

บทที่ 4

การออกแบบสร้างซอฟต์แวร์

บทนำ

จากบทที่ 3 ได้เสนอแนวความคิดและนิยามต่าง ๆ ที่จำเป็นกับการวิเคราะห์และสังเคราะห์วงจรตรรกเชิงลำดับโดยเสนอให้อยู่ในรูปของคณิตศาสตร์เชิงตรรก ในบทที่ 4 นี้จะนำนิยามคุณสมบัติและกฎต่างๆ ที่ได้กล่าวมาแล้วในตอนต้นมาออกแบบสร้างเป็นซอฟต์แวร์ก่อกำเนิดโปรแกรมสำหรับระบบควบคุมที่ใช้พีแอลซี บล็อกไดอะแกรมของซอฟต์แวร์ในส่วนที่สำคัญได้แสดง อยู่ในหัวข้อ 4.1 ซึ่งจะอธิบายการทำงานในแต่ละบล็อกอย่างละเอียดรวมทั้งโครงสร้างข้อมูลและวิธีการใช้งาน โฟลว์ชาร์ตของแต่ละบล็อกจะบอกถึงการทำงานของระบบซอฟต์แวร์ในแต่ละส่วน

ขั้นตอนการทำงานร่วมกับซอฟต์แวร์ก่อกำเนิดโปรแกรมที่ได้พัฒนาขึ้นมาในวิทยานิพนธ์ฉบับนี้สามารถสรุปได้ดังนี้

- 1) การกำหนดพฤติกรรมของระบบควบคุม (ปฏิบัติการโดยผู้ใช้)
- 2) การสังเคราะห์สมการตรรกจากพฤติกรรมที่กำหนด (ปฏิบัติการโดยซอฟต์แวร์)
- 3) การแปลสมการตรรกเป็นคำสั่งของพีแอลซี (ปฏิบัติการโดยซอฟต์แวร์)
- 4) การจำลองการทำงานรหัสคำสั่งที่ได้โดยใช้พีแอลซีจำลอง (PLC-Software Simulation)

4.1 วิธีการออกแบบสร้างซอฟต์แวร์

การออกแบบสร้างระบบซอฟต์แวร์ในงานวิจัยนี้ ได้ใช้วิธีการออกแบบโดยวิธีที่เรียกว่าการโปรแกรมเชิงวัตถุ (Object Oriented Programming หรือ OOP) หลักการเชิงวัตถุที่นำมาใช้ประกอบด้วย

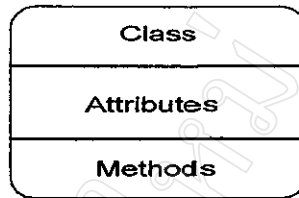
Abstraction คือการที่เราสามารถแสดงความจริงบางสิ่งบางอย่างที่ซับซ้อนอยู่ในรูปแบบจำลองที่ง่ายได้

Encapsulation คือการที่เราสามารถซ่อนรายละเอียดที่นำมาประกอบกันเป็นวัตถุ เพื่อให้ส่วนประกอบหรือวัตถุอื่น ๆ ที่อยู่ในระบบไม่ต้องมาสนใจรายละเอียดที่เก็บอยู่ในวัตถุนั้น

Inheritance คือการทำให้วัตถุสามารถรวมเอานิยามทั้งหมดหรือเพียงบางส่วนของวัตถุอื่น ๆ มาใช้ในการนิยามวัตถุนั้น

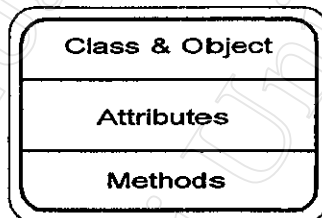
Polymorphism คือการตอบสนองที่แตกต่างกันของวัตถุต่างชนิดต่อข่าวสารชนิดเดียวกันและเพื่อให้ง่ายต่อการทำความเข้าใจต่อไปนี้เป็นารสรุปสัญลักษณ์ ที่ใช้ในการออกแบบซอฟต์แวร์ในการทำวิจัยครั้งนี้ดังต่อไปนี้

- สัญลักษณ์ของคลาส (Class)



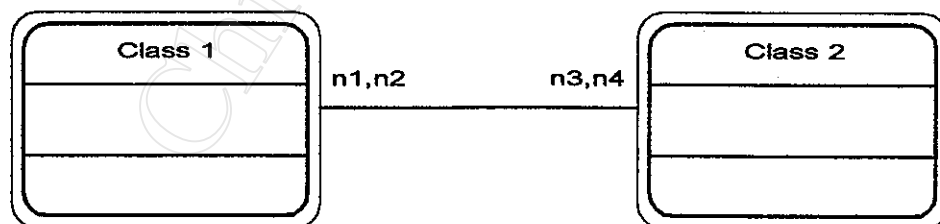
รูปที่ 4.1 สัญลักษณ์ของคลาส

- สัญลักษณ์ของคลาสและวัตถุ (Class & Object)



