



บทที่ 3

การออกแบบระบบการแปลงตารางตัดสินใจ เป็นส่วนของโปรแกรมคอมพิวเตอร์

แนวความคิดในการแปลงตารางตัดสินใจเป็นส่วนของโปรแกรมคอมพิวเตอร์

จากบทที่ผ่านมาจะพบประโยชน์ของตารางตัดสินใจมีมากโดยเฉพาะอย่างยิ่งเมื่อได้แปลงตารางตัดสินใจเป็นโปรแกรมคอมพิวเตอร์ ซึ่งเทคนิคการแปลงตารางตัดสินใจเป็นโปรแกรมคอมพิวเตอร์นั้นก็มีหลายวิธีดังที่ได้กล่าวมาแล้ว การแปลงตารางตัดสินใจเป็นโปรแกรมคอมพิวเตอร์โดยใช้เทคนิคต่าง ๆ ถ้าทำด้วยมือจะทำได้ยุ่งยากโดยเฉพาะอย่างยิ่งเมื่อมีเงื่อนไขที่สลับซับซ้อนเป็นจำนวนมาก ขั้นตอนการแปลงมีหลายขั้นตอนซึ่งในแต่ละขั้นตอนอาจเกิดข้อผิดพลาดได้จะทำให้ได้โปรแกรมที่ไม่ถูกต้อง ทั้งยังต้องใช้เวลาในการแปลงมากขึ้นตามจำนวนเงื่อนไขและยังต้องคำนึงถึงข้อกำหนดของภาษาคอมพิวเตอร์ที่จะแปลงด้วย นอกจากนี้เมื่อได้โปรแกรมเรียบร้อยแล้วถ้าตารางตัดสินใจมีการแก้ไขปรับปรุง เพิ่มหรือลดบางรายการจะต้องมีการแปลงตารางตัดสินใจที่ได้รับการแก้ไขแล้วให้เป็นโปรแกรมคอมพิวเตอร์ใหม่โดยการทำตั้งแต่เริ่มต้น ซึ่งจะทำให้เสียเวลามากดังนั้นถ้ามีโปรแกรมที่มาช่วยในการแปลงจากตารางตัดสินใจเป็นโปรแกรมคอมพิวเตอร์จะมีประโยชน์มาก โดยผู้ใช้ใช้คำสั่งเพียงไม่กี่คำสั่งก็สามารถแปลงตารางตัดสินใจเป็นโปรแกรมคอมพิวเตอร์ได้ ผู้ใช้ไม่จำเป็นต้องทราบตรรกะของการทำงานในการที่จะให้ได้ส่วนของโปรแกรมในส่วนนี้ออกมา โดยเฉพาะอย่างยิ่งเมื่อมีเงื่อนไขที่สลับซับซ้อนมากๆ จะเป็นการลดข้อผิดพลาดได้ ตารางตัดสินใจที่ผู้ใช้สร้างขึ้นนั้นจะถูกเก็บลงในแฟ้มข้อมูล ซึ่งผู้ใช้สามารถที่จะเรียกขึ้นมาแก้ไขปรับปรุงได้ตลอดเวลา เมื่อมีการแก้ไขหรือใช้คำสั่งเพียงไม่กี่คำสั่งในการแปลงตารางตัดสินใจที่ได้รับการแก้ไขนี้เป็นส่วนของโปรแกรมคอมพิวเตอร์นำไปใช้งานได้ต่อ ทำให้ประหยัดเวลาส่วนของโปรแกรมที่สร้างขึ้นนั้นสามารถนำไปเก็บลงในแฟ้มข้อมูลเพื่อใช้งานต่อไปหรือจะพิมพ์ออกทางเครื่องพิมพ์ หรือจะแสดงผลทางจอภาพก็สามารถทำได้โดยง่าย รวมทั้งตารางตัดสินใจที่สร้างขึ้นสามารถทำได้ทางจอภาพคอมพิวเตอร์ และสามารถสั่งให้พิมพ์ออกทางเครื่องพิมพ์เพื่อการตรวจสอบได้ จึงได้นำเอาแนวความคิดนี้มาใช้ในการออกแบบระบบการแปลงตารางตัดสินใจเป็นส่วนของโปรแกรมคอมพิวเตอร์

ในที่นี้ เราจะแปลงเป็นส่วนของโปรแกรมภาษาโคบอลไมโครคอมพิวเตอร์ ดังนั้นจึงต้องมีการศึกษาถึงข้อกำหนดของภาษา^{นี้}ซึ่งจะกล่าวถึงรายละเอียดต่อไป

ในการออกแบบจะใช้เทคนิคการแปลงตารางตัดสินใจตามวิธี Compiling Technique จะมี 2 ขั้นตอน คือขั้นตอนแรกเป็นการแปลงตารางตัดสินใจเป็นต้นไม้ตัดสินใจที่เหมาะสม ขั้นตอนที่ 2 คือการแปลงจากต้นไม้ตัดสินใจเป็นส่วนของโปรแกรมภาษาโคบอล โดยสร้างรูปแบบ Nested-IF

ขั้นตอนการแปลงตารางตัดสินใจเป็นต้นไม้ตัดสินใจนั้นจะมีหลายวิธี ซึ่งในที่นี้จะใช้วิธีของ Pollack ซึ่ง Pollack 's Algorithm มี 2 ขั้นตอนวิธี คือ

1. การลดเวลาการทำงาน (Minimize the Expected Execution Time) ของโปรแกรมที่สร้างจากตารางตัดสินใจ เป็นการลดจำนวนของเงื่อนไขที่ทดสอบ ซึ่งขึ้นอยู่กับความถี่ของกฎ (Rule Frequencies) และเป็นการลดเวลาที่คาดว่าจะใช้ทั้งหมดในการทำงานสำหรับโปรแกรมที่สร้างขึ้น

2. การลดขนาดของหน่วยเก็บ (Minimize the Storage Space) โปรแกรมที่สร้างจากตารางตัดสินใจต้องใช้ วิธีนี้เป็นการลดจำนวนของการทดสอบในต้นไม้ตัดสินใจให้น้อยที่สุด ดังนั้นจึงเป็นการลดขนาดหน่วยความจำที่ใช้เก็บโปรแกรมที่ถูกสร้างขึ้น

