระไตรปิฎก ประกอบด้วยแฟ้มข้อมูล 3 ชนิด ได้แก่ แฟ้มพจนานุกรม แฟ้มข้อมูลย้อนกลับ และแฟ้มข้อมูลพระไตรปิฎก

แฟ้มพจนานุกรม

เก็บข้อมูล**คำศัพท์** พร้อมทั้งรายละเอียดเกี่ยวกับคำศัพท์นั้น ข้อมูลเหล่านี้จะถูกแยกเก็บเป็นกลุ่ม ๆ ตามพยัญชนะเริ่มต้นของคำศัพท์ เนื่องจากในภาษาบาลีมีพยัญชนะ 33 ตัว จึงเก็บได้เป็น 33 กลุ่ม การจัดเก็บคำศัพท์ภายในกลุ่ม จะอยู่ในรูปของ **โครงสร้างแบบทรี** โดยมีตัวชี้โยงระหว่างกัน ซึ่งจะจำลองให้เห็นได้ดังภาพข้างล่างนี้ นอกจากนั้น ยังมี**แฟ้มดิกแพค** (Dict-Pack) ที่เก็บเฉพาะรายการคำศัพท์แยกมาอีกต่างหาก



รูปที่ 4.2 แสดงตัวอย่างการเชื่อมโยงของคำศัพท์ในโครงสร้างแบบทรี

แฟ้มข้อมูลย้อนกลับ

เก็บข้อมูลจำนวนการปรากฏในแต่ละเล่มของคำศัพท์แต่ละคำ รวมทั้งเก็บข้อมูล **บัญชีการปรากฏ** ได้แก่ ข้อมูลตำแหน่งทุกตำแหน่งของคำศัพท์แต่ละคำ ที่ปรากฏอยู่ในพระไตรปิฎก ข้อมูลตำแหน่ง เป็นรหัสระบุ หน้า/บรรทัด/คำที่ และ แฟล็ก ซึ่งใช้บ่งชี้ว่าเป็นคำสุดท้ายของบรรทัด หรือคำสุดท้ายของหน้า ในขบวนการสืบค้นสารสนเทศ โดยเฉพาะการตรวจสอบคำประชิด จำเป็นต้องใช้ข้อมูลส่วนนี้โดยตรง

แฟ้มข้อมูลพระไตรปิฎก

เก็บข้อมูลพระไตรปิฎกทั้ง 45 เล่ม โดยแยกเป็นเล่ม ๆ ไป ข้อความในพระไตรปิฎกจะถูกจัดเก็บไว้เป็นบล็อก และจัดระเบียนไว้ให้พร้อมที่จะดึงออกไปแสดงทางจอภาพ หรือพิมพ์ออกทางเครื่องพิมพ์ นอกจากนั้น ยังมีไดเรกทอรีประจำแฟ้มข้อมูลพระไตรปิฎก ซึ่งเก็บตัวเลขระบุตำแหน่งเริ่มต้นของข้อความแต่ละหน้าเอาไว้ ดังนั้น เมื่อต้องการดึงข้อความหน้าใด ๆ จากแฟ้มข้อมูลพระไตรปิฎกแล้ว ก็ต้องมาค้นหาตำแหน่งเริ่มต้นของหน้านั้น ๆ ในไดเรกทอรีนี้เสียก่อน

แฟ้มข้อมูลทั้ง 3 ประเภทจะมีตัวชี้เชื่อมโยงระหว่างกัน โดยเริ่มจากแต่ละระเบียนที่เก็บคำศัพท์แต่ละคำในแฟ้มพจนานุกรม จะมีตัวชี้บอกระเบียนเริ่มต้นที่เก็บข้อมูลจำนวนปรากฏและตำแหน่งของคำนั้น ๆ ในแฟ้มข้อมูลย้อนกลับ จากข้อมูลตำแหน่งของคำซึ่งระบุถึง เลขหน้า บรรทัดที่ และคำที่ จะสามารถนำไปค้นหาในไดเรกทอรีประจำแฟ้มข้อมูลพระไตรปิฎกได้ว่า ระเบียนเริ่มต้นของข้อความหน้านั้น อยู่ที่ใดในแฟ้มพระไตรปิฎก จากนั้น ก็สามารถจะเข้าถึงข้อความในพระไตรปิฎกหน้านั้น ๆ ได้ทุกบรรทัด ทุกคำ ข้อความพระไตรปิฎกแต่ละหน้าจะตั้งต้นที่ระเบียนเริ่มต้นของหน้านั้น ไล่ไปตามลำดับจนกระทั่งพบเครื่องหมายจบหน้า

แผนผังในรูปที่ 2.3 ของบทที่ 2 แสดงให้เห็นโครงสร้างทั้งหมดของฐานข้อมูลพระไตรปิฎก และการเชื่อมโยงระหว่างแฟ้มข้อมูลทั้ง 3 ประเภท ดังกล่าว

4.2.2 โครงสร้างกลุ่มแฟ้มข้อมูล F-Set

กลุ่มแฟ้มข้อมูล F-Set ประกอบด้วยแฟ้มไดเรกทอรีของ F-Set 1 แฟ้ม และแฟ้ม F-Set ตั้งแต่ 1 แฟ้มขึ้นไป

ที่มาและความหมายของแฟ้มข้อมูล F-Set

จากหลักการค้นของระบบ ต้องเก็บผลลัพธ์การค้นไว้ทุกครั้ง ทั้งนี้เพื่อให้สามารถขยายการค้นออกไปได้กว้างขวางขึ้น เช่น ให้ผู้ใช้สามารถทำการค้นสารสนเทศย่อยลงไป ในกลุ่ม

1174136811

สารสนเทศผลลัพธ์ขั้นต้น ซึ่งจะทำให้ได้ผลลัพธ์ที่ตรงเป้าหมายมากยิ่งขึ้น หรือให้ผู้ใช้สามารถรวมเอาผลลัพธ์ของการค้นหลาย ๆ ครั้งเข้าเป็นสารสนเทศกลุ่มเดียว และตัดส่วนที่ซ้ำกันออกไปด้วย เป็นต้น เพียงความต้องการเท่านี้ ก็จะเห็นถึงความจำเป็นแล้วว่า ต้องจัดให้มีแฟ้มชั่วคราวที่จะเก็บสารสนเทศผลลัพธ์ของการค้นแต่ละครั้งเอาไว้ และได้ให้ชื่อเรียกแฟ้มประเภทนี้ว่า F-Set

<u>แนวทางการออกแบบแฟ้ม F-Set</u>

สิ่งที่ต้องคำนึงถึงในการออกแบบแฟ้ม F-Set มีหลายประการด้วยกัน ดังนี้

1. ขั้นแรกต้องกำหนดขอบเขตของสารสนเทศเป็นส่วน ๆ โดยที่แต่ละส่วน เมื่อถูกสืบค้นได้เป็นสารสนเทศผลลัพธ์แล้ว จะให้ความหมายที่สอดคล้องกับความต้องการค้นของผู้ใช้ครั้งนั้น ๆ
2. ข้อมูลในแฟ้ม F-Set ต้องสามารถอ้างอิงถึงกลุ่มสารสนเทศที่เป็นผลลัพธ์ของการค้นนั้น ๆ ได้
3. ข้อมูลในแฟ้ม F-Set ต้องสามารถนำไปใช้ดำเนินปฏิบัติการแบบบูล และปฏิบัติการตรวจคำประชิด ในขบวนการสืบค้นสารสนเทศได้
4. ไม่ควรเก็บผลลัพธ์เป็นข้อความของพระไตรปิฎกโดยตรง เพราะจะซ้ำซ้อนกับที่มีเก็บในฐานข้อมูลพระไตรปิฎกแล้ว
5. ใช้เนื้อที่การเก็บข้อมูลของแฟ้ม F-Set ให้สิ้นเปลืองน้อยที่สุด

<u>การออกแบบโครงสร้างข้อมูลในแฟ้ม F-Set</u>

เนื่องจากเนื้อความในพระไตรปิฎกแต่ละเล่มจะถูกแบ่งไว้เป็นข้อ ๆ แต่ละข้อมีหมายเลขข้อนำหน้าข้อความ เลขข้อจะเริ่มตั้งแต่ (1) เรียงไปตามลำดับ แต่เดิม การค้นหาข้อความที่ต้องการ และการอ้างถึงข้อความตอนใดในพระไตรปิฎกฉบับพิมพ์นั้น มักมีการใช้หมายเลขข้อเหล่านี้เป็นตัวอ้างอิง ซึ่งจะเป็นที่เข้าใจตรงกันในหมู่ผู้ศึกษาพระไตรปิฎก

ดังนั้น ข้อมูลใน F-Set ควรจะอ้างอิงถึงสารสนเทศผลลัพธ์เป็นข้อ ๆ ไป และเพื่อไม่ให้เก็บสารสนเทศผลลัพธ์แต่ละข้อไปซ้ำซ้อนกับในฐานข้อมูลพระไตรปิฎก จึงออกแบบให้เก็บกลุ่มหมายเลขข้อของสารสนเทศผลลัพธ์ไว้แทน ซึ่งนับว่ามีความเหมาะสม เนื่องจากหมายเลขข้อเหล่านี้ ไม่มีการซ้ำซ้อนกัน (unique) สามารถบ่งชี้ว่าเป็นสารสนเทศข้อใดได้โดยไม่สับสน การเก็บข้อมูลเป็นหมายเลขข้อนี้ เป็นลักษณะของแฟ้มข้อมูลย้อนกลับ จึงทำให้สามารถดำเนินปฏิบัติการแบบบูลบนแฟ้ม F-Set ได้ และเพื่อให้สามารถดำเนินปฏิบัติการตรวจคำประชิด บนแฟ้ม F-Set ได้ ก็จำเป็นต้องเก็บข้อมูลตำแหน่งคำลงไปด้วย และจากหลักการที่ว่า F-Set เป็นข้อมูลผลลัพธ์ที่ได้จากการสืบค้นสารสนเทศแต่ละครั้ง ในการค้นจะต้องมีการกำหนดคำค้นต่าง ๆ ไว้ในคำสั่งค้น ดังนั้น แทนที่จะเก็บเพียงหมายเลขข้อของสารสนเทศที่ปรากฏคำค้นเหล่านั้นไว้ จึงได้เก็บตำแหน่งคำของคำค้นเหล่านั้นลงไปด้วย ซึ่งจากฐานข้อมูลพระไตรปิฎก จะสามารถดึงข้อมูลตำแหน่งคำของ

ทุกคำศัพท์เป็น เลขหน้า/เลขบรรทัด/เลขที่คำในบรรทัด มาได้ และนำมาเก็บเพิ่มในแฟ้ม F-Set ดังนั้น ข้อมูลในแฟ้ม F-Set จะเก็บรวบรวมข้อมูล เลขข้อ/เลขหน้า/เลขบรรทัด/เลขที่คำ ซึ่งจะสามารถอ้างอิงสารสนเทศผลลัพธ์ทุกข้อและยังบอกละเอียดไปถึงตำแหน่งของคำด้นด้วย การเก็บเป็นข้อมูลตัวเลขนี้จะทำให้สิ้นเปลืองเนื้อที่เก็บน้อยกว่า เมื่อเทียบกับการเก็บตัวสารสนเทศตรง ๆ