รูปที่ 4.2 สัญลักษณ์ของคลาสและวัตถุ

● สัญลักษณ์ของความสัมพันธ์ระหว่างวัตถุ วัตถุของคลาส 1 มีความสัมพันธ์ต่อวัตถุของคลาส 2 มีจำนวนตั้งแต่ $n1$ ถึง $n2$ ขึ้น และวัตถุของคลาส 2 มีความสัมพันธ์ต่อวัตถุของคลาส 1 มีจำนวนตั้งแต่ $n3$ ถึง $n4$ ขึ้น



รูปที่ 4.3 สัญลักษณ์ของความสัมพันธ์ระหว่างวัตถุ

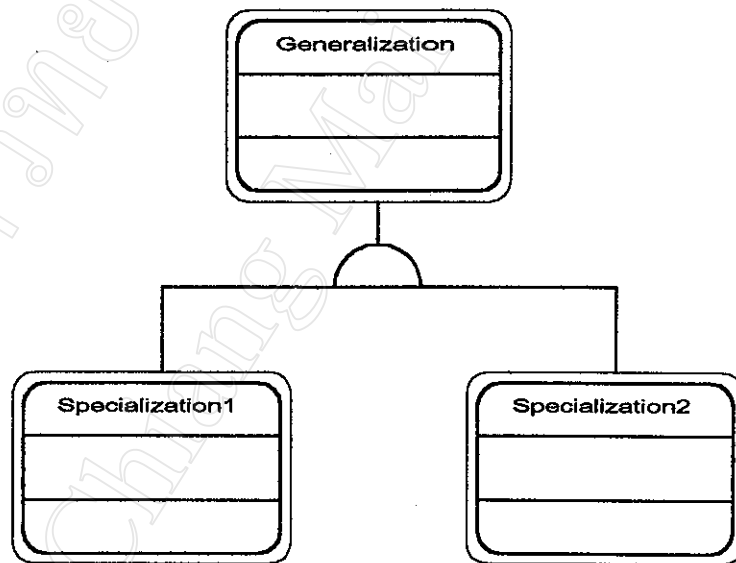
● สัญลักษณ์ของการติดต่อระหว่างวัตถุ แสดงการติดต่อกันระหว่างวัตถุของคลาส Sender กับวัตถุของคลาส Receiver โดยการส่งข่าวสาร ซึ่งเป็นการติดต่อกันระหว่างวัตถุเป็นแบบ 2 ทาง โดยที่

วัตถุของคลาส Sender จะเป็นตัวเริ่มส่งข่าวสารก่อน ส่วนวัตถุของคลาส Receiver จะเป็นตัวรับข่าวสาร และตอบสนองข่าวสารนั้น



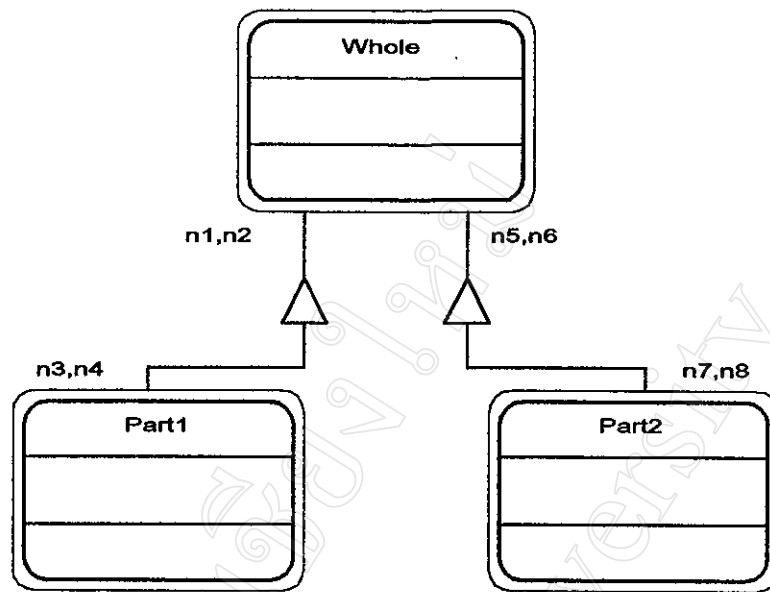
รูปที่ 4.4 สัญลักษณ์ของการติดต่อระหว่างวัตถุ

- สัญลักษณ์ของลำดับชั้นแบบ Gen-Spec Structure ใช้แสดงลำดับชั้นของวัตถุที่สืบทอดกันเป็นลำดับชั้นซึ่งหมายความว่าทั้งคลาสและวัตถุ Specialization1 และ Specialization2 สืบทอดคุณสมบัติและพฤติกรรมจากคลาสและวัตถุ Generalization



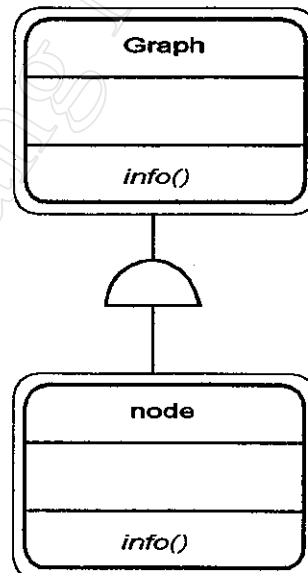
รูปที่ 4.5 สัญลักษณ์ของลำดับชั้นแบบ Gen-Spec Structure