ขั้นตอนวิธีที่ 1 นั้นไม่สามารถประมาณค่าได้ถ้าปราศจากสารสนเทศเพิ่มเติม เนื่องจากวิธีการนี้ขึ้นอยู่กับความน่าจะเป็นของการปรากฏของกฎแต่ละกฎ (Probability of Occurrence of Each of the Rule) ดังนั้นสิ่งนำเข้าสำหรับการทำงานนี้จะต้องมีค่าความน่าจะเป็นเพิ่มเติมมาด้วย ตารางตัดสินใจที่จะใช้วิธีการนี้จะต้องมีการเก็บสถิติของการใช้ตารางมาเป็นระยะเวลาพอควรเพื่อหาค่าความน่าจะเป็นนี้ซึ่งในทางปฏิบัติจะยุ่งยาก การออกแบบจึงใช้ขั้นตอนวิธีที่ 2 ที่จะใช้ในการแปลงตารางตัดสินใจให้เป็นต้นไม้ตัดสินใจโดยมีข้อกำหนดว่าตารางตัดสินใจที่จะใช้แปลงจะเป็นตารางชนิดการรับเข้าจำกัดและไม่มี ความกำกวม (Ambiguity) ดังนั้นระบบที่สร้างจะมีการตรวจสอบในจุดนี้

ขั้นตอนการแปลงตารางตัดสินใจเป็นต้นไม้ตัดสินใจตามขั้นตอนวิธีที่ 2 มีขั้นตอนดังต่อไปนี้

1. คำนวณ Column Count (CC) ของแต่ละกฎในตารางตัดสินใจ ถ้ากำหนดให้ r เป็นจำนวน dash(-) ในกฎ ค่า column count = 2^r

2. กำหนดค่า Dash Count (DC) ให้กับเงื่อนไขทุกแถวในตารางตัดสินใจค่า Dash Count ได้จากการรวมค่า Column Count ของกฎทั้งหมดที่มีค่าของเงื่อนไขเป็น dash(-) ในแถวนั้น ๆ

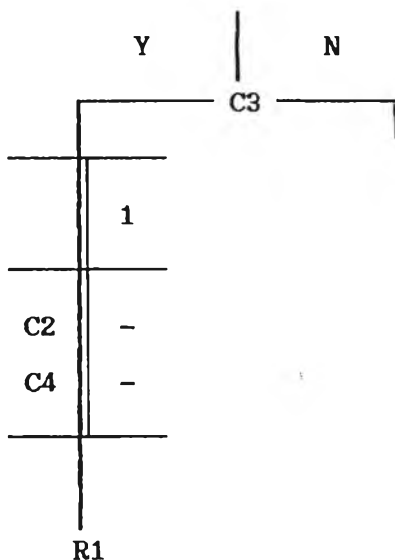
ตัวอย่างการคำนวณค่า Column Count และ Dash Count แสดงในตารางที่ 3.1

CC	::	4	2	2	2	
	::	R1	R2	R3	R4	: DC
C1	::	Y	N	N	N	: 0 <---
C2	::	-	N	Y	N	: 4
C3	::	Y	N	-	-	: 4
C4	::	-	-	Y	Y	: 6

ตารางที่ 3.1 แสดงตารางตัดสินใจที่มีการคำนวณค่า Column Count และค่า Dash Count

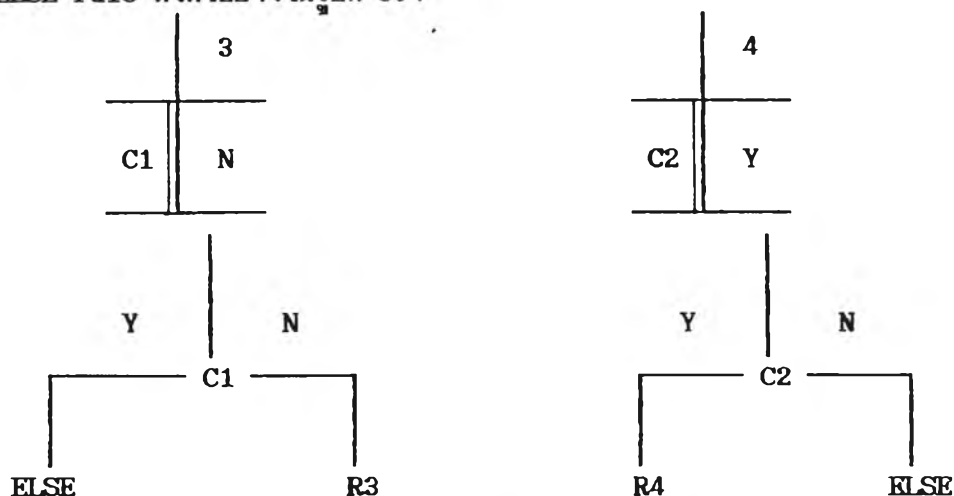
3. แยกตารางตัดสินใจออกเป็นตารางย่อย 2 ตารางโดยการทดสอบเงื่อนไข C_k และตัดเงื่อนไขนั้นออกจากตาราง การเลือกเงื่อนไข (C_k) ให้เลือกจากแถวที่มีค่า Dash Count น้อยที่สุด ในกรณีที่แถวของเงื่อนไขที่มีค่าเท่ากันมากกว่า 1 แถวจะเลือกเงื่อนไขจากแถวนั้น ๆ โดยไม่มีกฎเกณฑ์ ในกรณีที่นี้เราจะให้โปรแกรมเลือกทดสอบเงื่อนไขที่มีค่า Dash Count น้อยที่สุดแถวที่พบก่อน จากตัวอย่างในตารางที่ 3.1 จะเลือกเงื่อนไข C1 ให้ทดสอบเป็นเงื่อนไขแรก

6.2 ตารางย่อยที่มีกฎเพียง 1 กฎและมีค่าเป็นdash(-) สาขานี้จะชี้ไปยังกฎนั้นโดยไม่มี การตรวจสอบอีก ดังตัวอย่างในรูปที่ 3.3