การเรียกชื่อแฟ้ม F-Set จะมีการกำหนดหมายเลขให้แก่แฟ้ม F-Set ที่ถูกสร้างขึ้นใหม่ในทันที เรียกว่า **หมายเลขเซท** หมายเลขนี้จะยูนิค ดังนั้น จะในการอ้างถึงแฟ้ม F-Set แฟ้มใดก็ให้ใช้หมายเลขเซทนี้ โดยเติมเครื่องหมาย "#" ไว้ข้างหน้าด้วยทุกครั้ง เช่น #1, #2 เป็นต้น ทั้งนี้เพื่อให้เห็นว่า มีความหมายแตกต่างจากตัวเลขธรรมดา คือเป็นหมายเลขเซท ในการกำหนดหมายเลขเซทให้แก่ F-Set แฟ้มใหม่นั้น จะเริ่มตั้งแต่หมายเลข 1 เป็นต้นไป และได้ใช้เนื้อที่ในหน่วยความจำสำหรับเก็บค่าตัวเลขล่าสุด ที่ถูกกำหนดให้แก่ F-Set ไปแล้ว

ในรูปที่ 4.3 ได้แสดงให้เห็นถึงโครงสร้างของกลุ่มแฟ้มข้อมูล F-Set

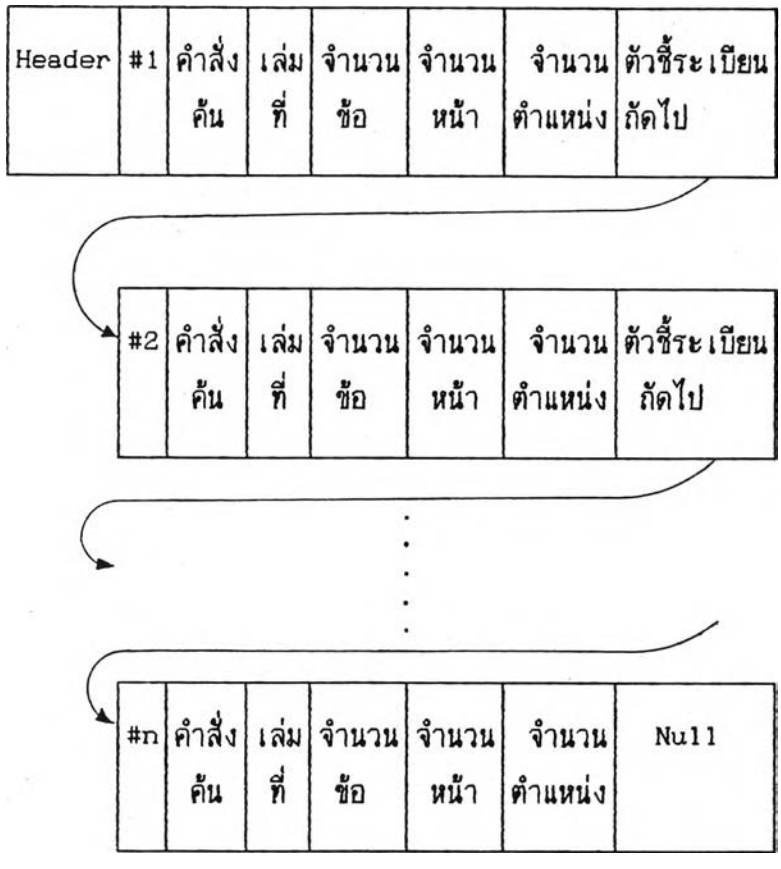
<u>แฟ้มไดเรกทอรีของ F-Set</u>

เนื่องจากแฟ้ม F-Set แต่ละแฟ้ม เกิดขึ้นจากการค้นสารสนเทศแต่ละครั้ง โดยมีคำสั่งค้นต่าง ๆ กัน ซึ่งโดยทั่วไปแล้ว ผู้ใช้ย่อมต้องการทราบถึงที่มาของสารสนเทศผลลัพธ์เหล่านั้นที่ถูกเก็บไว้ในแฟ้ม F-Set ต่าง ๆ ในรูปแบบพิเศษ และนับเป็นเรื่องที่ยุ่งยากมากที่จะให้ผู้ใช้จดจำไว้เอง จึงจัดให้มีแฟ้มไดเรกทอรีของ F-Set สำหรับเก็บคำสั่งค้นแต่ละครั้ง รวมทั้งปริมาณของสารสนเทศผลลัพธ์แต่ละครั้งว่า มีกี่ข้อ กี่หน้า และจำนวนคำค้นมีพบอยู่ทั้งหมดกี่ตำแหน่ง ข้อมูลแต่ละระเบียนในแฟ้มไดเรกทอรี จะเก็บข้อมูลเกี่ยวกับแฟ้ม F-Set แต่ละแฟ้ม ระเบียนข้อมูลในแฟ้มไดเรกทอรีจะโยงต่อกันเป็นลักษณะของ **รายการเชื่อมโยง** (Linked List) แบบทิศทางเดียว และได้ใช้เนื้อที่ในหน่วยความจำสำหรับเก็บตัวชี้ระเบียนแรก และตัวชี้ระเบียนสุดท้าย ไว้ด้วย

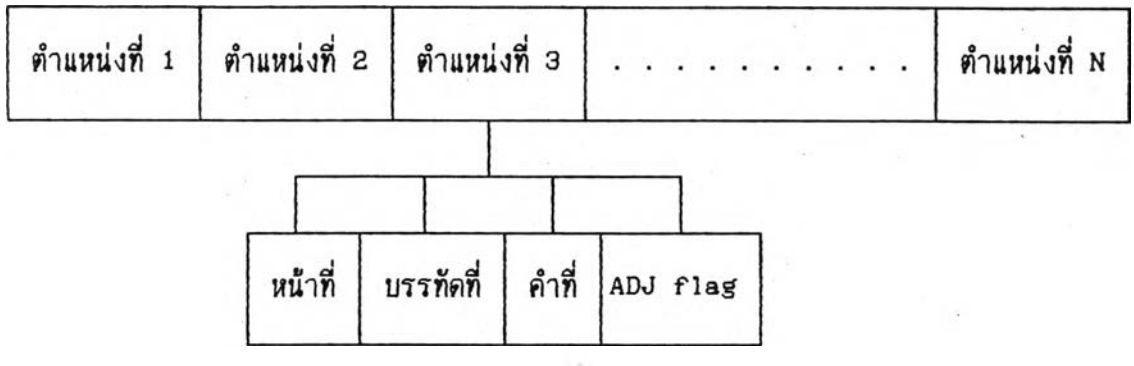
<u>การเพิ่ม-ลบ F-Set ในฐานข้อมูล</u>

ในการเพิ่มแฟ้ม F-Set แฟ้มใหม่เข้าในฐานข้อมูล จะมีการเพิ่มค่าตัวเลขนับหมายเลขเซทขึ้น 1 แล้วใช้ค่านั้น กำหนดหมายเลขเซทให้แก่ F-Set แฟ้มใหม่ พร้อมกับบันทึกข้อมูลหมายเลขเซท คำสั่งค้น จำนวนข้อ จำนวนหน้า จำนวนตำแหน่งคำ ของสารสนเทศผลลัพธ์นั้นเข้าเป็นระเบียนใหม่ในแฟ้มไดเรกทอรีของ F-Set ด้วย โดยจะเพิ่มเข้าทางตอนท้ายของ **รายการเชื่อมโยง** (Linked List) เท่านั้น เนื่องจากมีตัวชี้บอกระเบียนสุดท้ายไว้ จึงสามารถเก็บลงในระเบียนถัดไปได้ และในการนี้ จะต้องมีการเปลี่ยนค่าตัวชี้ระเบียนสุดท้าย ให้ชี้มายังระเบียนใหม่ และกำหนดตัวชี้ของระเบียนก่อนหน้า ซึ่งเดิมมีค่าเป็น Null อยู่ ให้เก็บตำแหน่งของระเบียนใหม่นี้ไว้แทน แล้วกำหนดให้ตัวชี้ของระเบียนใหม่เป็น Null ดังภาพจำลองในรูปที่ 4.4

แฟ้มไดเรกทอรีของ F-Set

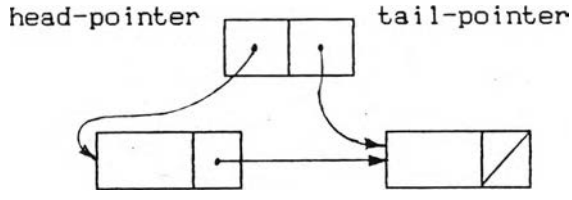


แฟ้ม F-Set

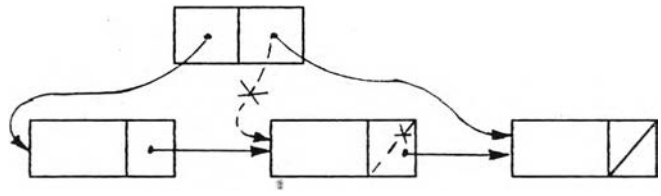


รูปที่ 4.3 แสดงโครงสร้างของกลุ่มแฟ้มข้อมูล F-Set และข้อมูลที่เก็บ

ลักษณะแฟ้มไดเรกทอรีของ F-Set ที่มีข้อมูลอยู่ 2 ระเบียน



การเพิ่มระเบียนใหม่เข้าในแฟ้มไดเรกทอรีของ F-Set



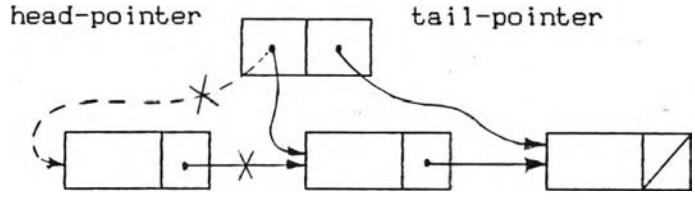
รูปที่ 4.4 แสดงการเพิ่มระเบียนเข้าไดเรกทอรี เมื่อมี F-Set แฟ้มใหม่เพิ่มเข้าในฐานข้อมูล

ส่วนการลบแฟ้ม F-Set แฟ้มใดแฟ้มหนึ่งออกจากฐานข้อมูล ก็ต้องลบระเบียนข้อมูลในแฟ้มไดเรกทอรี ที่เก็บข้อมูลเกี่ยวกับ F-Set นั้น ๆ ออกไป 1 ระเบียน โดยการเปลี่ยนตัวชี้ของระเบียนก่อนหน้า หรืออาจเป็นตัวชี้ระเบียนแรกของ รายการเชื่อมโยง นั้น (ถ้าระเบียนที่จะลบออกเป็นระเบียนแรก) ให้ชี้ไปยังระเบียนถัดไปแทน แต่ถ้าระเบียนที่จะลบออกเป็นระเบียนสุดท้ายของ Link-List ก็จะต้องกำหนดตัวชี้ของระเบียนก่อนหน้าให้เป็น Null แทน ดังในรูปที่ 4.5 เป็นภาพจำลองของการลบข้อมูลในไดเรกทอรีของ F-Set ออก 1 ระเบียน เมื่อมีการลบแฟ้ม F-Set ออกจากฐานข้อมูล 1 แฟ้ม

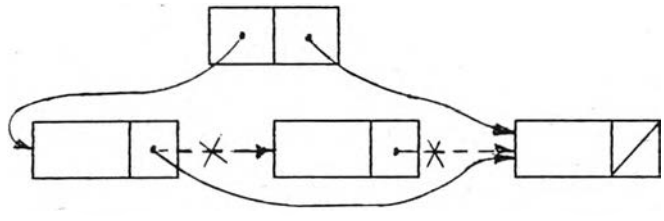
ข้อจำกัดเกี่ยวกับ F-Set

จำนวนแฟ้ม F-Set ที่จะเก็บบันทึกไว้ได้ ถูกจำกัดด้วยปริมาณเนื้อที่ว่างในจานแม่เหล็ก ว่ามีสำรองไว้มากเพียงใด