- สัญลักษณ์ของลำดับชั้นแบบ Whole-Part Structure วัตถุของคลาส Whole ต้องประกอบด้วยวัตถุของคลาส Part1 ตั้งแต่ $n1$ ถึง $n2$ ชิ้น และวัตถุของคลาส Part2 จำนวน $n5$ ถึง $n6$ ชิ้น ในทางกลับกันแต่ละวัตถุของคลาส Part1 และ Part2 อาจจะเป็นส่วนประกอบของวัตถุของคลาส Whole ตั้งแต่ $n3$ ถึง $n4$ ชิ้นและ $n7$ ถึง $n8$ ชิ้น ตามลำดับ



รูปที่ 4.6 สัญลักษณ์ของลำดับชั้นแบบ Whole-Part Structure

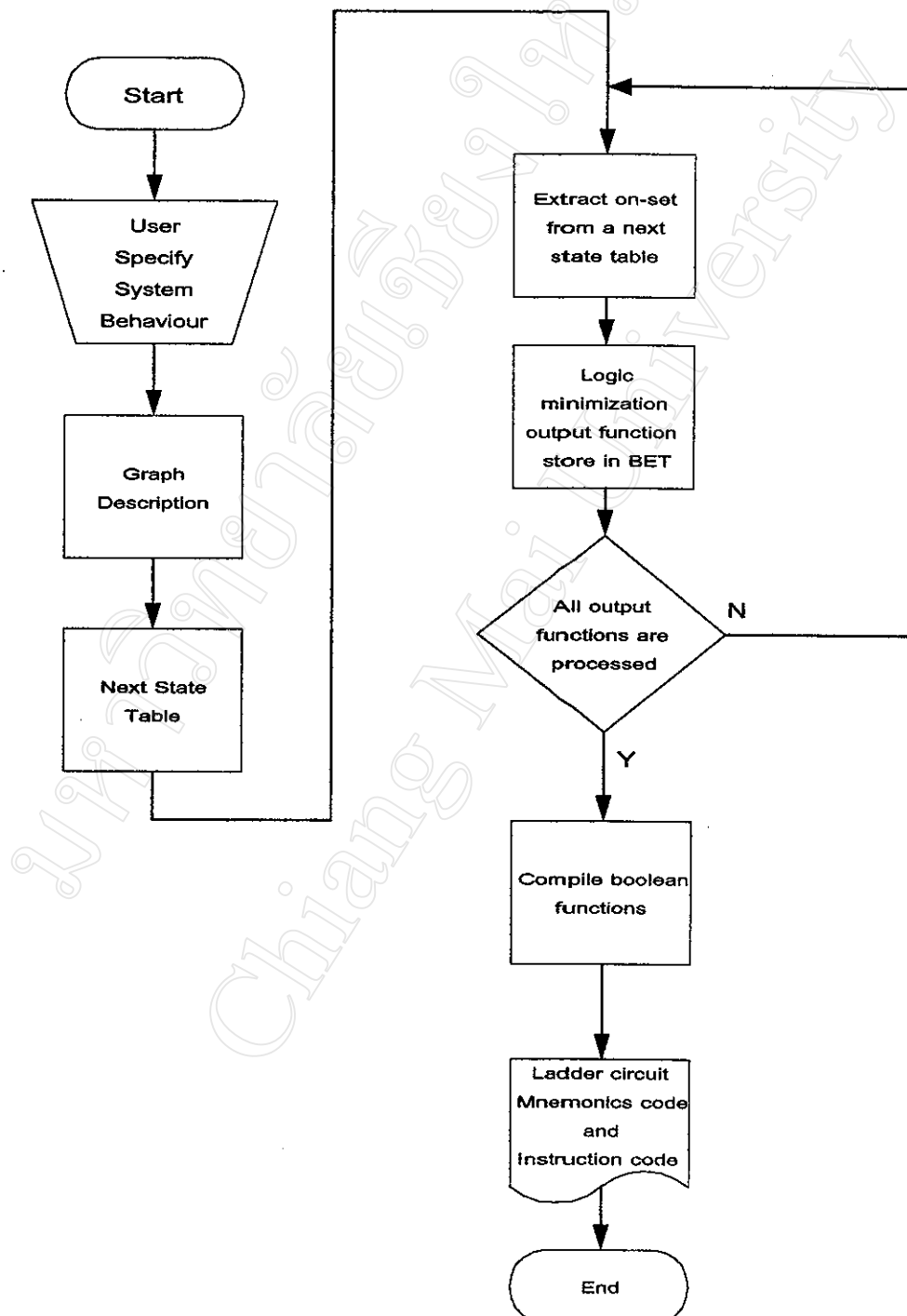
• สัญลักษณ์ของฟังก์ชันเสมือน (Virtual Function) การเป็นฟังก์ชันเสมือนของคลาสใด ๆ ที่สืบทอดคุณสมบัติไปนั้น วิธีการทำงานของฟังก์ชันอาจถูกกำหนดใหม่ได้ โดยไม่ต้องเหมือนกันกับคลาสที่ได้สืบทอดมา