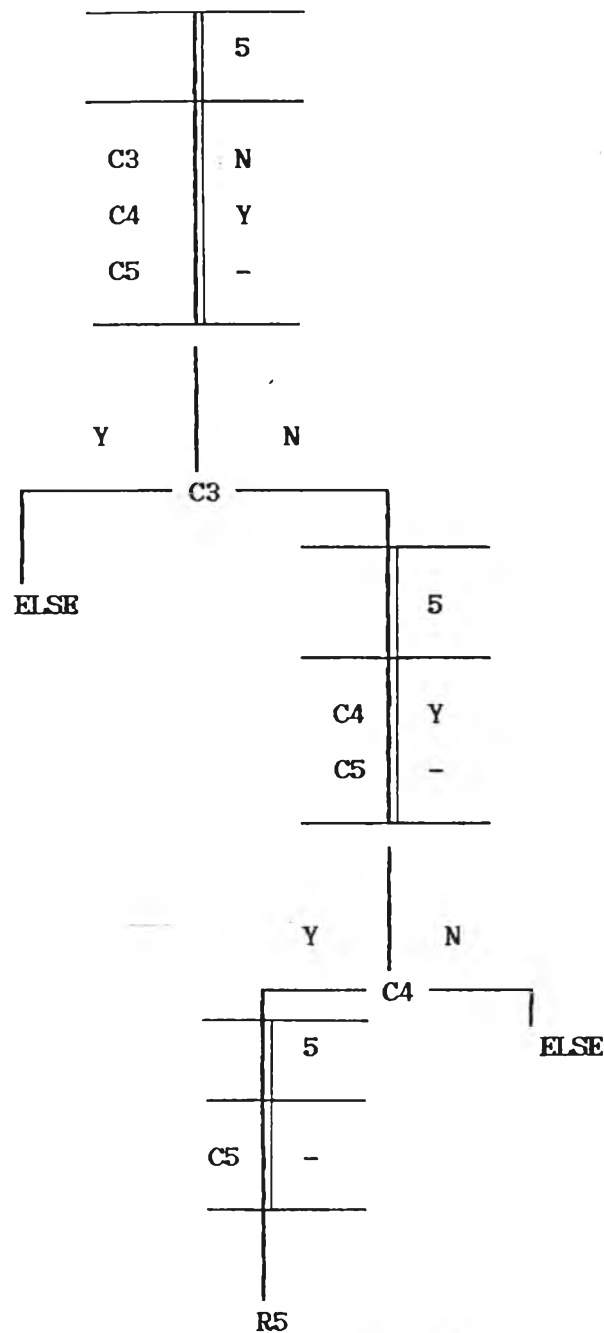
รูปที่ 3.3 แสดงตารางย่อยที่มีกฎ 1 กฎและมีค่าของเงื่อนไขเป็น dash

6.3 ตารางย่อยที่มีกฎเพียง 1 กฎและมีรายการเพียง 1 รายการ ที่มีค่าเป็น 'Y' หรือ 'N' จะตรวจสอบเงื่อนไขนั้น ๆ และสาขาจะชี้ไปที่กฎนั้น อีกสาขาจะเป็น ELSE rule ดังตัวอย่างในรูปที่ 3.4



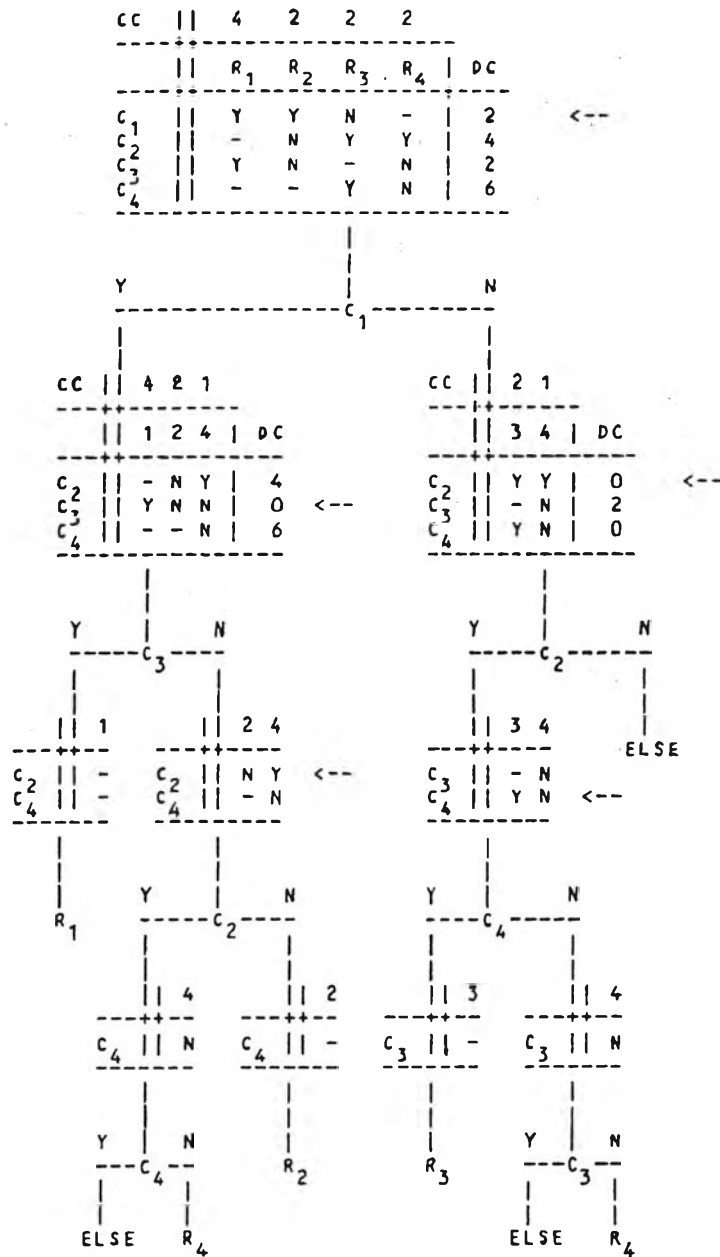
รูปที่ 3.4 แสดงตารางย่อยที่มีกฎ 1 กฎและมี 1 รายการ

6.4 ตารางข้อที่กฎ 1 กฎและมีแถวเงื่อนไขมากกว่า 1 เงื่อนไขและ
 อย่างน้อยที่สุดมีรายการที่มีค่าเป็น 'Y' หรือ 'N' 1 รายการ ในกรณีนี้จะเลือกเงื่อนไข
 ทดสอบและสร้างตารางข้อต่อไปอีกซึ่งจะมีแถวเงื่อนไขอย่างน้อย 1 แถวและสาขาอีกด้าน
 จะเป็น ELSE rule จะทำเช่นนั้นจนกว่าจะเกิดกรณีใดกรณีหนึ่งใน 3 กรณีข้างต้น ดังตัว
 อย่างใน รูปที่ 3.5