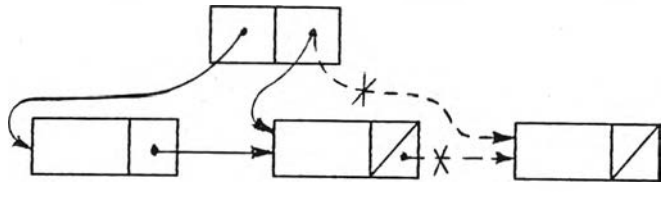
การลบข้อมูลระเบียนแรกออกจากแฟ้มไดเรกทอรีของ F-Set



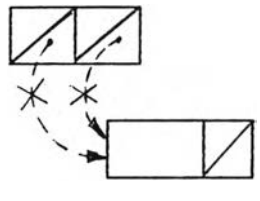
การลบข้อมูลระเบียนกลาง ๆ ออกจากแฟ้มไดเรกทอรีของ F-Set



การลบข้อมูลระเบียนท้ายออกจากแฟ้มไดเรกทอรีของ F-Set



การลบข้อมูลที่เหลือเพียงระเบียนเดียวออกจากแฟ้มไดเรกทอรีของ F-Set



รูปที่ 4.5 แสดงการลบระเบียนของไดเรกทอรี เมื่อ F-Set ถูกลบออกจากฐานข้อมูล 1 แฟ้ม

4.3 โครงสร้างขบวนการสืบค้นสารสนเทศพระไตรปิฎกในระบบใหม่

4.3.1 ปฏิบัติการแบบบูล

ความหมายของปฏิบัติการแบบบูล AND, OR, NOT เมื่อนำมาใช้ในทางปฏิบัติ จะใช้ปฏิบัติการของเซท ได้แก่ อินเทอเซกชั่น ยูเนียน และ ดิฟเฟอเรนซ์ ตามลำดับ โดยมอง F-Set เป็นเซท ๆ หนึ่ง ที่มีสมาชิกเป็นหมายเลขข้อต่าง ๆ และในการกำหนดปฏิบัติการแบบบูลไว้ในคำสั่งค้นนั้น จะใช้สัญญลักษณ์แทน ดังนี้

ชนิดของปฏิบัติการ	สัญญลักษณ์ที่ใช้
AND	&
OR	+
NOT	-

ดังนั้น เมื่อต้องการใช้ปฏิบัติการแบบบูลกระทำบน F-Set 2 เซท จะระบุในคำสั่งได้ ดังตัวอย่าง

#1 & #2 หมายถึง F-Set หมายเลข 1 **AND** F-Set หมายเลข 2

#1 + #2 หมายถึง F-Set หมายเลข 1 **OR** F-Set หมายเลข 2

#1 - #2 หมายถึง F-Set หมายเลข 1 **NOT** F-Set หมายเลข 2

ความหมายของ #1 & #2 ในทางปฏิบัติ

หมายความว่า F-Set หมายเลข 1 อินเทอเซกกับ F-Set หมายเลข 2 โดยมองว่า F-Set ทั้งสองเป็นเซทของหมายเลขข้อของสารสนเทศผลลัพธ์ ที่ได้จากการสืบค้นสารสนเทศครั้งที่ 1 และครั้งที่ 2 ตามลำดับ

ดังนั้น **#1 & #2** จะได้ผลเป็นหมายเลขข้อกลุ่มหนึ่ง ที่พบว่าเป็นสมาชิกอยู่ในทั้งสองเซท และกลุ่มของหมายเลขข้อนี้ รวมทั้งข้อมูลตำแหน่งคำที่อยู่ภายในข้อเหล่านี้ จะถูกเก็บบันทึกไว้ใน F-Set แฟ้มใหม่ รูปที่ 4.6 แสดงผังงานการดำเนินปฏิบัติการ AND ระหว่างเซท #21 และเซท #22 ผลลัพธ์(ถ้ามี) จะถูกเก็บไว้ในเซท #23

ความหมายของ #1 + #2 ในทางปฏิบัติ

หมายความว่า F-Set หมายเลข 1 ยูเนียนกับ F-Set หมายเลข 2 โดยที่ F-Set ทั้งสองเป็นเซทของหมายเลขข้อของสารสนเทศผลลัพธ์ ที่ได้จากการสืบค้นสารสนเทศครั้งที่ 1 และครั้งที่ 2 ตามลำดับ

ดังนั้น #1 + #2 จะได้ผลเป็นหมายเลขข้อกลุ่มหนึ่ง ที่พบว่าเป็นสมาชิกอยู่ในเซท #1 หรือ #2 หรืออาจเป็นสมาชิกอยู่ในทั้งสองเซทก็ได้ และกลุ่มของหมายเลขข้อนี้ รวมทั้งข้อมูลตำแหน่งคำที่อยู่ภายในข้อเหล่านี้ จะถูกเก็บบันทึกไว้ใน F-Set แฟ้มใหม่ รูปที่ 4.7 แสดงผังงานการดำเนินปฏิบัติการ OR ระหว่างเซท #21 และเซท #22 ผลลัพธ์จะถูกเก็บไว้ในเซท #23

ความหมายของ #1 - #2 ในทางปฏิบัติ

หมายความว่า F-Set หมายเลข 1 ทำเซทดิฟเฟอเรนซ์ด้วย F-Set หมายเลข 2 โดยที่ F-Set ทั้งสองเป็นเซทของหมายเลขข้อของสารสนเทศผลลัพธ์ ที่ได้จากการสืบค้นสารสนเทศครั้งที่ 1 และครั้งที่ 2 ตามลำดับ

ดังนั้น #1 - #2 จะได้ผลเป็นหมายเลขข้อกลุ่มหนึ่ง ที่พบว่าเป็นสมาชิกอยู่ในเซท #1 แต่ไม่ได้เป็นสมาชิกอยู่ในเซท #2 และกลุ่มของหมายเลขข้อนี้ รวมทั้งข้อมูลตำแหน่งคำที่อยู่ภายในข้อเหล่านี้ จะถูกเก็บบันทึกไว้ใน F-Set แฟ้มใหม่ รูปที่ 4.8 แสดงผังงานการดำเนินปฏิบัติการ NOT ระหว่างเซท #21 และเซท #22 ผลลัพธ์(ถ้ามี) จะถูกเก็บไว้ในเซท #23

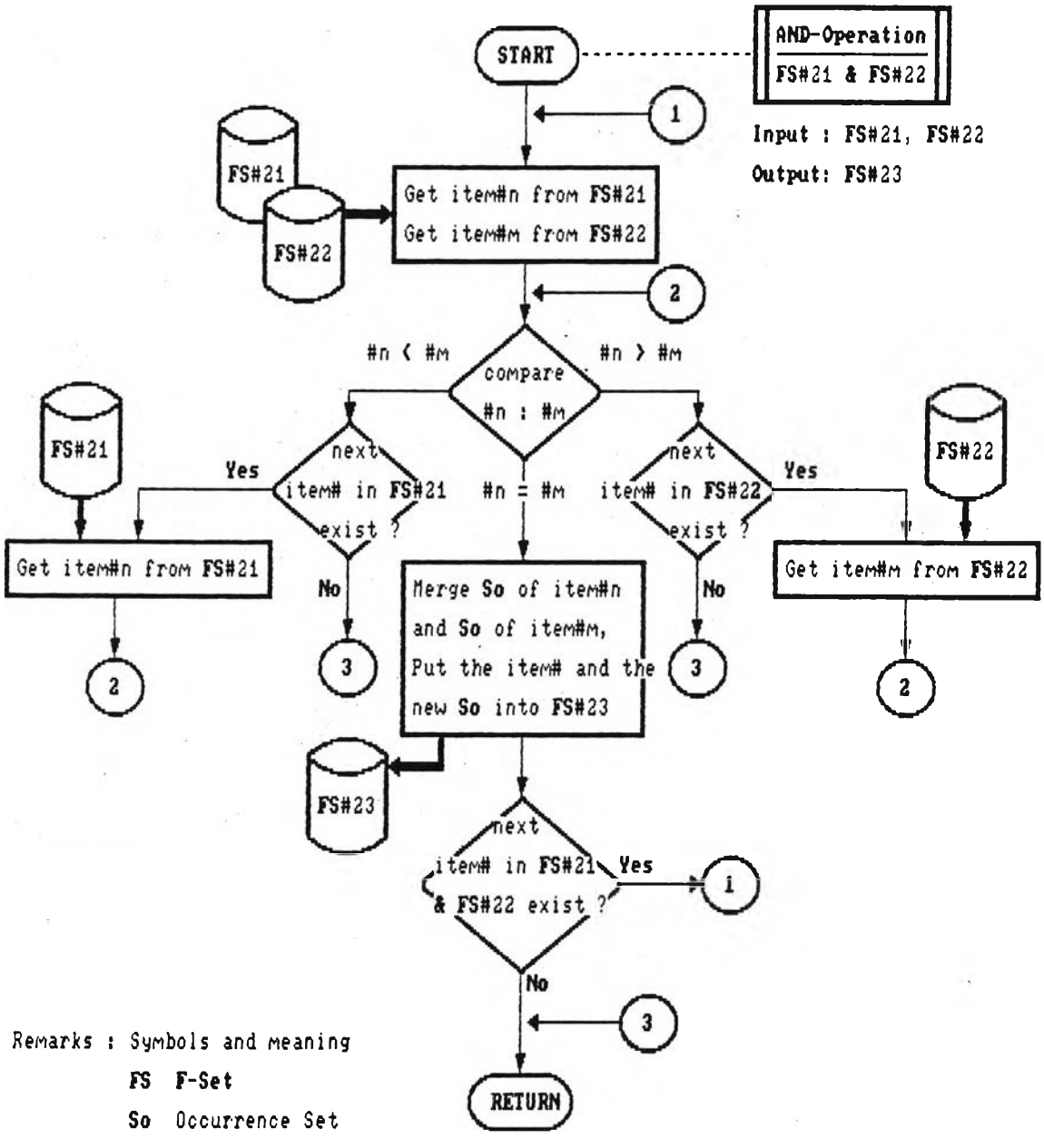
4.3.2 ปฏิบัติการประชิด

ความหมายของปฏิบัติการประชิด หรือ ADJ มีความหมายดังทราบกันแล้วว่า จะตรวจหาตำแหน่งของคำ 2 คำ ว่ามีที่ปรากฏอยู่ในตำแหน่งถัดกันไปบ้างหรือไม่ เมื่อนำมาใช้กระทำบน F-Set ในขบวนการสืบค้นสารสนเทศของ BUDSIR-III นี้ จะดำเนินการโดย มอง F-Set เป็นกลุ่มของตำแหน่งคำ ซึ่งจะระบุตำแหน่งแต่ละตำแหน่งเป็น เลขข้อ/เลขหน้า/เลขบรรทัด/เลขที่คำในบรรทัด ดังนั้น จึงสามารถนำข้อมูลตำแหน่งคำนี้ มาใช้พิจารณาหาคำประชิด ในระหว่าง F-Set 2 เซทได้ โดยกำหนดให้ใช้สัญญลักษณ์ "@" สำหรับแทนปฏิบัติการประชิด เมื่อต้องการระบุไว้ในคำสั่งค้น ตัวอย่างเช่น