รูปที่ 4.7 สัญลักษณ์ของฟังก์ชันเสมือน

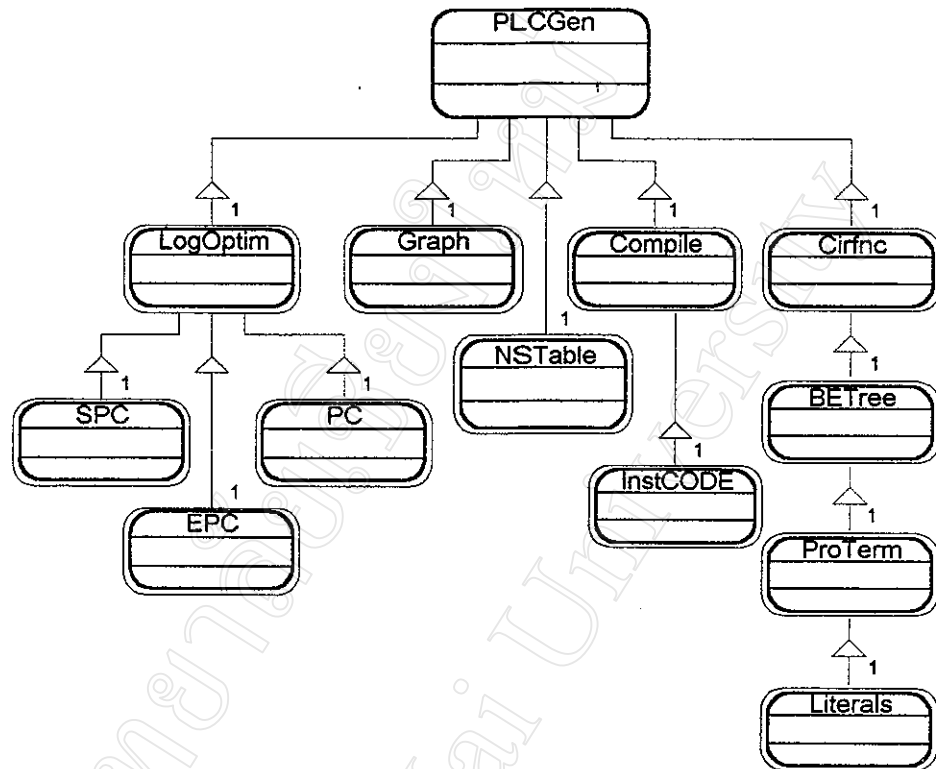
4.2 แนวความคิดในการออกแบบซอฟต์แวร์

โฟลว์ชาร์ตดังแสดงในรูปที่ 4.8 แสดงให้เห็นขั้นตอนการทำงานของซอฟต์แวร์ในการสังเคราะห์ วงจรการตรรก เพื่อนำไปแปลเป็นรหัสคำสั่งของพีแอลซีและแลคเคอร์โคอะแกรมต่อไป



รูปที่ 4.8 แสดงลำดับการทำงานของซอฟต์แวร์

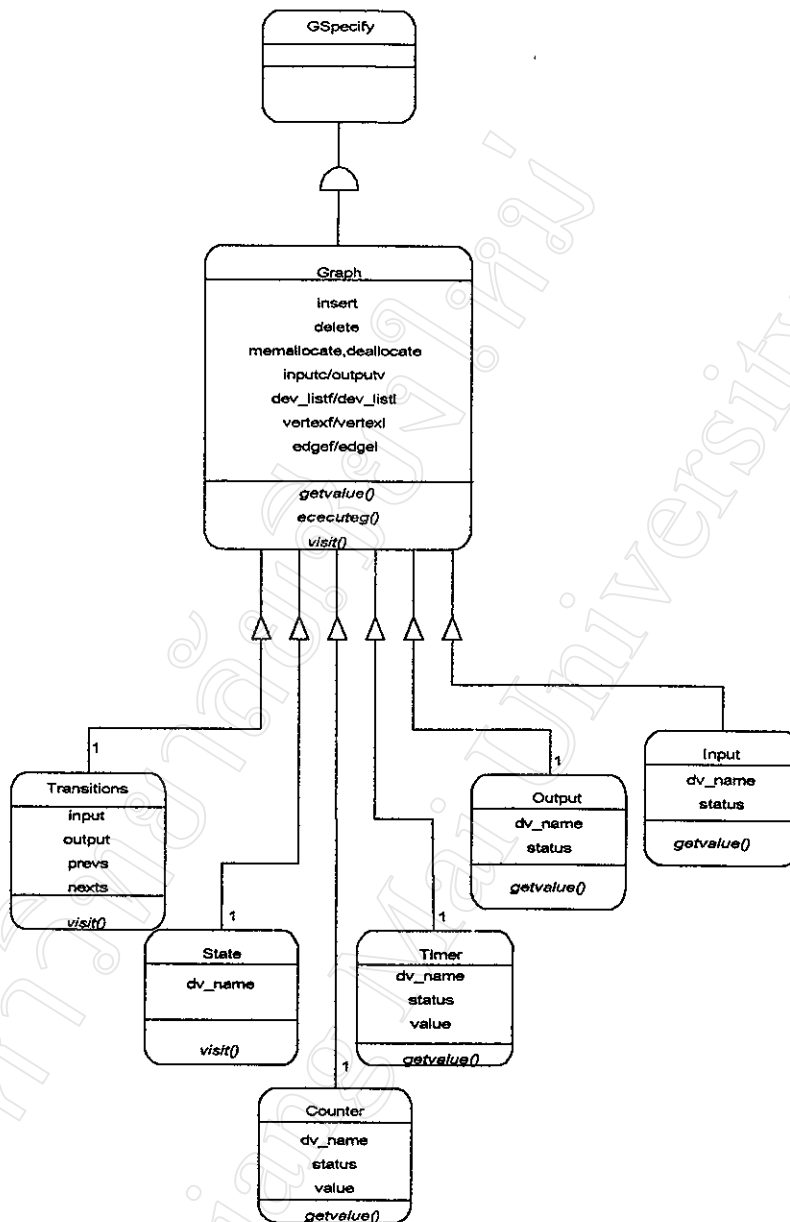
4.2.1 คลาสและวัตถุของตัวก่อกำเนิดโปรแกรมสำหรับพีแอลซี