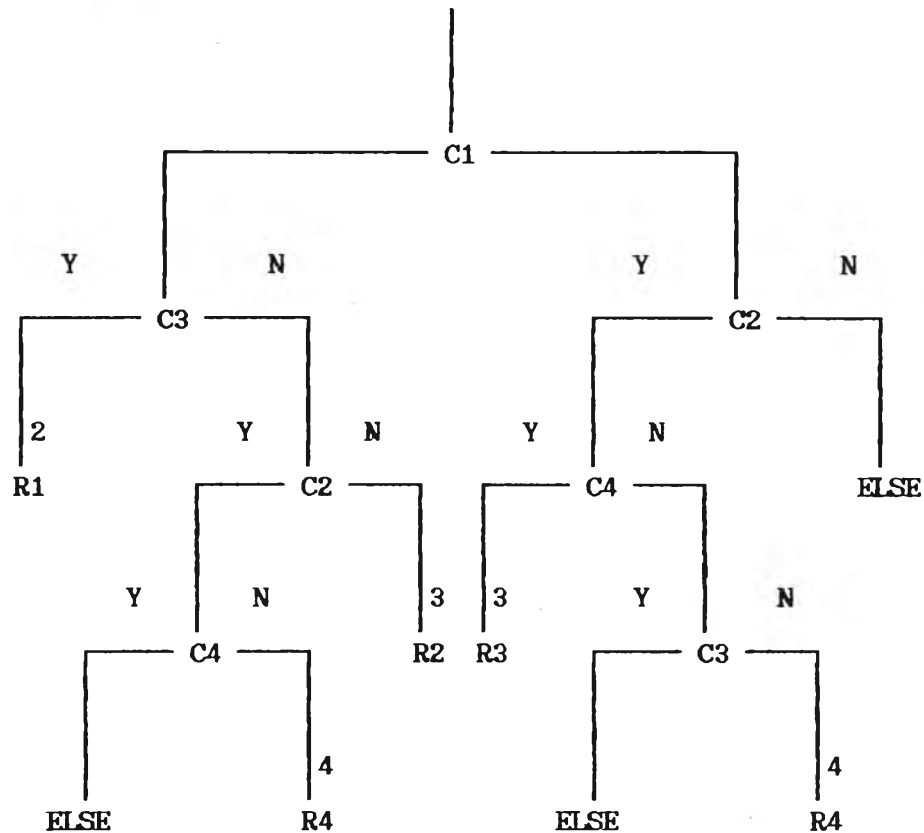
รูปที่ 3.5 แสดงตัวอย่างของกรณีที่เกิดขึ้นในข้อ 6.4

เมื่อกำลังขึ้นตอนทั้งหมดนี้จะ ได้ต้น ไม้ตัดสินใจจากตารางตัดสินใจที่กำหนด ดังตัว
 อย่างขั้นตอนการสร้างต้น ไม้ตัดสินใจจากตารางตัดสินใจอย่างสมบูรณ์ในรูปที่ 3.6



รูปที่ 3.6 แสดงขั้นตอนการสร้างต้น ไม้ตัดสินใจอย่างสมบูรณ์

ถ้าตัดตารางย่อยที่สร้างกันในขณะสร้างต้นไม้ตัดสินใจออกจะได้น้ต้นไม้ตัดสินใจของตัวอย่างเดียวกันนี้ดังในรูปที่ 3.7



รูปที่ 3.7 แสดงต้นไม้ตัดสินใจที่ตัดตารางย่อยออก

เมื่อขั้นตอนที่ 1 ของระบบการแปลงตารางตัดสลับใจเป็นส่วนของโปรแกรมคอมพิวเตอร์เสร็จเรียบร้อยแล้วขั้นต่อไปคือขั้นตอนการแปลงตารางตัดสลับใจเป็นต้นไม้ตัดสลับใจเสร็จจะได้ต้นไม้ตัดสลับใจที่สมบูรณ์เก็บในหน่วยความจำต่อจากนั้นจะทำการขั้นตอนที่ 2 คือการสร้างส่วนของโปรแกรมภาษาโคบอลจากต้นไม้ตัดสลับใจนั้นเป็นการสร้าง NESTED-IF-THEN-ELSE โดยมาจากการวิ่งไปตามสาขาของต้นไม้และเงื่อนไขที่ทดสอบเพื่อหากฎแต่ละกฎในตาราง เมื่อวิ่งจนครบทั้งต้นไม้ตัดสลับใจจะได้ส่วนของโปรแกรมที่แสดงถึงเงื่อนไขและการกระทำของตารางตัดสลับใจทั้งหมด การสร้างส่วนของโปรแกรมนี้เป็นเพียงส่วนหนึ่งของโปรแกรมที่สมบูรณ์ ซึ่งส่วนนี้จะเป็นส่วนที่แสดงถึงเงื่อนไขของโปรแกรม ดังนั้นการออกแบบนี้เราจึงรับสิ่งนำเข้าของเงื่อนไขและการกระทำในรูปของสายอักขระ (String) ซึ่งจะไม่มีการตรวจสอบวากยสัมพันธ์ (Syntax) ของเงื่อนไขและการกระทำว่าได้ใส่ข้อมูลถูกต้องหรือไม่ ตัวแปรต่างๆ ที่แสดงในข้อความของเงื่อนไขและการกระทำจะถือได้ว่าได้มีการประกาศ (declaration) ไว้ก่อนหน้านั้นในโปรแกรมหลัก ซึ่งผู้ใช้จำเป็นต้องรับผิดชอบในส่วนนี้เอง ส่วนของโปรแกรมภาษาโคบอลที่สร้างขึ้นนี้จะนำไปประกอบกับส่วนอื่น ๆ จะได้โปรแกรมที่สมบูรณ์สามารถแปลชุดคำสั่งได้ (Compile) ได้

การออกแบบระบบการแปลงตารางตัดสินใจเป็นส่วนของโปรแกรมคอมพิวเตอร์

การแปลงตารางตัดสินใจเป็นส่วนของโปรแกรมคอมพิวเตอร์ โดยการใช้อคอมพิวเตอร์ช่วยในการแปลงนี้จะต้องมีการพัฒนาโปรแกรมขึ้นเพื่อช่วยแปลง ซึ่งประกอบไปด้วยส่วนต่าง ๆ 4 ส่วน ดังต่อไปนี้