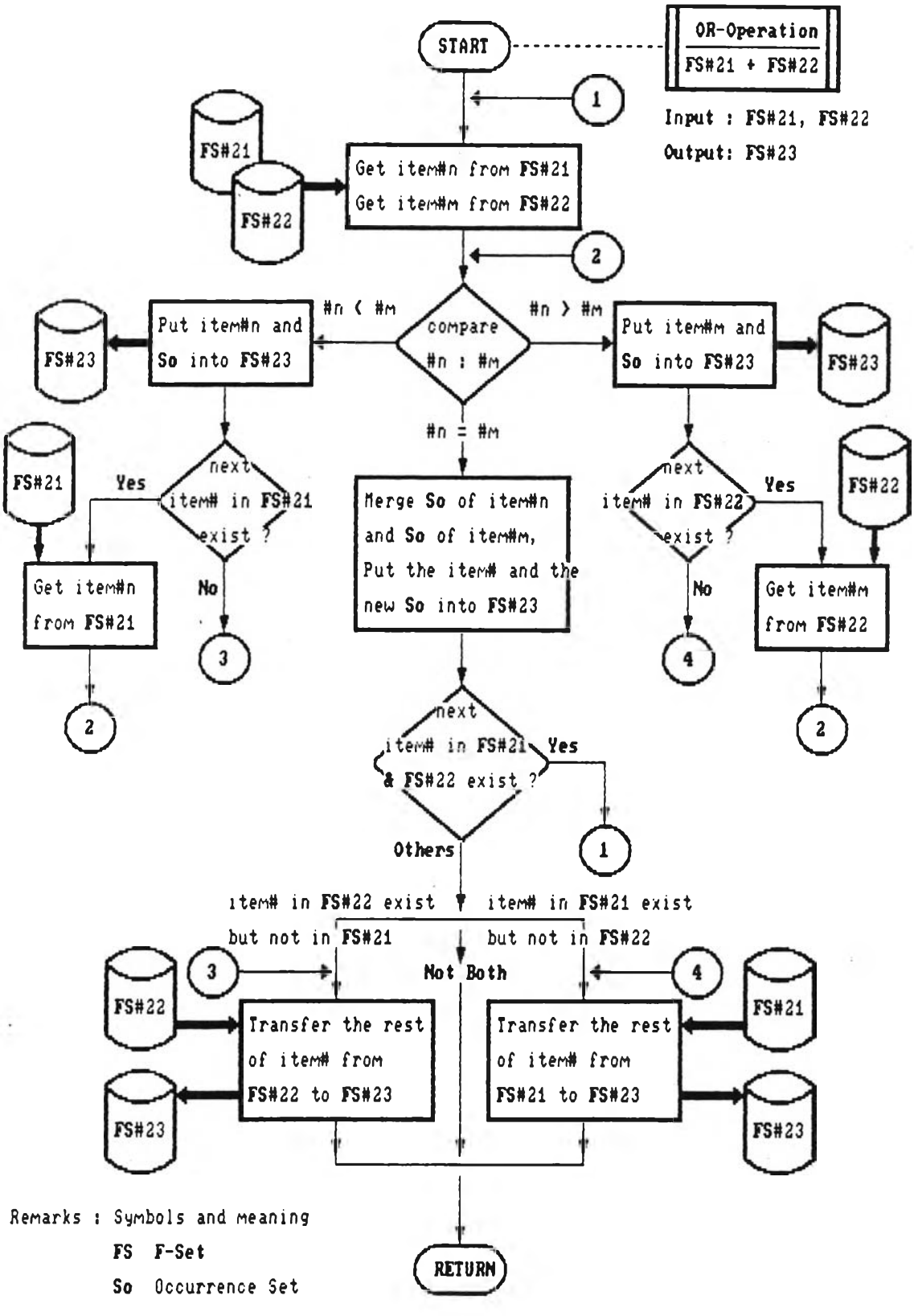
#1 @ #2 หมายถึง F-Set หมายเลข 1 กระทำ ADJ กับ F-Set หมายเลข 2

ความหมายของ #1 @ #2 ในทางปฏิบัติ

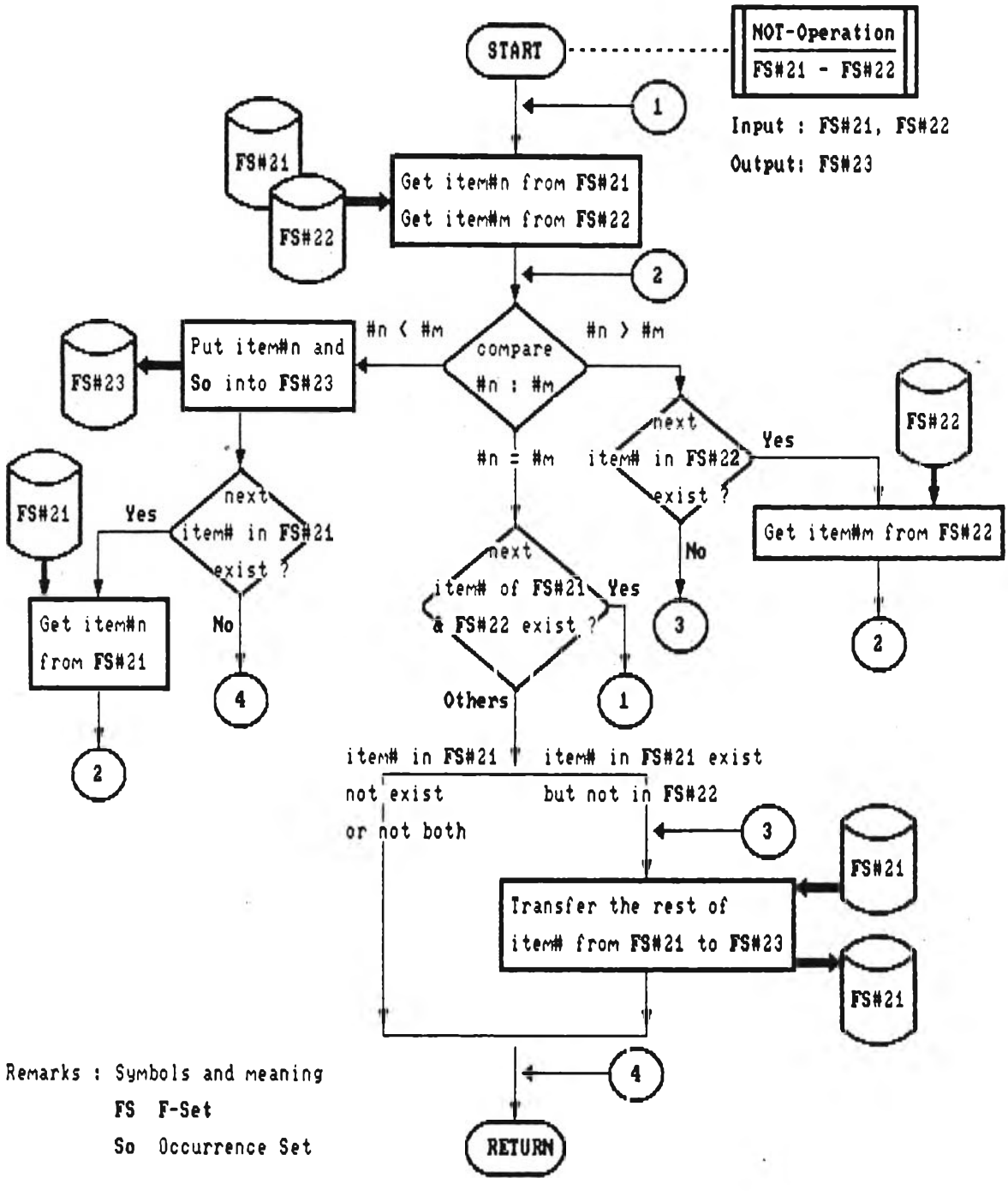
เมื่อมอง F-Set เป็นเซทของข้อมูลตำแหน่งคำ ที่ได้มาจากการสืบค้นสารสนเทศขั้นต้น ซึ่งข้อมูลตำแหน่งเหล่านี้เป็นบอกถึงตำแหน่งของคำต่างที่ใช้ระบุไว้ในคำสั่งค้น ในการดำเนินปฏิบัติการ ADJ ระบบจะไม่จำเป็นต้องรู้ว่า ใน F-Set ทั้งสองเก็บตำแหน่งของคำใดบ้าง แต่จะพิจารณาแต่ละตำแหน่งของทั้งสองเซทว่า ตำแหน่งเหล่านั้น อยู่ใน ข้อ/หน้าใด บรรทัดใด และคำที่เท่าไรของบรรทัด



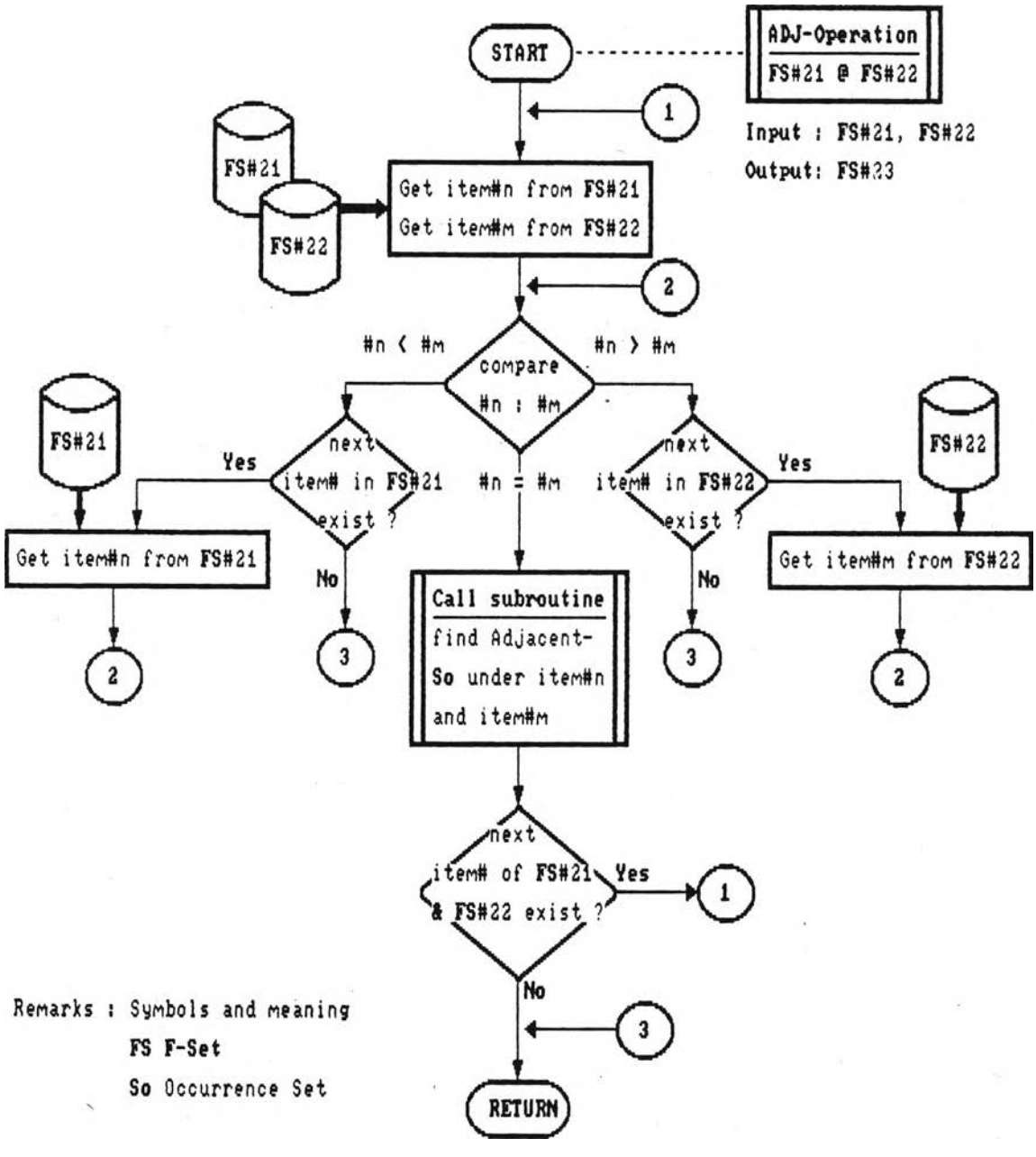
รูปที่ 4.6 แสดงผังงานการดำเนินปฏิบัติการ AND ระหว่างเซท #21 และเซท #22