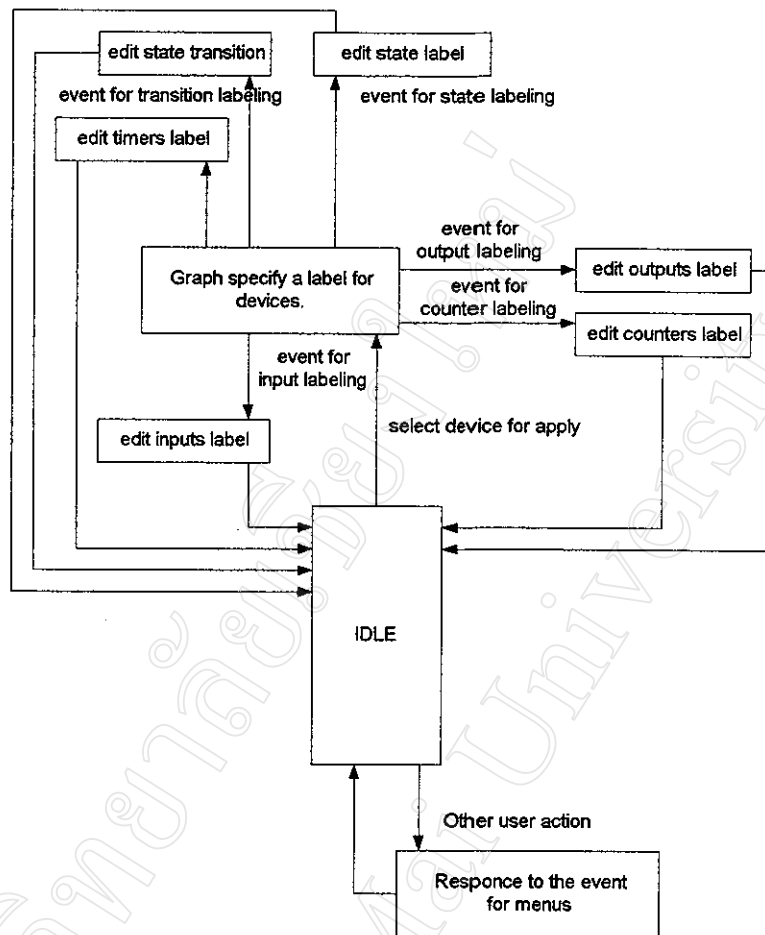
รูปที่ 4.9 ส่วนประกอบภายใต้การทำงานของซอฟต์แวร์

วัตถุแต่ละชิ้นออกแบบขึ้นตามการทำงานตามขั้นตอนการทำงานดังในรูปที่ 4.8 โดยวัตถุเหล่านี้ประกอบด้วย วัตถุ *Graph* ซึ่งทำหน้าที่ในระหว่างกาหนดพฤติกรรมของระบบควบคุม วัตถุ *NSTable* ทำหน้าที่ สร้างตารางสภาวะต่อไปจากการกำหนดพฤติกรรมการทำงาน วัตถุ *CirFunc* ทำหน้าที่สร้างสมการตรรกให้อยู่ในรูปของบูลีนเอ็กซ์เพรสชัน วัตถุ *LogOptim* ทำหน้าที่ลดรูปสมการตรรก และวัตถุ *Compile* ทำหน้าที่แปลจากสมการตรรกที่เก็บอยู่ในรูปของบูลีนเอ็กซ์เพรสชันเป็นรหัสนิโมนิกส์และรหัสคำสั่งของพีแอลซี