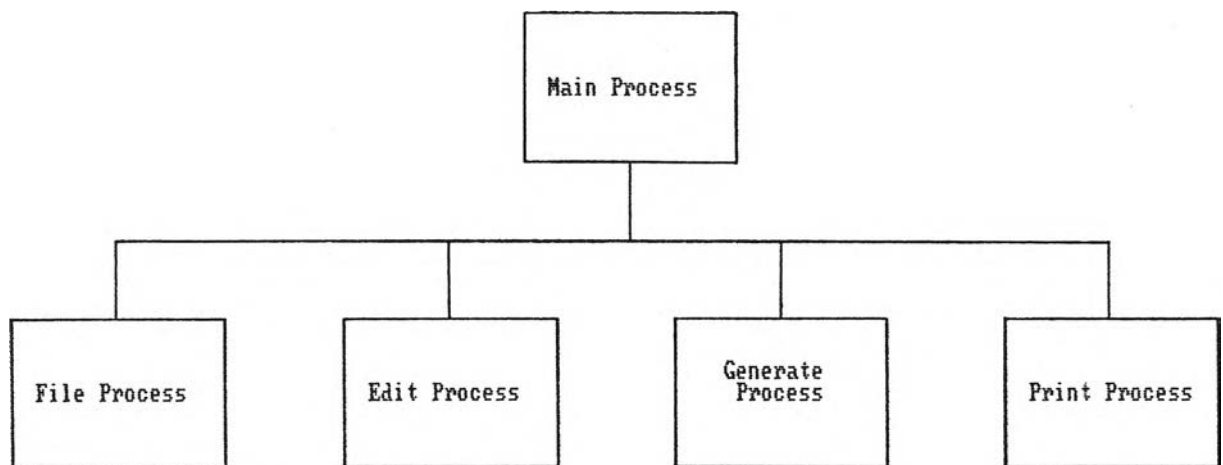
1. การจัดการแฟ้มข้อมูล (File Processing) เป็นส่วนจัดการลบล้างหน่วยความจำ (Clear Memory) สร้างแฟ้มข้อมูลใหม่ การดึงแฟ้มข้อมูลที่สร้างไว้เรียบร้อยแล้วขึ้นมาแก้ไข หรือการเก็บข้อมูลตารางตัดสินใจที่ใช้ในการแปลงเป็นส่วนของโปรแกรมลงแฟ้มข้อมูล

2. การบรรณาธิกรข้อมูล (Edit Processing) เป็นการกำหนดรายละเอียดเกี่ยวกับตารางตัดสินใจ เช่น รายละเอียดเกี่ยวกับเงื่อนไข รายละเอียดของการกระทำจำนวนกฎ รายละเอียดของกฎทั้งหมดในตารางตัดสินใจ การใส่รายละเอียดเหล่านี้สามารถทำได้ทันทีบนจอภาพ ซึ่งข้อมูลทั้งหมดที่ได้ในขั้นตอนนี้จะนำไปใช้ในขั้นตอนการสร้างโปรแกรม

3. การสร้างส่วนของโปรแกรม (Program Generating) เป็นการนำข้อมูลทั้งหมดที่ได้จากการนำเข้าจะนำมาสร้างต้นไม้ตัดสินใจ (Decision Tree Generating) แล้วจึงมีการแปลงจากต้นไม้ตัดสินใจให้อยู่ในรูปของ NESTED_IF_THEN_ELSE ตามรูปแบบของภาษาโคบอล ส่วนของโปรแกรมที่ได้นี้สามารถจะจัดการกับเงื่อนไขตามที่กำหนดผลลัพธ์ที่ได้จากขั้นตอนนี้จะนำมาแสดงทางจอภาพ

4. การพิมพ์ส่วนของโปรแกรมและตารางตัดสินใจ (Print Processing) เป็นการนำส่วนของโปรแกรมที่สร้างจากขั้นตอนการสร้างส่วนของโปรแกรมและข้อมูลของตารางตัดสินใจที่ได้จากการนำเข้ามาแสดงผลออกทางเครื่องพิมพ์หรือเก็บลงแฟ้มข้อมูล

ความสัมพันธ์ของขั้นตอนทั้งหมดนำมาแสดงเป็นผังงานระบบได้ดังในรูปที่ 3.8



รูปที่ 3.8 ผังระบบงานแสดงความสัมพันธ์ของระบบการแปลงตารางตัดสินใจเป็นส่วน
ของโปรแกรมคอมพิวเตอร์

โครงสร้างข้อมูล (Data Structure)

โครงสร้างของข้อมูลที่ใช้ในระบบ ซึ่งนำไปใช้ในการสร้างส่วนของโปรแกรม โดบอลมีดังต่อไปนี้

1. โครงสร้างของข้อมูลส่วนที่แสดงรายการเลือก (Menu) บนจอภาพ

การแสดงรายการให้เลือกทางจอภาพนั้นจะมีทั้งรายการหลักและรายการย่อย ซึ่งรายการย่อยนั้นจะแสดงเป็นกรอบรายการย่อย (Window) หลังจากที่ได้ทำการเลือกรายการหลักซึ่งมีลักษณะเป็นรายการเลือกแบบดึงลง (Pull-down Menu) การเลือกรายการหลักนั้นผู้ใช้สามารถเลือกได้โดยการเลื่อนแถบสว่างไปยังรายการที่ต้องการโดยใช้แป้นลูกศรซ้ายและขวาแล้วกดแป้นป้อนเข้า (ENTER) หรือโดยการกดแป้นฮอตคีย์ (Hot key) คือเลือกกดตัวอักษรตัวแรกของรายการ ส่วนการเลือกรายการย่อยผู้ใช้สามารถเลือกได้ โดยการเลื่อนแถบสว่างขึ้นหรือลงไปยังรายการที่ต้องการโดยใช้แป้นลูกศรขึ้นและลง แล้วกดแป้นป้อนเข้า เมื่ออยู่ในรายการย่อยของรายการหลักใดก็สามารถกดแป้นลูกศรซ้ายและขวาเพื่อเลื่อนไปยังรายการย่อยของรายการหลักอื่นได้ด้วย ดังนั้นโครงสร้างของข้อมูลที่เหมาะสมคือโครงสร้างของข้อมูลแบบ Structure ดังต่อไปนี้

```
Struct menu_frame
{
    int startx, starty, endx, endy; /*ตำแหน่งของกรอบแสดงรายการ*/
    unsigned char *p; /*ตัวชี้ไปยังรายการเลือกแต่ละ
        รายการ*/
    char **menu; /*ชื่อของรายการเลือก*/
    char *keys; /*ตัวอักษรที่ใช้เป็นฮอตคีย์*/
    int border; /*ถ้าไม่ตีกรอบจะมีค่าเป็น 0 */
    int count; /*จำนวนของรายการในรายการเลือก
        ที่ให้เลือก*/
    int curr;
    int active; /*สถานะว่าแสดงอยู่บนจอภาพหรือไม่*/
};
```

2. โครงสร้างของข้อมูลส่วนที่แสดงข้อความหลังจากเลือกรายการย่อย

เมื่อมีการเลือกรายการหลักจะดึงรายการย่อยของรายการหลักที่เลือก หลังจากนั้นผู้ใช้จะเลือกรายการย่อยที่แสดงทางจอภาพ รายการย่อยเหล่านั้นจะมีการผุดขึ้น (Pop-up Menu) กรอบรายการแสดงข้อความบางอย่างและมีบางส่วนของกรอบนั้นให้ผู้ใช้ได้ใส่ข้อความตามที่ระบุลงในช่องที่ได้กำหนดไว้ เช่นให้ผู้ใช้ใส่ filename หลังจากที่ได้เลือก File Load ในรายการเลือกก่อนหน้านี้