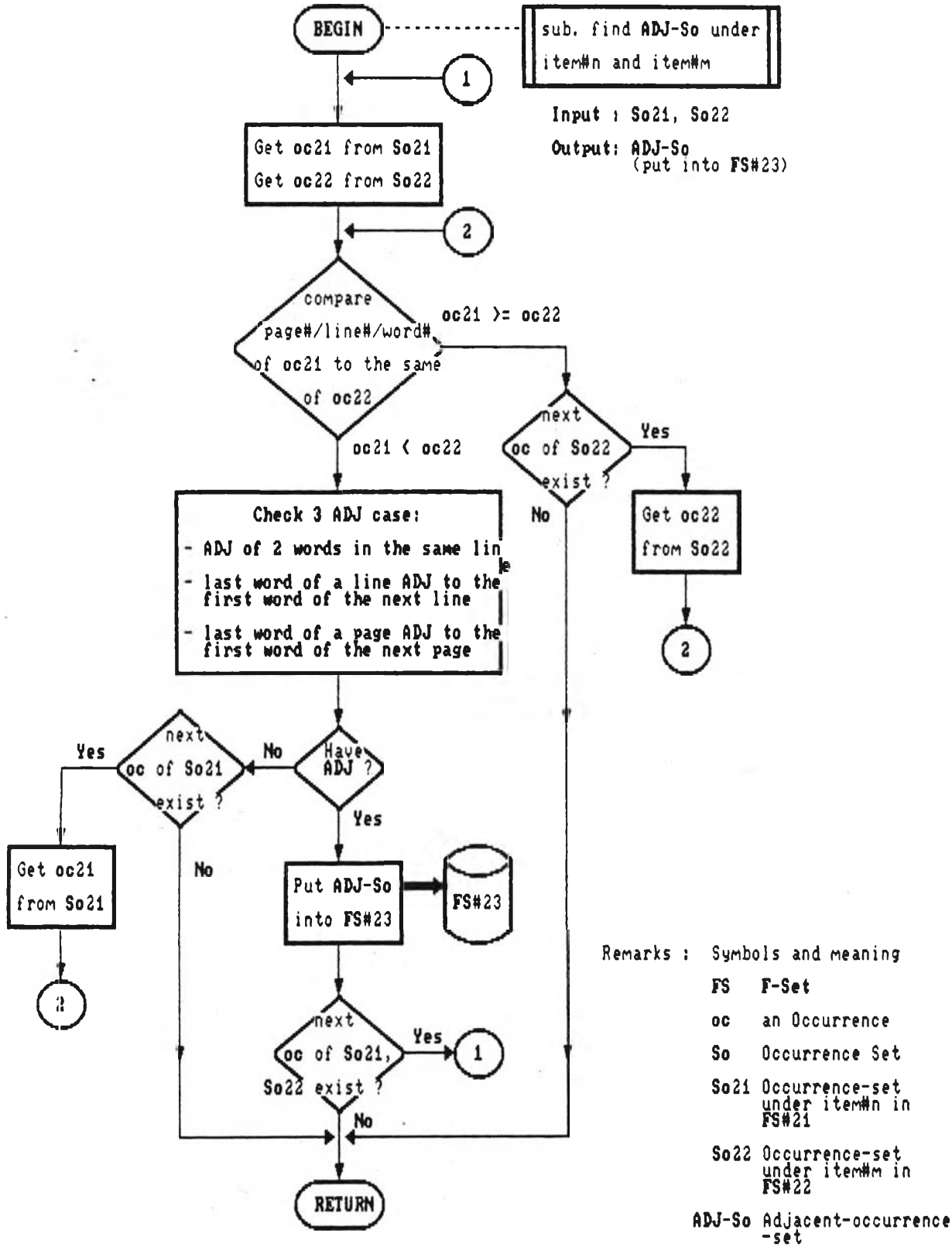
รูปที่ 4.7 แสดงผังงานการดำเนินปฏิบัติการ OR ระหว่างเซท #21 และเซท #22



รูปที่ 4.8 แสดงผังงานการดำเนินปฏิบัติการ NOT ระหว่างเซท #21 และเซท #22



รูปที่ 4.9/1 แสดงผังงานการดำเนินปฏิบัติการ ADJ ระหว่างเซท #21 และเซท #22



รูปที่ 4.9/2 แสดงผังงานการดำเนินปฏิบัติการ ADJ ระหว่างเซท #21 และเซท #22

ดังนั้น #1 @ #2 จะได้ผลเป็นกลุ่มของหมายเลขข้อ ที่มีคำประชิดกันปรากฏอยู่ในข้อเหล่านั้น หมายเลขข้อกลุ่มนี้ จะถูกบันทึกเก็บไว้ในแฟ้มข้อมูล F-Set แฟ้มใหม่ พร้อมทั้งตำแหน่งของคำประชิดในหัวข้อเหล่านั้น จะถูกเก็บลงไปด้วย โดยจะใส่แฟล็กกำกับไว้ทุกตำแหน่งในกลุ่มของตำแหน่งที่ประชิดกันภายในข้อหนึ่ง ๆ และที่ตำแหน่งสุดท้ายของกลุ่ม จะใช้แฟล็กที่แตกต่างออกไป เพื่อบอกว่าเป็นตำแหน่งสุดท้ายของกลุ่มตำแหน่งที่ประชิดกันในข้อนั้น ๆ ในรูปที่ 4.9 ได้แสดงผังงานของการดำเนินปฏิบัติการ ADJ ระหว่างเซท #21 และเซท #22 ผลลัพธ์(ถ้ามี) จะถูกเก็บไว้ในเซท #23

4.3.3 การกำหนดลำดับก่อนหลังของการดำเนินปฏิบัติการ

เมื่อกำหนดให้สามารถเติมตัวปฏิบัติการ หลายตัวไว้ในคำสั่งค้นได้ ก็จำเป็นจะต้องกำหนดหลักเกณฑ์ไว้ให้เป็นที่เข้าใจตรงกันระหว่างผู้ใช้ กับระบบว่า ปฏิบัติการใด จะถูกดำเนินการก่อนหลังอย่างไรบ้าง สำหรับใน BUDSIR-III จะกำหนดลำดับก่อนหลังของการดำเนินปฏิบัติการต่าง ๆ จากลำดับแรกไปลำดับสุดท้าย ดังนี้

ADJ

AND, NOT

OR

สำหรับตัวปฏิบัติการที่มีศักย์เท่ากัน หรือเป็นตัวปฏิบัติการเดียวกัน ก็ให้ถือว่า จะมีลำดับของการดำเนินปฏิบัติการจากซ้ายไปขวา นอกจากนั้น ยังอนุญาติให้ใช้วงเล็บ สำหรับเปลี่ยนกฎเกณฑ์จากที่กำหนดลำดับไว้ดังกล่าวได้ โดยระบบจะพิจารณาดำเนินการในส่วนที่อยู่ภายในวงเล็บเป็นลำดับแรกสุด และกรณีที่มีวงเล็บซ้อนกันหลายชั้น ก็จะเริ่มทำจากวงเล็บชั้นในสุดออกมาทีละชั้น การใช้วงเล็บต้องใส่วงเล็บเปิด และวงเล็บปิดให้ถูกต้อง มิฉนั้น จะถือว่าผิดรูปแบบการป้อนคำสั่งค้นและจะยกเลิกการค้นทันที

4.3.4 ขบวนการสืบค้นสารสนเทศพระไตรปิฎก

ในขบวนการสืบค้นสารสนเทศของ BUDSIR-III จำเป็นต้องกำหนดแนวทางของการดำเนินขั้นตอนที่สำคัญ ๆ ดังต่อไปนี้ คือ

- การแปลย่อยคำสั่งค้น (search formula interpretation)
- การค้นหาคำในโครงสร้างแบบทรี ของแฟ้มพจนานุกรม
- การค้นหาส่วนของคำ หรือ partial word (ประเภทที่ 1) ในแฟ้มพจนานุกรม ด้วยวิธีการของ คนูท-มอริส-แพรท สตริงแมชชิ่ง
- การค้นหาส่วนของคำ หรือ partial word (ประเภทที่ 2) ในแฟ้มพจนานุกรม ด้วย

วิธีการของ คนูท-มอริส-แพรท สตริงแมชชิ่ง (ปรับปรุงใหม่) สำหรับรายละเอียดของแต่ละขั้นตอนนั้น ก็จะอธิบายเป็นลำดับต่อไป

การแปลย่อยคำสั่งค้น

เนื่องจากคำสั่งค้นที่ผู้ใช้ป้อนเข้ามาจะอยู่ในรูปของ **สัญกรณ์แบบเติมกลาง** (infix notation) คือ ป้อนตัวปฏิบัติการไว้ระหว่างคำค้นหรือหมายเลขเซท 2 ตัว ซึ่งเป็นรูปแบบที่ผู้ใช้เข้าใจง่าย แต่เมื่อมองในแง่ของการประมวลผล จะมีความยุ่งยากในการที่ต้องกวาดดูคำสั่งนั้น ๆ กลับไปกลับมา เพื่อจะประมวลผลไปได้ตามลำดับที่ถูกต้อง

ด้วยเหตุนี้ จึงต้องเปลี่ยนคำสั่งค้นที่ผู้ใช้ป้อนเข้ามานั้น ให้อยู่ในรูปของ **สัญกรณ์แบบเติมท้าย** (suffix notation) (Tremblay 1981: 457-472) ทั้งนี้ เป็นการตรวจสอบไปด้วยว่า คำสั่งที่ป้อนเข้ามานั้น มีความผิดพลาดทางวากยสัมพันธ์ (syntax error) หรือไม่ และคำสั่งค้นที่แปลงรูปแล้วนี้ จะมีการเรียงลำดับของตัวปฏิบัติการที่ต้องทำก่อนหลัง จากซ้ายไปขวา ซึ่งพร้อมที่จะถูกนำไปประมวลผลขั้นต่อไป

ตัวอย่างเช่น	**(อานนท + อานนโท) & อายสมา**	เป็นสัญกรณ์แบบเติมกลาง
และจะได้ว่า	**อายสมา อานนท อานนโท + &**	เป็นสัญกรณ์แบบเติมท้าย

ในขั้นตอนประมวลผลถัดไป ก็จะเป็นการแปลคำสั่ง โดยจะต้องใช้ **ตัวเรียงทับซ้อน** (stack) สำหรับเป็นที่เก็บข้อมูลชั่วคราวด้วย ส่วนการประมวลผลจะมีขั้นตอนย่อย ๆ ดังนี้

ก. ดึงข้อมูลจากคำสั่งค้นมา 1 โทเคน (token)

ข. พิจารณาโทเคนนั้นว่าเป็น ตัวปฏิบัติการ (operator) หรือ ตัวถูกปฏิบัติการ (operand)

- ถ้าเป็นตัวถูกปฏิบัติการ จะพิจารณาว่าเป็น **คำศัพท์** หรือเป็น **หมายเลขเซท**ของ F-Set
  - ถ้าเป็นหมายเลขเซท (#n) จะถูกตรวจสอบว่ามีอยู่จริงหรือไม่ แล้วจึงนำไปฝากเก็บไว้ใน ตัวเรียงทับซ้อน จากนั้นไปทำข้อ จ.
  - ถ้าเป็นคำศัพท์ จะใช้คำนั้นค้นสารสนเทศจากแฟ้มพระไตรปิฎก แล้วสร้างเป็นแฟ้ม F-Set ชั่วคราว (ซึ่งจะยังไม่ได้รับหมายเซท และลงทะเบียนในแฟ้มไดเรกทอรีของ F-Set) แล้วเก็บชื่อแฟ้มชั่วคราวนี้ไว้ใน ตัวเรียงทับซ้อน จากนั้นไปทำข้อ จ.
- ถ้าเป็นตัวปฏิบัติการจะดึง ตัวถูกปฏิบัติการที่ฝากเก็บไว้ใน ตัวเรียงทับซ้อน ออกมา 2 ตัว แล้วไปทำข้อ ค.