ส่วนของการกำหนดพฤติกรรมของระบบควบคุม จะเป็นส่วนที่ยอมให้ผู้ใช้สามารถตั้งชื่อของสัญญาณอินพุต สัญญาณเอาต์พุต ตัวคั้งเวลา และตัวนับได้เอง แต่มีความยาวของชื่อที่ตั้งตามที่กำหนดเอาไว้ ในส่วนของการกำหนดการเปลี่ยนสภาวะผู้ใช้สามารถเลือกสัญญาณอินพุตและสภาวะของการเปลี่ยนแปลงโดยใช้วิธีการเลือกจากเมนุรายการ ซึ่งส่วนการติดต่อกับผู้ใช้นี้เป็นส่วนที่น่าเอาความสามารถของตัวแปลภาษาที่มีรองรับอยู่แล้ว ดังนั้นในหัวข้อนี้จะกล่าวถึงเฉพาะส่วนที่เป็นโครงสร้างข้อมูล และการกำหนดคลาสของการกำหนดพฤติกรรมของระบบควบคุมเท่านั้น

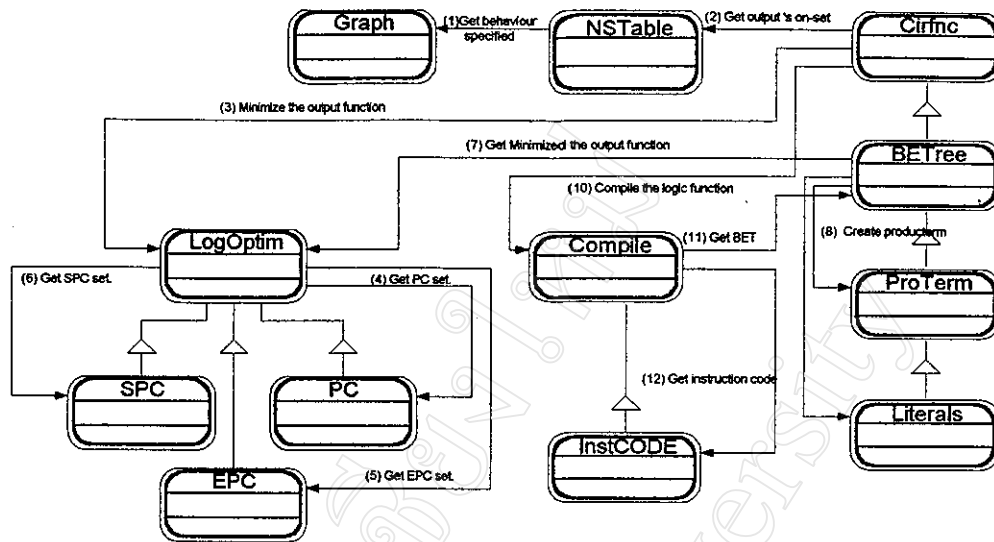


รูปที่ 4.10 คลาส Graph



รูปที่ 4.11 พฤติกรรมเมื่ออยู่ในโหมดของการสร้างกราฟ

ในขณะที่อยู่ในโหมดการกำหนดพฤติกรรมของระบบควบคุม เหตุการณ์ที่เข้ามาคือ การเลือกเมนูบนเมนูบาร์ และการเลือกกำหนดชื่อให้กับอุปกรณ์ต่าง ๆ ที่ต้องการใช้ โหมดการกำหนดกราฟนี้จะทำงานอยู่ภายใต้การทำงานของโปรแกรมหลักอีกที ซึ่งเป็นระบบการจัดการในตัวแปลภาษาของบอร์แลนด์ ซี (Borland C) รุ่น 3.1



รูปที่ 4.12 แสดงการติดต่อระหว่างวัตถุในการทำงานร่วมกัน

รูปที่ 4.12 แสดงการทำงานและการส่งผ่านการติดต่อระหว่างวัตถุ เมื่ออยู่ในระหว่างการสังเคราะห์วงจรและการแปลรหัสคำสั่งสำหรับพีแอลดี

4.2.2 ตารางสถานะต่อไป

ส่วนนี้เป็นกระบวนการสร้างตารางสถานะต่อจากกราฟการกำหนดพฤติกรรมของระบบควบคุม กระบวนการในส่วนนี้เป็นการรับเอาข้อมูลเข้ามาและทำการหาคำตอบส่งออกไปโดยไม่ตอบสนองต่อเหตุการณ์ใด ๆ ที่เข้ามาทั้งสิ้น กราฟอธิบายการเปลี่ยนแปลงสถานะของระบบควบคุมเป็นอินพุตและเอาต์พุตที่ได้คือตารางสถานะต่อไป การเข้าถึงทุกสถานะในกราฟอธิบายการเปลี่ยนแปลงสถานะโดยผ่านทุก ๆ การเปลี่ยนแปลงสถานะไปยังทุก ๆ สถานะ ใช้วิธีการเข้าถึงข้อมูลที่เรียกว่า Depth First Search ซึ่งทำงานภายใต้ฟังก์ชันการวนซ้ำ (Recursive Function)