ดังนั้นโครงสร้างที่เหมาะสมคือ โครงสร้างแบบ Structure ดังต่อไปนี้

```
Struct win_frame
{
    int startx, endx, starty, endy; /* ตำแหน่งของกรอบรายการ */
    int posx, posy;                /* ตำแหน่งของช่องที่ผู้ใช้ใส่
                                   ข้อความ */
    unsigned char *p;              /* ตัวชี้ไปยังที่เก็บรายการ */
    char *header;                  /* header message */
    char *head-col;               /* header on column */
    int border;                   /* ตีกรอบหรือไม่ ถ้าไม่ตีกรอบ
                                   จะมีค่าเป็น 0 */
    int active;                   /* สถานะว่าแสดงบนจอหรือไม่ */
};
```

3. โครงสร้างข้อมูลของจุดแตกกิ่ง (node) ที่ใช้ในการสร้างต้นไม้ตัดสินใจ

การสร้างส่วนของโปรแกรมภาษาโคบอลจากตารางตัดสินใจนั้น จะต้องทำการแปลงตารางตัดสินใจเป็นต้นไม้ตัดสินใจก่อนจึงทำการแปลงจากต้นไม้ตัดสินใจเป็นส่วนหนึ่งของโปรแกรม ซึ่งการทำงานทั้งหมดจะทำงานกันในหน่วยความจำหลัก ทุกครั้งที่มีการสร้างต้นไม้ตัดสินใจจะต้องมีการขอเนื้อที่ของหน่วยความจำหลัก และเมื่อมีการเลิกใช้แล้วจะทำการคืนเนื้อที่ให้แก่ระบบ ดังนั้นโครงสร้างของข้อมูลที่เหมาะสมคือ โครงสร้างแบบ Structure ดังต่อไปนี้

```

Struct node
{
    int no_cond;           /* จำนวนเงื่อนไข */
    int no_rule;          /* จำนวนกฎ */
    int cond;             /* ลำดับของเงื่อนไขที่เลือกมาทดสอบ */
    struct node *f_node;  /* ตัวชี้ที่ไปยัง father node */
    struct node *l_node;  /* ตัวชี้ที่ไปยังจุดแตกกิ่งทางด้านซ้ายของต้นไม้
                          ตัดสินใจ ในกรณีที่ไม่มีจุดแตกกิ่งทางด้านซ้าย
                          จะมีค่าเป็น NULL */
    struct node *r_node;  /* ตัวชี้ที่ไปยังจุดแตกกิ่งทางด้านขวาของต้นไม้
                          ตัดสินใจ ในกรณีที่ไม่มีจุดแตกกิ่งทางด้านขวา
                          จะมีค่าเป็น NULL */

    char l_select;        /* ค่าของเงื่อนไขที่ใช้ทดสอบทางด้านซ้าย */
    char r_select;        /* ค่าของเงื่อนไขที่ใช้ทดสอบทางด้านขวา */
    int *rule_arr;        /* ตัวชี้ไปยังตารางเก็บลำดับที่ของกฎ */
    int *cond_arr;        /* ตัวชี้ไปยังตารางเก็บลำดับที่ของเงื่อนไข */
    char *val_tbl;        /* ตัวชี้ไปยังตารางเก็บค่าของตาราง
                          ตัดสินใจ */

    char pass;           /* pass flag */
};

```

4. โครงสร้างของข้อมูลสำหรับส่วนใช้เก็บข้อมูลชั่วคราว

ระบบการแปลงตารางตัดสนใจเป็นส่วนของโปรแกรมคอมพิวเตอร์นี้ จะมีการรับข้อมูลของตารางตัดสนใจมาเก็บในหน่วยความจำหลักชั่วคราวและนำข้อมูลนั้น ๆ ไปใช้ในการทำงานตามส่วนต่าง ๆ เช่น ใช้ในการสร้างต้นไม้ตัดสนใจ ใช้ในการสร้างส่วนของโปรแกรมภาษาโคบอล ใช้ในการพิมพ์ รวมทั้งการจัดเก็บข้อมูลนั้นลงแฟ้มข้อมูลชนิดต่าง ๆ ดังนั้นส่วนที่ใช้เป็นที่เก็บข้อมูลชั่วคราวจะมีโครงสร้างของข้อมูลที่เหมาะสมคือ โครงสร้างแบบตาราง ดังต่อไปนี้

4.1 cond_tbl เป็นตารางของตัวอักษร ที่เก็บค่าของเงื่อนไขของทุก ๆ กฎในตารางตัดสนใจ ซึ่งมีขนาดเท่ากับจำนวนเงื่อนไข (no_cond) คูณกับจำนวนเงื่อนไข (no_rule)

4.2 act_tbl เป็นตารางของตัวอักษร ที่เก็บค่าของการกระทำของทุก ๆ กฎในตารางจะมีขนาดเท่ากับจำนวนการกระทำ (no_act) คูณกับจำนวนกฎ (no_rule)

4.3 rule_pos เป็นตารางของตัวเลข ที่เก็บลำดับที่ของกฎมีขนาดเท่ากับจำนวนกฎ

4.4 cond_str เป็นตารางของตัวอักษร ที่เก็บข้อความของเงื่อนไขแต่ละเงื่อนไข

4.5 act_str เป็นตารางของตัวอักษร ที่เก็บข้อความของการกระทำแต่ละการกระทำ

4.6 val_tbl เป็นตารางของตัวอักษร ที่เก็บค่าของเงื่อนไขของทุก ๆ กฎในตารางตัดสนใจที่สร้างขึ้นในขั้นตอนการสร้างตารางย่อย

4.7 cond_arr เป็นตารางของตัวเลข ที่เก็บตำแหน่งของเงื่อนไขจะถูกสร้างขึ้นในขณะสร้างตารางย่อยเมื่อทำการสร้างต้นไม้ตัดสนใจ

4.8 rule_arr เป็นตารางของตัวเลข ที่เก็บตำแหน่งของกฎจะถูกสร้างขึ้นในขณะสร้างตารางย่อยเมื่อทำการสร้างต้นไม้ตัดสนใจ

4.9 prog_str เป็นตารางของตัวอักษร ที่เก็บข้อความที่จะแสดงทางจอภาพหรือพิมพ์ออกทางเครื่องพิมพ์เป็นข้อความของส่วนของโปรแกรมภาษาโคบอลรูปแบบ Nested-IF ที่สร้างขึ้น prg_str จะทำหน้าที่เป็นบัฟเฟอร์ก่อนส่งไปพิมพ์

โครงสร้างของแฟ้มข้อมูล

แฟ้มข้อมูลทั้งหมดของระบบการแปลงตารางตัดสินใจเป็นส่วนของโปรแกรมคอมพิวเตอร์ มีดังต่อไปนี้