ค. พิจารณาว่า ตัวปฏิบัติการเป็น AND, OR, NOT หรือ ADJ แล้วนำตัวถูกปฏิบัติการ 2 ตัวนั้นไปดำเนินการตามขั้นตอนของปฏิบัติการนั้น ๆ จนเสร็จ แล้วไปทำข้อ ง.

ง. นำผลลัพธ์ที่ได้จากการดำเนินปฏิบัติการไปฝากเก็บไว้ใน ตัวเรียงทับซ้อน แล้วไปทำข้อ จ.

จ. พิจารณาว่า ยังมีโทเคนถัดไปในคำสั่งค้นหรือไม่
- ถ้ายังมี ให้กลับไปทำข้อ ก.
- ถ้าไม่มี ให้ดึงผลลัพธ์จาก ตัวเรียงทับซ้อน ออกมาเป็นผลลัพธ์สุดท้าย ซึ่งจะถูกนำไปแสดงแก่ผู้ใช้ต่อไป

<u>การค้นคำในโครงสร้างแบบทรี ของแฟ้มพจนานุกรม</u>

เนื่องจากแฟ้มพจนานุกรม แยกเก็บคำศัพท์ไว้เป็นกลุ่ม ๆ ตามพยัญชนะตัวแรกของคำศัพท์ที่เหมือนกัน แล้วภายในกลุ่มจะเก็บคำศัพท์ไว้ในโครงสร้างแบบทรี ลักษณะของเทอมที่จะนำมาใช้ค้นหาคำศัพท์ในแฟ้มพจนานุกรมได้โดยตรง มี 3 รูปแบบ ได้แก่
- คำศัพท์เดี่ยว เช่น อานนฺท ปญฺญา สงฺขาร
- ส่วนของคำเติมท้ายด้วย * เช่น วิญฺญาณ* สงฺสาร*
- ส่วนของคำเติมท้ายด้วย ? เช่น สงฺสาร??? ธมฺวิ????

ส่วนขั้นตอนการค้นคำ มีดังต่อไปนี้

ก. เริ่มแรกจะนำส่วนของคำไม่รวม * หรือ ? มาพิจารณาก่อนว่า พยัญชนะตัวแรกของเทอมนั้นตรงกับกลุ่มคำศัพท์ใดในแฟ้มพจนานุกรม ซึ่งในกรณีที่อักษรตัวแรกของเทอมเป็นสระ "เ" หรือ "โ" ก็จะข้ามไปดูอักษรถัดไปที่เป็นพยัญชนะ ต่อไปจะใช้ส่วนของคำนั้น ไล่ค้นหาคำศัพท์ที่เหมือนกันในโครงสร้างแบบทรี

ข. ใช้ส่วนของคำนั้น เทียบเคียงกับคำศัพท์ที่เป็นส่วนยอด (root) ถ้าตรงกันให้ไปทำข้อ ฉ. แต่ถ้าไม่ตรงกันให้ไปทำข้อ ค.

ค. พิจารณาว่า ส่วนของคำมีค่าน้อยกว่า หรือมากกว่าส่วนยอดนั้น
- ถ้ามีค่าน้อยกว่าส่วนยอด ให้ไล่ลงไปทางโหนดซ้าย (left node) ของส่วนยอดนั้น
- ถ้ามีค่ามากกว่าส่วนยอด ให้ไล่ลงไปทางโหนดขวา (right node) ของส่วนยอดนั้น

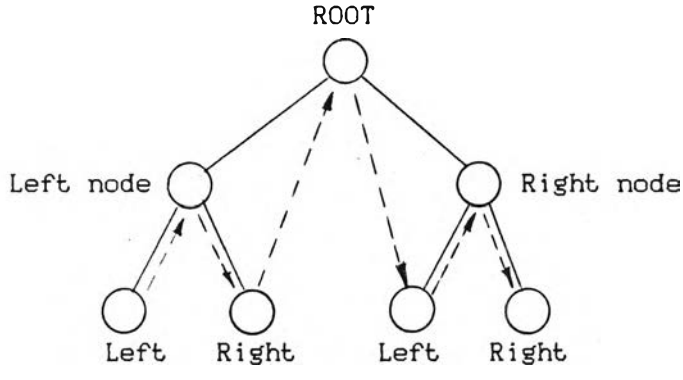
และพิจารณาว่า มีโหนดซ้ายหรือขวาในระดับถัดไป ตามต้องการหรือไม่
- : ถ้าไม่มีโหนดระดับถัดไปทางด้านที่จะไล่ลงนั้น แสดงว่า ไม่มีคำที่ค้นหา ให้ไปทำข้อ ฉ.
- : ถ้ายังมีโหนดระดับถัดไป ก็ให้มองโหนดนั้นเป็นส่วนยอดอีก แล้วย้อนกลับไปทำข้อ ข.

ง. เมื่อได้คำศัพท์ที่ตรงกันแล้ว ต้องดูความถูกต้องอีกครั้ง โดยดูว่า เทอมที่ใช้ค้นเป็น คำเต็ม หรือส่วนของคำที่มี ?, * เติมท้ายอยู่
- ถ้าเทอมใช้ค้นเป็น คำเต็ม แสดงว่า ได้คำศัพท์ที่ตรงกันทั้งคำแล้ว ให้ไปทำข้อ ฉ.
- ถ้าเป็นส่วนของคำที่มี "*" เติมท้าย แสดงว่า คำศัพท์ที่พบนั้น เป็น 1 คำที่อยู่ในข่ายความต้องการ เพราะขึ้นต้นเหมือนส่วนของคำที่ใช้ค้น แล้วไปทำข้อ จ.

- ถ้าเป็นส่วนของคำที่มี "?" เติมท้าย ต้องมานับจำนวนอักษรข้างท้ายของคำศัพท์ ถัดจากส่วนที่ตรงกับเทอมค้นนั้น ว่ามีกี่ตัว ถ้ามีจำนวนอักษรไม่เกินจำนวน "?" ที่เติมท้ายเทอมใช้ค้นก็จะได้ว่า คำศัพท์ที่พบนั้น เป็น 1 คำที่อยู่ในข่ายความต้องการ แล้วไปทำข้อ จ.

จ. หาคำศัพท์อื่น ๆ ที่อยู่ในข่ายความต้องการ โดยไล่ตามตัวชี้บอกคำศัพท์ที่มีค่าน้อยกว่า กับตัวชี้บอกคำศัพท์ที่มีค่ามากกว่า และพิจารณาคำศัพท์นั้น ๆ ว่ามีคุณสมบัติตามต้องการหรือไม่ ด้วยวิธีนี้ ก็จะได้กลุ่มของคำศัพท์ทั้งหมดที่ค้นหา แล้วไปทำข้อ ฉ.

ฉ. บอกผลการค้นแก่ผู้ใช้ว่า ค้นหาพบหรือไม่ และสิ้นสุดการค้นคำ



—— เส้นชี้โยงระหว่างโหนด root-left-right

- - - ทิศทางไล่หาคำตามค่าแอสกี ใช้สำหรับกรณีค้นด้วยส่วนของคำ

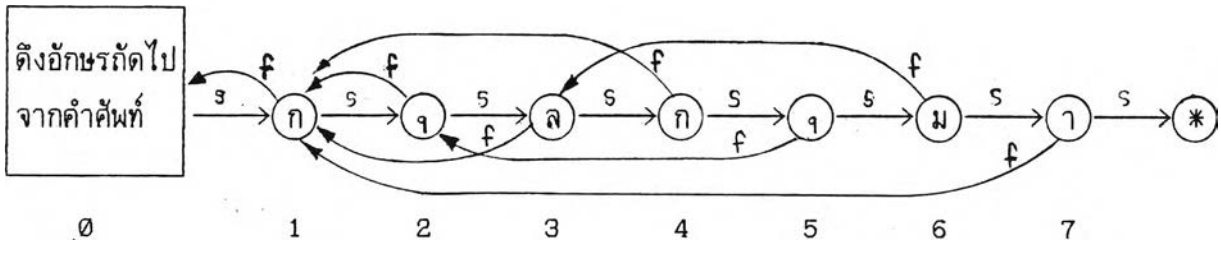
รูปที่ 4.10 แสดงภาพจำลองวิธีการไล่ค้นหาคำศัพท์ในโครงสร้างแบบทรี

<u>การค้นหาส่วนของคำ (ประเภท 1) ในแฟ้มพจนานุกรมโดยวิธีคนุท-มอริส-แพรท</u>

รูปแบบของ ส่วนของคำ (ประเภท 1) ได้แก่ กลุ่มตัวอักษรที่มี "*" นำหน้า และอาจจะมี "*" หรือ "?" เติมไว้ท้ายเทอมที่ใช้ค้นได้ด้วย ตัวอย่างเช่น *กุมาร *กุมาร* *กุมาร?? เป็นต้น

การค้นหาส่วนของคำรูปแบบนี้ จะทำการค้นจากแฟ้มดิกแพค (Dict-Pack) ของพจนานุกรม เมื่อพบคำศัพท์หรือกลุ่มคำศัพท์ที่อยู่ในข่ายความต้องการแล้ว ก็สามารถไปค้นข้อมูลรายละเอียดเกี่ยวกับคำศัพท์นั้น ๆ จากในฐานข้อมูลพระไตรปิฎกได้เช่นเดียวกัน