4.2.2.1 วิธีการสร้างตาราง

การสร้างตารางสถานะต่อไปสามารถสร้างโดยใช้ข้อกำหนดในบทที่ 3 ในขณะที่เข้าถึงสถานะเริ่มต้นของกราฟอธิบายการเปลี่ยนแปลงสถานะ เราทำการเก็บสถานะเริ่มแรกของอินพุตและเอาต์พุตทุก ๆ ตัวไว้ใช้แทนค่าตัวแปรในการเปลี่ยนแปลงสถานะเพื่อหาคำของอินพุตและเอาต์พุตในสถานะต่อไป

Procedure NSTB(*Graph G*)

State = initial-state;

Layer = initial layer;

Enter the initial state in the initial layer;

Repeat

Execute state-transition label;

IF output not conflict **THEN****BEGIN**

Enter next-state of the path traversed in the current layer of next-state table;

IF next-specification state not processed **THEN**

State = Next-specification state ;

ELSE**BEGIN**

Go back to the neared ancestor with unprocessed child node;

State = state of next unprocessed child node;

END.

Layer = layer of parent node;

END. ELSE**BEGIN**

Back to the last specification state;

Layer = new layer;

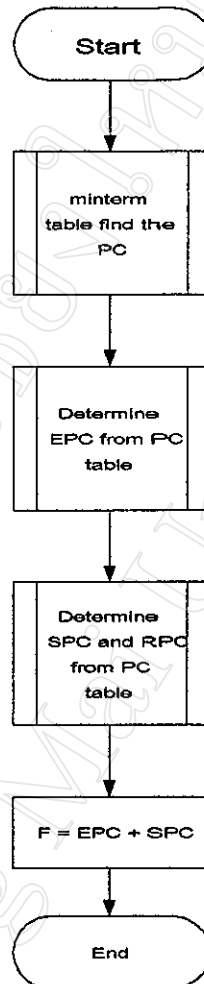
END.**END.** Until every node are processed;**End procedure.**

รูปที่ 4.13 แสดงกระบวนการสร้างตารางสถานะต่อไป

4.2.2.2 การลดทอนของสมการโดยใช้วิธีของ Quine-McCluskey

ออนเซตของฟังก์ชันของเอาต์พุตแต่ละตัวคือมินเทอม (สำหรับงานวิจัยนี้) ซึ่งจัดอยู่ในรูปของผลบวกของเทอมผลคูณ เพื่อนำมาลดรูปของสมการโดยใช้ Quine-McCluskey วิธีการได้กล่าวถึง

แล้วในบทที่ 3 และในบทนี้จะกล่าวถึงการใช้คอมพิวเตอร์แทนการสร้างตารางและหาคำตอบด้วยใช้มือ
วิธีการดำเนินการแสดงในรูปที่ 4.14



รูปที่ 4.14 แสดงวิธีการลดรูปสมการ โดยวิธี Quine-McCluskey

การค้นหา PC จากมินเทอมโดยการเปรียบเทียบระหว่างเซตของมินเทอมที่มีน้ำหนักต่างกัน 1 ระดับ และรวมทั้งสองเทอมเข้าด้วยกันถ้าสามารถรวมกันได้โดยนำผลของการรวมกันไปสร้างตารางใหม่ กระบวนการทำงานดังแสดงในรูปที่ 4.15

Procedure EPC(PC,M)

$P = \{p_1, p_2, \dots, p_r\}$; is a set of PC

$M = \{m_1, m_2, \dots, m_k\}$; is a set of minterm

Repeat

IF $m_i \in p_j \wedge m_i \notin p_1 p_2 \dots p_{j-1} p_{j+1} \dots p_r$ **THEN**

p_j is EPC

Until all M 's members are processed.

End procedure.

รูปที่ 4.15 แสดงกระบวนการเลือก EPC

Procedure SPC(M,PC,EPC)

$EPC = \{p_1 p_2 \dots p_i\}$;

$M_{epc} = P(EPC)$; is all minterms are selected by EPC set.

$C_{spc} = PC \cap EPC$;

Repeat

Begin

$c_i = p_i \cap C_{spc}$;

IF $p_i \subseteq EPC \cup c$ **THEN** p_i is RPC

ELSE p_i is SPC

END Until all p_i in C_{spc} are processed.

$Logic\ function = EPC + SPC$.

End procedure.