1. แฟ้มข้อมูลชนิด TBL (Data Table file)

คือแฟ้มข้อมูลที่เก็บรายละเอียดทั้งหมดของตารางตัดสินใจ ซึ่งเกิดขึ้นในขั้นตอนการบรรณาธิกรตารางตัดสินใจ (Edit Process) เป็นข้อมูลนำเข้าที่ใช้ในการสร้างส่วนของโปรแกรมภาษาโคบอล

1.1 ชื่อแฟ้มข้อมูล

ผู้ใช้กำหนดชื่อแฟ้มข้อมูลชนิดนี้โดยยึดตามหลักมาตรฐานการตั้งชื่อของโปรแกรมคอมพิวเตอร์ระบบ MS-DOS กล่าวคือชื่อแฟ้มข้อมูลมีความยาวสูงสุด 8 ตัวอักษร และระบบจะเพิ่ม extension .TBL ให้โดยอัตโนมัติ

1.2 องค์ประกอบของแฟ้มข้อมูล

1.2.1 ระเบียบความคุม (File Header Record) ประกอบด้วย

1.2.1.1 เขตแรกของแฟ้มข้อมูลเก็บตารางตัดสินใจจะมีคำว่า 'DECISION'

1.2.1.2 จำนวนเงื่อนไขทั้งหมดของตารางตัดสินใจ

1.2.1.3 จำนวนการกระทำทั้งหมดของตารางตัดสินใจ

1.2.1.4 จำนวนกฎทั้งหมดของตารางตัดสินใจ

1.2.2 เนื้อหาของเงื่อนไขเป็นส่วนเก็บรายละเอียดของเงื่อนไขแต่ละเงื่อนไข ขนาดของเนื้อหาขึ้นอยู่กับจำนวนเงื่อนไขที่กำหนดและจำนวนตัวอักษรของแต่ละเงื่อนไข โดยเรียงตามลำดับที่ป้อนข้อมูล ซึ่งได้กำหนดให้ใส่เงื่อนไขได้สูงสุด 8 เงื่อนไข

1.2.3 เนื้อหาของการกระทำ เป็นส่วนเก็บรายละเอียดของการกระทำแต่ละการกระทำ ขนาดของเนื้อหาขึ้นอยู่กับจำนวนการกระทำที่กำหนดและจำนวนตัวอักษรของแต่ละการกระทำ โดยจะเรียงตามลำดับการป้อนข้อมูล กำหนดให้ใส่การกระทำได้สูงสุด 8 การกระทำ

1.2.4 รายละเอียดของกฎ (Rule Entry) ในตารางตัดสินใจ จะเก็บข้อมูลของกฎทั้งหมดเรียงตามลำดับ จากเงื่อนไขที่ 1 ของกฎที่ 1 จนถึงเงื่อนไขสุดท้ายของกฎสุดท้าย ตามด้วยการกระทำที่ 1 ของกฎที่ 1 จนถึงการกระทำสุดท้ายของกฎสุดท้ายและจะเพิ่มการกระทำของ ELSE rule เงื่อนไขและการกระทำของกฎแต่ละชุดจะขึ้นด้วยช่องว่าง 1 ช่อง ค่าของเงื่อนไข (Condition Entry) แต่ละรายการจะเป็น Y, N หรือ - ส่วนค่าของการกระทำ (Action Entry) แต่ละรายการจะเป็น X หรือช่องว่าง

ตัวอย่างการเก็บข้อมูลของแฟ้มข้อมูล TBL เช่น กำหนดให้ ตารางตัดสินใจมีเงื่อนไข 2 เงื่อนไข การกระทำ 3 การกระทำ กฎ 4 กฎจะปรากฏรายละเอียดของตารางตัดสินใจที่เก็บในแฟ้มข้อมูลดังนี้

```
DECISION2 3 4CONDITION1
CONDITION2
ACTION1
ACTION2
ACTION3
NNY NYY NY- XX X   X XX XX
```

2. แฟ้มข้อมูลชนิด COB (Source Program File)

คือแฟ้มข้อมูลที่เก็บส่วนของโปรแกรมภาษาโคบอลที่ระบบสร้างขึ้น แฟ้มข้อมูลชนิดนี้เกิดขึ้นในขั้นตอนการสร้างส่วนของโปรแกรม (Generate program)

2.1 ชื่อแฟ้มข้อมูล

ผู้ใช้กำหนดชื่อแฟ้มข้อมูลชนิดนี้ โดยยึดตามหลักมาตรฐานการตั้งชื่อของโปรแกรมคอมพิวเตอร์ระบบ MS-DOS กล่าวคือชื่อแฟ้มข้อมูลมีความยาวสูงสุด 8 ตัวอักษร หรือแฟ้มข้อมูลชนิดนี้มีชื่อตามแฟ้มข้อมูลชนิด TBL หรือ LST (ถ้ามีการกำหนดไว้ก่อนหน้า) และระบบจะเพิ่ม extension .COB ให้โดยอัตโนมัติ

2.2 องค์ประกอบของแฟ้มข้อมูล

ประกอบด้วยส่วนของโปรแกรมภาษาโคบอลในรูปของ NESTED-IF-THEN-ELSE ที่ระบบสร้างขึ้น

3. แฟ้มข้อมูลชนิด LST (Listing Table File)

คือแฟ้มข้อมูลที่เก็บตารางตัดสินใจและรายละเอียดของตารางที่จะใช้พิมพ์
ออกรายงาน แฟ้มข้อมูลชนิดนี้จะถูกสร้างขึ้นเมื่อผู้ใช้สั่งพิมพ์ตารางลงแฟ้มข้อมูล (Print
Table To File)

3.1 ชื่อแฟ้มข้อมูล

ผู้ใช้กำหนดชื่อแฟ้มข้อมูลชนิดนี้โดยยึดตามหลักมาตรฐานการตั้งชื่อ
ของโปรแกรมคอมพิวเตอร์ระบบ MS-DOS กล่าวคือชื่อแฟ้มข้อมูลมีความยาวสูงสุด 8 ตัวอักษร
หรือแฟ้มข้อมูลชนิดนี้ตั้งชื่อตามแฟ้มข้อมูลชนิด TBL หรือ COB (ถ้ามีการกำหนดไว้ก่อนหน้านี)
และระบบจะเพิ่ม extension .LST ให้โดยอัตโนมัติ