วิธีการที่นำมาใช้สำหรับค้นหาส่วนของคำ (ประเภท 1) เรียกว่า **วิธีเทียบกลุ่มตัวอักษรของคนุท-มอริส-แพรท** (Knuth-Morris-Pratt string-matching) (Baase 1978: 175-183) ซึ่งมีหลักการที่แตกต่างจากการเทียบเคียงกลุ่มตัวอักษรแบบธรรมดา คือ แทนที่จะนำอักษรที่จะค้นหา ไปเทียบเคียงกับอักษรในคำศัพท์ของพจนานุกรม กลับนำกลุ่มอักษรนั้น มาสร้างผังงานตรวจลอบอักษร ที่ประกอบด้วยขั้นตอนย่อย ๆ ชุดหนึ่ง แล้วนำคำศัพท์จากแฟ้มดิกแพค มาผ่านขั้นตอนการตรวจตามผังงานดังกล่าวแทน ซึ่งก็จะได้ผลลัพธ์ออกมาว่า พบกลุ่มอักษรที่ค้นหา ในคำศัพท์นั้นหรือไม่ ตัวอย่างเช่น ต้องการค้น *กุลกุมา* จะได้ผังงานของคนุท-มอริส-แพรท เป็นดังนี้



รูปที่ 4.11 แสดงแผนผังของคนุท-มอริส-แพรท สร้างจากกลุ่มอักษร "กุลกุมา"

ความหมายของสัญญลักษณ์ต่าง ๆ ในแผนผัง มีดังนี้

(a) วงกลมที่มีตัวอักษรบรรจุอยู่ คือ จุดตรวจสอบอักษรแต่ละจุดของผังงาน โดยเปรียบเทียบอักษรที่ดึงมาจากคำศัพท์ว่า ตรงกับอักษรที่บรรจุอยู่ในวงกลมหรือไม่

(*) วงกลมที่มี "*" บรรจุอยู่ คือ จุดสิ้นสุดการตรวจ ถ้าการตรวจดำเนินผ่านทุกขั้นตอนไปถึงจุดนี้ แสดงว่า พบกลุ่มอักษรที่ต้องการในคำศัพท์ที่นำมาตรวจ

$\xrightarrow{s}$ ลูกศรที่มีอักษร s กำกับ คือ success link หมายถึง ตัวชี้บอกทิศทางที่จะไปเมื่อการเปรียบเทียบอักษรที่จุดตรวจนั้น ๆ สำเร็จ คือ เปรียบเทียบแล้วอักษรตรงกัน

$\xrightarrow{f}$ ลูกศรที่มีอักษร f กำกับ คือ failure link หมายถึง ตัวชี้บอกทิศทางที่จะไปเมื่อการเปรียบเทียบอักษรที่จุดตรวจนั้น ๆ ล้มเหลว คือ เปรียบเทียบแล้วอักษรไม่ตรงกัน

<u>การสร้างผังงานของคนุท-มอริส-แพรท</u> มีขั้นตอนดังนี้

ก. กำหนดจุดตรวจสอบในผังงาน โดยใช้อักษรแต่ละตัวของกลุ่มอักษรต้นแบบ และให้หมายเลขประจำแก่จุดตรวจเหล่านั้น เริ่มตั้งแต่เลข 1 เป็นต้นไป ส่วนจุดเริ่มต้นสำหรับดึงอักษรถัด

ไปจากคำที่จะตรวจ ก็กำหนดให้เป็น หมายเลข ๐

ข. สร้างตัวชี้ FLINK (failure link) ของแต่ละจุดตรวจ สำหรับบอกตำแหน่งที่จะไป เมื่อการเปรียบเทียบอักษรที่จุดนั้น ๆ ให้ผลล้มเหลว การบอกตำแหน่งจะใช้เป็นตัวเลข ตามที่กำหนดไว้ในข้อ ก. ซึ่งการกำหนดค่าให้แก่ FLINK แต่ละตัว มีหลักการ ดังนี้คือ

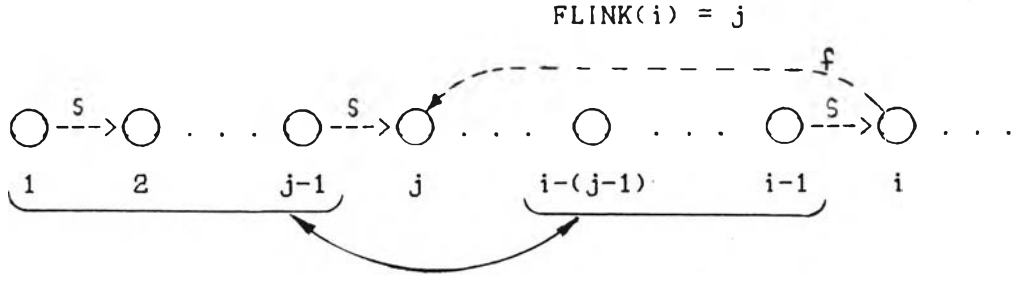
- กำหนดให้ FLINK ของจุดที่ 1 เป็นตำแหน่งที่ ๐ คือ

  FLINK(1) = ๐

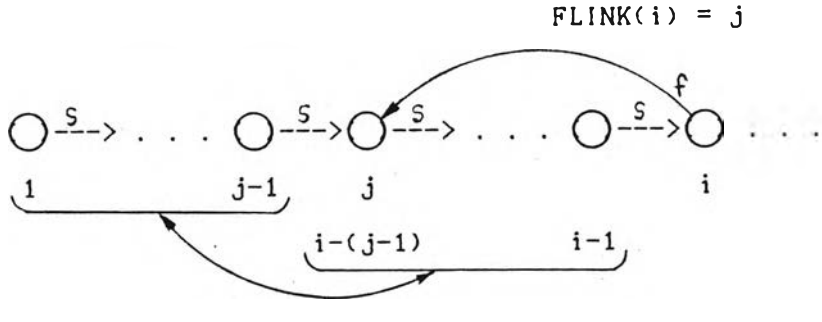
- กำหนดให้ FLINK ของจุดอื่น ๆ คือจุดที่ i เป็นตำแหน่ง j

  FLINK(i) = j

  โดยที่ j < i และมีกลุ่มอักษรย่อยที่จุดเริ่มต้นจำนวน j-1 ตัว ที่เหมือนกันตัวต่อตัว กับกลุ่มอักษรย่อยจำนวน j-1 ตัว ที่นำหน้าอักษรตำแหน่งที่ i ในที่นี้ j ต้องเป็นจำนวนเต็มใหญ่ที่สุด ที่เป็นไปได้ตามข้อกำหนดทั้งสอง (ดังแสดงในรูปที่ 4.13) **แต่** ถ้าไม่มีโหนด j ที่เป็นไปตามเงื่อนไขดังกล่าว ให้กำหนด FLINK(i) = 1



หรืออาจอยู่ในอีกรูปแบบหนึ่ง ดังนี้



รูปที่ 4.12 แสดงการกำหนดค่า FLINK ในกรณีที่มีกลุ่มอักษรย่อยซ้ำกันภายใน

การใช้ผังงานของคนูท-มอริส-แพรทตรวจหาส่วนของคำประเภท 1 ในคำศัพท์

ก. ดึงอักษรมา 1 ตัว จากคำศัพท์ที่นำมาตรวจหา ส่วนของคำ แล้วไปที่จุดตรวจถัดไป

ข. เปรียบเทียบอักษรจากคำศัพท์ กับอักษรที่จุดตรวจว่าตรงกันหรือไม่

- ถ้าตรงกัน ให้ไปทำข้อ ค.
- ถ้าไม่ตรงกัน ให้ย้ายจุดไปตามที่ระบุไว้ใน FLINK ของจุดนั้น แล้วพิจารณาว่าจุดที่ย้ายมานี้เป็น ø หรือไม่
  - ถ้าเป็นตำแหน่ง ø ให้ไปทำข้อ ง.
  - ถ้าไม่เป็นตำแหน่ง ø ให้ใช้อักษรเดิมแล้วไปทำข้อ ข.

ค. ถึงจุดสิ้นสุดแล้วหรือไม่

- ถ้ายังไม่ถึงจุดสิ้นสุด ให้ไปทำข้อ ง.
- ถ้าถึงจุดสิ้นสุดแล้ว ก็จบการตรวจแบบ success end

ง. ตรวจดูว่า ยังมีอักษรถัดไปในคำศัพท์หรือไม่

- ถ้ามี ให้ไปทำข้อ ก.
- ถ้าไม่มีอักษรให้ดึงแล้ว ก็จบการตรวจแบบ failure end

การค้นหาส่วนของคำ (ประเภท 2) ในแฟ้มพจนานุกรมโดยวิธีคนูท-มอริส-แพรท(ปรับปรุงใหม่)

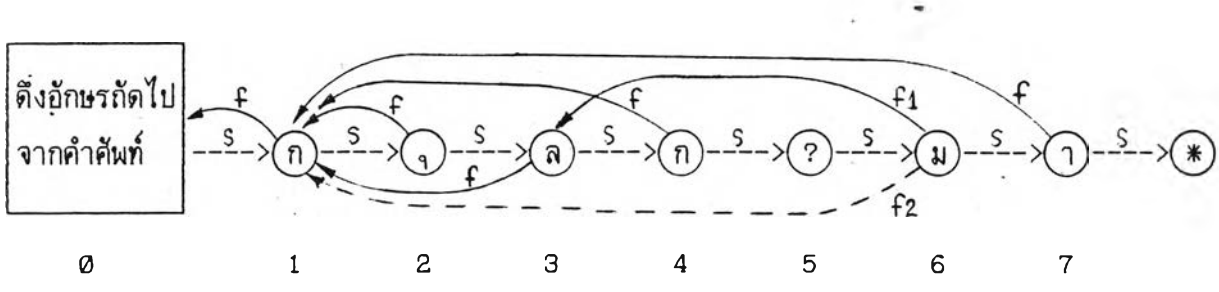
รูปแบบของ ส่วนของคำหรือกลุ่มอักษร (ประเภท 2) ได้แก่ ส่วนของคำที่มี "?" แทรกอยู่ในกลุ่มอักษร และอาจมี "*" นำหน้า หรือมี "*", "?" เติมไว้ท้ายเทอมที่ใช้ค้นได้ด้วย ตัวอย่างเช่น โ?วาท *โ?วา* *โ?วา?ท?? เป็นต้น

การค้นหากลุ่มอักษรรูปแบบนี้ จะทำการค้นจากแฟ้มดิกแพคของพจนานุกรม เช่นเดียวกับประเภทแรก และเมื่อพบคำศัพท์ หรือกลุ่มคำศัพท์ที่อยู่ในข่ายความต้องการแล้ว จะสามารถไปค้นข้อมูลรายละเอียดเกี่ยวกับคำศัพท์นั้น ๆ จากในฐานข้อมูลพระไตรปิฎกได้เช่นเดียวกัน

วิธีการที่นำมาใช้สำหรับค้นหากลุ่มอักษร (ประเภท 2) เรียกว่า **วิธีเทียบเคียงกลุ่มอักษร KMP แบบปรับปรุง** ซึ่งมีหลักการคล้ายคลึงกับที่ใช้ในประเภท 1 คือ นำกลุ่มอักษรนั้น มาสร้างผังงานตรวจสอบอักษร ที่ประกอบด้วยขั้นตอนย่อย ๆ ชุดหนึ่ง แล้วนำคำศัพท์ที่จะตรวจหากลุ่มอักษรที่ค้น มาผ่านขั้นตอนการตรวจตามผังงานดังกล่าว