3.2 องค์ประกอบของแฟ้มข้อมูล

3.2.1 จำนวนเงื่อนไข

3.2.2 จำนวนการกระทำ

3.2.3 จำนวนกฎ

3.2.4 รายละเอียดของเงื่อนไขแต่ละเงื่อนไข

3.2.5 รายละเอียดของรายการแต่ละรายการ

3.2.6 ตารางตัดสินใจซึ่งแสดงกฎทั้งหมดที่มีอยู่ในตาราง

การศึกษาคำสั่งเงื่อนไขของภาษาโคบอล (COBOL Condition Statement)

ภาษาโคบอลที่ใช้ในการวิจัยครั้งนี้ ได้ศึกษาจากภาษาโคบอลของบริษัทไมโครซอฟท์ เวอร์ชัน 2.0 การทำงานของคำสั่งเงื่อนไขจะมีการตรวจสอบเงื่อนไขที่กำหนดไว้ว่าเป็นจริงหรือไม่ ผลที่ได้จากการตรวจสอบจะทำให้การทำงานแยกออกเป็น 2 ส่วน คือ ส่วนของการทำงานเมื่อเงื่อนไขที่ทำการทดสอบเป็นจริง และส่วนของการทำงานเมื่อเงื่อนไขที่ทดสอบเป็นเท็จ คำสั่งมีรูปแบบดังต่อไปนี้

```

                Statement-1          ELSE Statement-2
IF Condition
                NEXT SENTENCE      ELSE NEXT SENTENCE
  
```

Where : Condition is relation condition, sign condition
class condition, condition-name condition, or
a compound condition using AND or OR.

ส่วนของ ELSE NEXT SENTENCE อาจจะได้ถ้าพบพบกับจุดสิ้นสุดของประโยคทันที คำสั่งที่ตามหลังคำสั่ง IF จะถูกปฏิบัติเมื่อเงื่อนไขที่ทดสอบเป็นจริง ส่วนคำสั่งที่ตามหลัง ELSE จะถูกปฏิบัติเมื่อเงื่อนไขเป็นเท็จ เมื่อจบคำสั่งทุกครั้งจะต้องใส่จุด(.) เพื่อแสดงว่าจบการทำงานของคำสั่งนี้

การวิจัยครั้งนี้ได้สร้างคำสั่งเงื่อนไขในรูปแบบของ Nested-IF เป็นการรวมคำสั่ง IF หลาย ๆ คำสั่งในประโยคเดียวกัน ซึ่งมีรูปแบบดังนี้

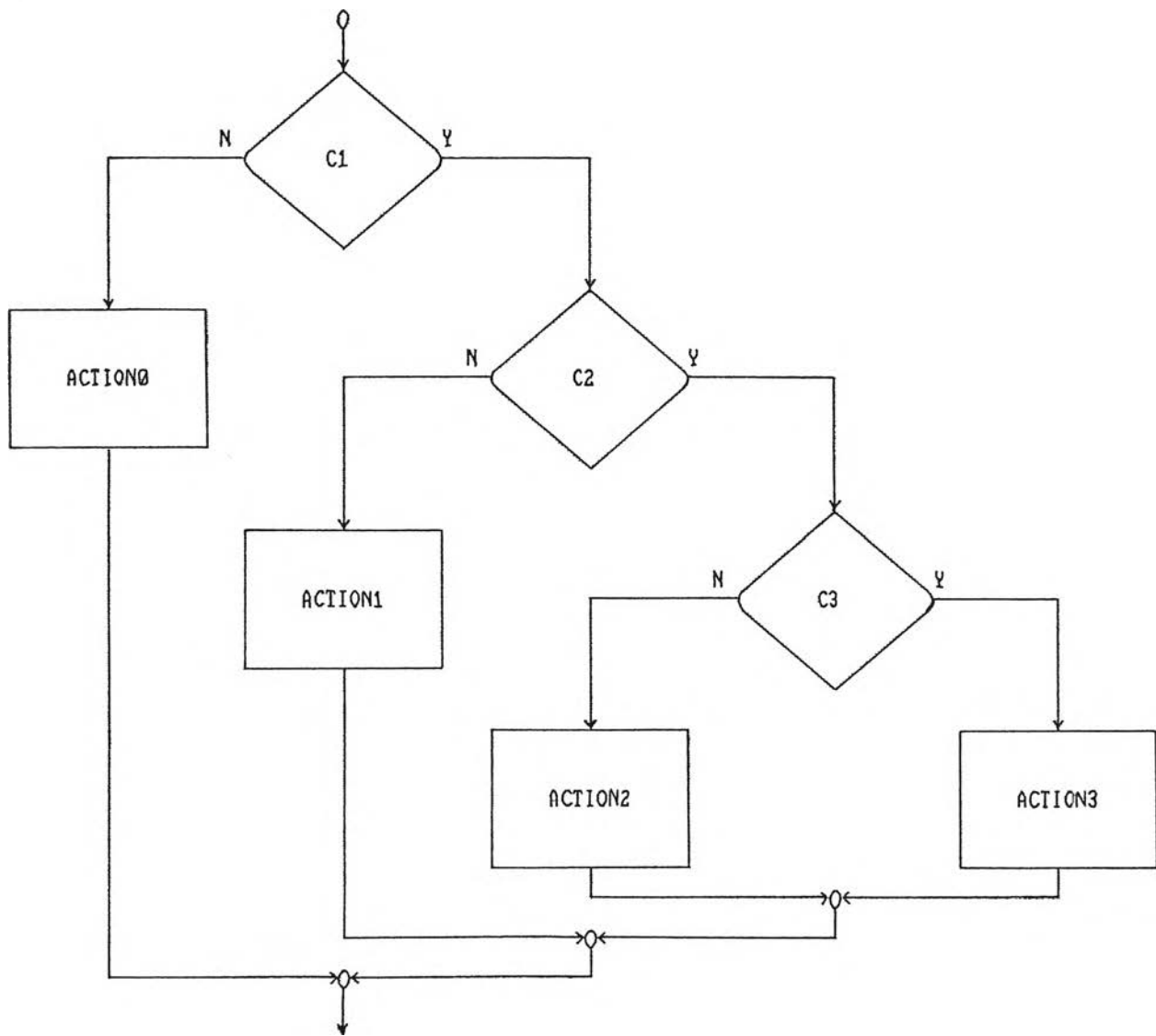
```

IF ... IF ... IF ... ELSE ... ELSE ... ELSE ...
  
```

ตัวอย่างของ Nested-IF ดังต่อไปนี้

```
IF C1
  IF C2
    IF C3
      ACTION3
    ELSE
      ACTION2
  ELSE
    ACTION1
ELSE
  ACTION0.
```

จากตัวอย่างนำมาแสดงเป็นรูปร่างงาน ได้ดังนี้



รูปที่ 3.9 แสดงผังงานของ Nested-IF ของตัวอย่าง