ส่วนที่แตกต่างจากวิธีการของประเภท 1 ก็คือ จะมีการสร้าง FLINK ไว้สองชุด เฉพาะของจุดตรวจที่ i ซึ่งอยู่ถัดจาก "?" (คือจุดตรวจที่ i-1 และที่จุดนี้จะยอมรับอักษรทุกตัว) เนื่องจากไม่อาจคาดเดาได้ว่า หน้าตำแหน่ง i จะมีกลุ่มอักษรย่อยซ้ำกันหรือไม่ ค่า FLINK(i) ค่าแรกใช้สำหรับกรณีมีอักษรซ้ำ ส่วนอีกค่าคือ FLINK(i)=1 ซึ่งใช้ในกรณีไม่มีอักษรซ้ำ จากนั้นในการตรวจคำศัพท์ เมื่อผ่านจุดตรวจ "?" มาแล้ว ตัวอักษรได้ถูกกำหนดลงไปที่จุดตรวจนั้น

และเมื่อนั้น จึงจะสามารถเลือกได้ว่า FLINK(i) เป็นค่าใด ด้วยเหตุนี้ ในการตรวจอักษร ถ้ามีการย้อนกลับไปตั้งต้นที่จุดตรวจที่ 1 ใหม่ ก็จะถือว่า FLINK(i) ต้องมีการพิจารณาค่าใหม่ทุกคราวไป หลังจากสิ้นการตรวจคำศัพท์แต่ละคำ ก็จะได้ผลลัพธ์ออกมาตามต้องการว่า พบกลุ่มอักษรที่ค้นหา ในคำศัพท์นั้นหรือไม่ ตัวอย่างเช่น ต้องการค้น *กุลก?มา* จะได้ผังงานของ KMP แบบปรับปรุง เป็นดังนี้



f1 : node 5 = "ุ" f2 : node 5 ≠ "ุ"

รูปที่ 4.13 แสดงแผนผังของ KMP แบบปรับปรุง ที่สร้างจากกลุ่มอักษร "กุลก?มา"

<u>การใช้ผังงานของ KMP ปรับปรุงใหม่ ในการตรวจหาส่วนของคำประเภท 2 ในคำศัพท์</u>

ก. ดึงอักษรมา 1 ตัว จากคำศัพท์ที่นำมาตรวจหา ส่วนของคำ แล้วไปที่จุดตรวจถัดไป

ข. ถ้าจุดตรวจนั้นเป็น "?" ให้ถือว่าผ่านการตรวจที่จุดนั้น แล้วพิจารณาอักษรที่นำมาตรวจ ว่าเป็นตัวที่จะทำให้เกิดอักษรซ้ำภายในกลุ่มอักษรต้นแบบ ได้หรือไม่
   - ถ้าจะเกิดอักษรซ้ำ ให้กำหนดค่า FLINK ของจุดตรวจถัดไปตามที่เตรียมไว้
   - ถ้าไม่เกิดอักษรซ้ำ ให้กำหนดค่า FLINK ของจุดตรวจถัดไปเป็น 1

   จากนั้น ไปทำข้อ จ.

ค. เปรียบเทียบอักษรจากคำศัพท์ กับอักษรประจำจุดตรวจว่าตรงกันหรือไม่
   - ถ้าตรงกัน ให้ไปทำข้อ ง.
   - ถ้าไม่ตรงกัน ให้ย้ายจุดไปตามที่ระบุไว้ใน FLINK ของจุดนั้น แล้วพิจารณาว่าจุดที่ย้ายมานี้เป็น ø หรือไม่
     -: ถ้าเป็นตำแหน่ง ø ให้ไปทำข้อ จ.
     -: ถ้าไม่เป็นตำแหน่ง ø ให้ใช้อักษรเดิมแล้วไปทำข้อ ค.

ง. ถึงจุดสิ้นสุดแล้วหรือไม่

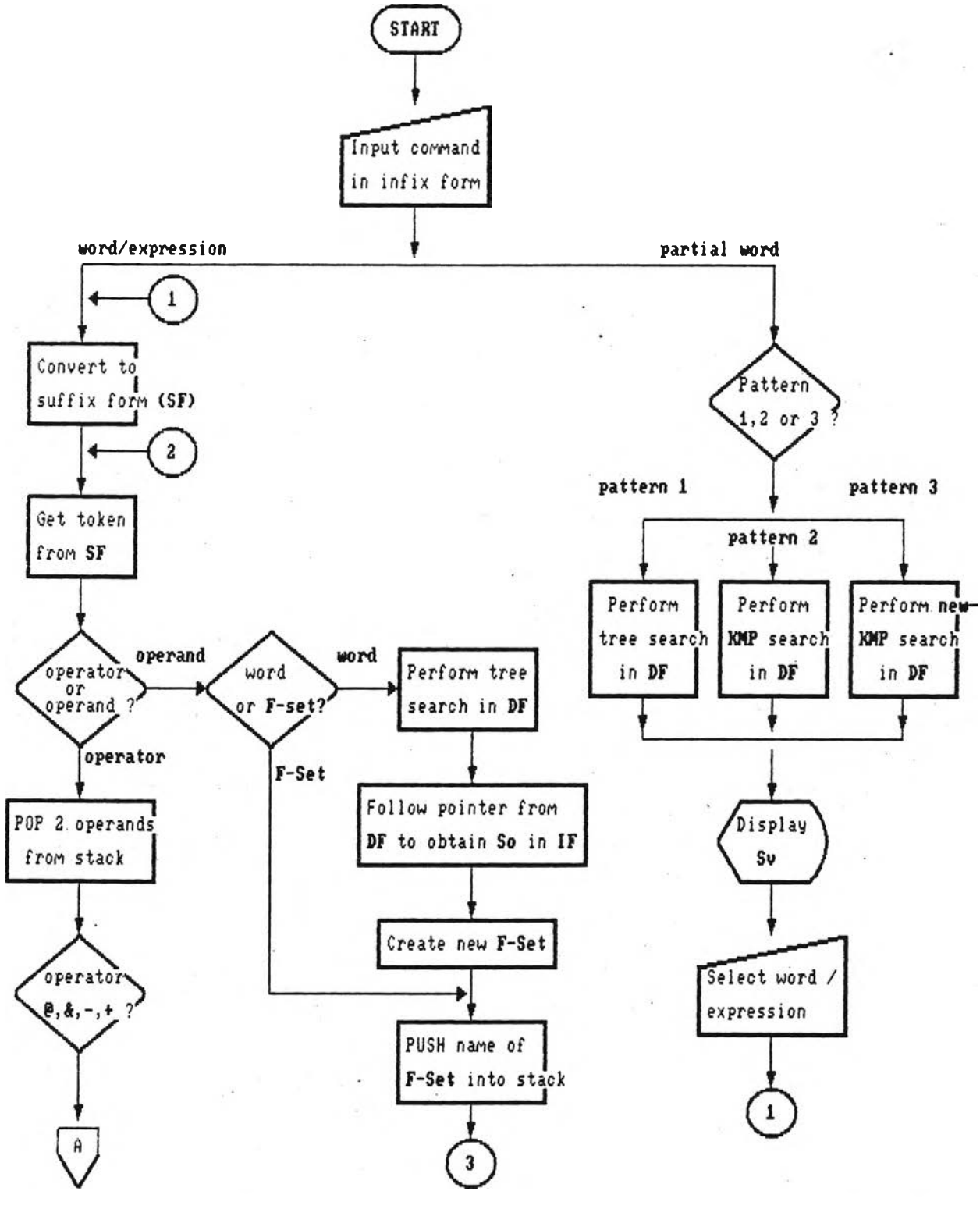
- ถ้ายังไม่ถึงจุดสิ้นสุด ให้ไปทำข้อ จ.
- ถ้าถึงจุดสิ้นสุดแล้ว ก็จบการตรวจแบบ success end

จ. ตรวจดูว่า ยังมีอักษรถัดไปในคำศัพท์หรือไม่

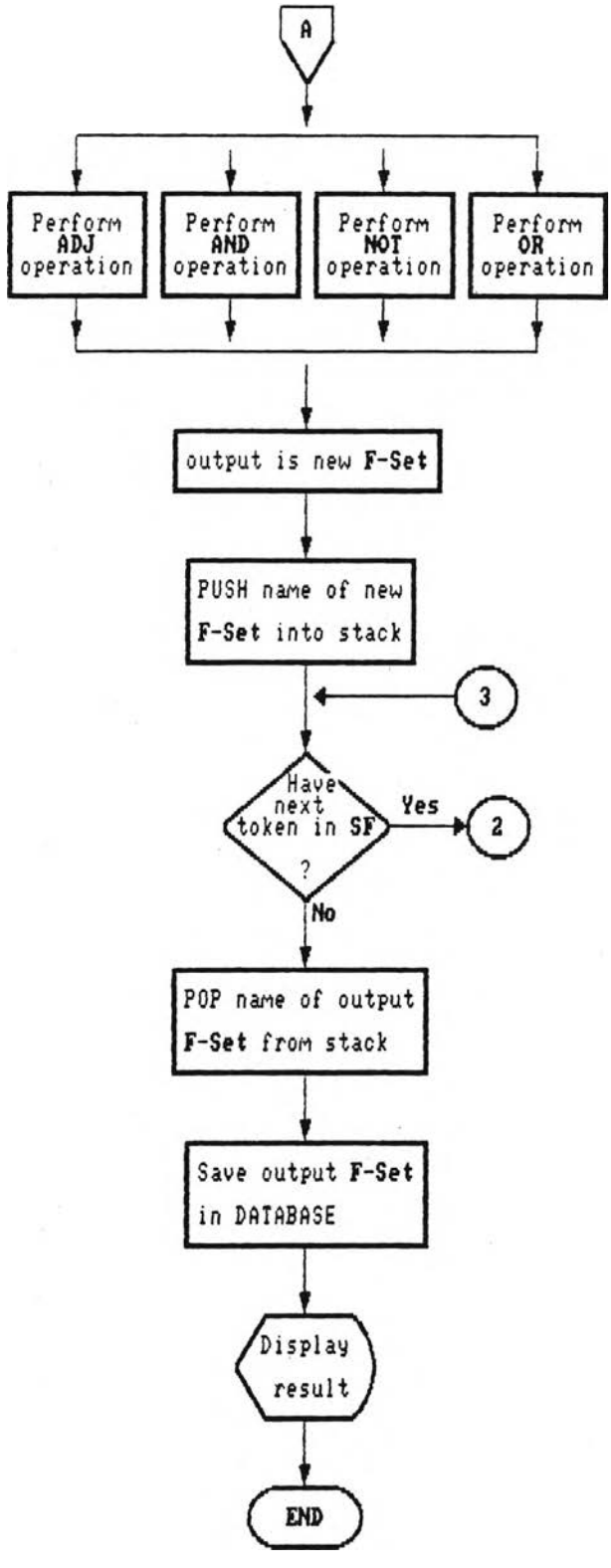
- ถ้ามี ให้ไปทำข้อ ก.
- ถ้าไม่มีอักษรให้ดึงแล้ว ก็จบการตรวจแบบ failure end

ผังงานของขบวนการสืบค้นสารสนเทศของ BUDSIR-III

ขั้นตอนต่าง ๆ ที่กล่าวมาข้างต้น จะถูกนำมาประกอบกันเข้าเป็นขบวนการสืบค้นสารสนเทศพระไตรปิฎกระบบใหม่ ดังแสดงในรูปที่ 4.14



รูปที่ 4.14/1 แสดงผังงานของขบวนการสืบค้นสารสนเทศพระไตรปิฎก ในระบบใหม่



Remarks : Symbols and meaning

SF	Suffix form
DF	Dictionary File
IF	Inverted File
Sv	Vocabulary Set
So	Occurrence Set
v	Vocabulary
p	Phrase

รูปที่ 4.14/2 แสดงผังงานของขบวนการสืบค้นสารสนเทศพระไตรปิฎก ในระบบใหม่