รูปที่ 4.16 แสดงกระบวนการเลือก RPC และ SPC

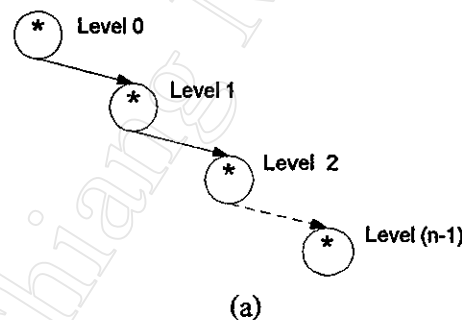
4.2.3 บูลีนเอ็กซ์เพรสชันทรี

การสร้างบูลีนเอ็กซ์เพรสชันทรี ในขั้นตอนนี้ของซอฟต์แวร์จะ ไม่มีการตอบสนองต่อเหตุการณ์ใด ๆ ที่เข้ามาในระบบเช่นเดียวกับกระบวนการสร้างตารางสภาวะต่อไป กระบวนการนี้มีอินพุตเป็นตารางสภาวะ และเอาต์พุตของกระบวนการนี้คือบูลีนเอ็กซ์เพรสชันทรีของเอาต์พุตแต่ละตัว กระบวน

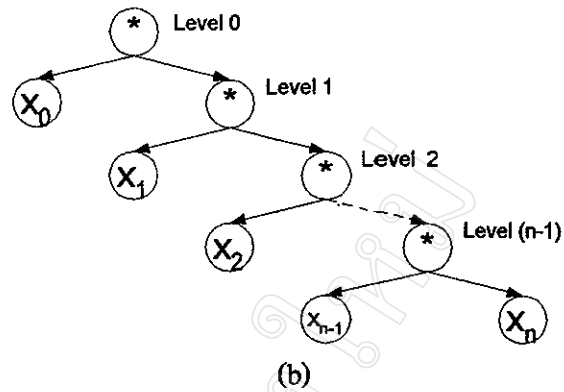
การสร้างจะเป็นไปตามนิยามในบทที่ 3 และกระบวนการดังรูปที่ 4.15 และ 4.16 กระบวนการสร้างทั้งสองขั้นตอนดังนี้

1) สร้างซับบูลีนเอ็กซ์เพรสชันทรีจากเทอมผลคูณแต่ละเทอม การสร้างซับบูลีนเอ็กซ์เพรสชันทรีจากเทอมผลคูณ ทำการสร้างเวอร์เท็กซ์ตัวดำเนินการที่มีคุณสมบัติเป็นตัวดำเนินการ *AND* จำนวนเท่ากับ $n-1$ เวอร์เท็กซ์เมื่อ n คือจำนวนตัวแปรทั้งหมดของเทอมผลคูณ แต่ละเวอร์เท็กซ์ขยายลำดับลงมาทางด้านขวาของเวอร์เท็กซ์ราก หลักจากนั้นสร้างเวอร์ตวแปรเป็นเวอร์เท็กซ์ลูกทางด้านซ้ายของเวอร์เท็กซ์ตัวดำเนินการทุก ๆ เวอร์เท็กซ์จนถึงเวอร์เท็กซ์สุดท้ายของเวอร์เท็กซ์ตัวดำเนินการซึ่งไม่มีเวอร์เท็กซ์ลูกทั้งด้านซ้ายและขวาจึงวางเวอร์เท็กซ์ตัวแปรทั้งสองด้านซึ่งจะครบทั้งหมดและสิ้นสุดการสร้างซับบูลีนเอ็กซ์เพรสชันทรี (กำหนดให้เวอร์เท็กซ์ตัวแปรทุกตัวชี้ไปที่เวอร์เท็กซ์ปลายทางอยู่แล้ว)

2) สร้างบูลีนเอ็กซ์เพรสชันทรีจากซับบูลีนเอ็กซ์เพรสชันทรี การสร้างบูลีนเอ็กซ์เพรสชันทรีจากซับบูลีนเอ็กซ์เพรสชันทรี ทำการสร้างเวอร์เท็กซ์ตัวดำเนินการที่มีคุณสมบัติเป็นตัวดำเนินการ *OR* จำนวนเท่ากับ $m-1$ เวอร์เท็กซ์เมื่อ m คือจำนวนเทอมผลคูณทั้งหมด แต่ละเวอร์เท็กซ์ขยายลำดับลงมาทางด้านซ้ายของเวอร์เท็กซ์ราก หลักจากนั้นนำซับบูลีนเอ็กซ์เพรสชันทรีเป็นเวอร์เท็กซ์ลูกทางด้านขวาของเวอร์เท็กซ์ตัวดำเนินการทุก ๆ เวอร์เท็กซ์จนถึงเวอร์เท็กซ์สุดท้ายของเวอร์เท็กซ์ตัวดำเนินการซึ่งไม่มีเวอร์เท็กซ์ลูกทั้งด้านซ้ายและขวาจึงวางซับบูลีนเอ็กซ์เพรสชันทรีทั้งสองด้านซึ่งจะครบทั้งหมดและสิ้นสุดการสร้างบูลีนเอ็กซ์เพรสชันทรี

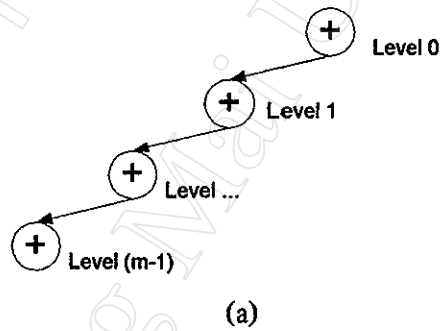


รูปที่ 4.17 กระบวนการสร้างซับบูลีนเอ็กซ์เพรสชันทรีจากเทอมผลคูณ

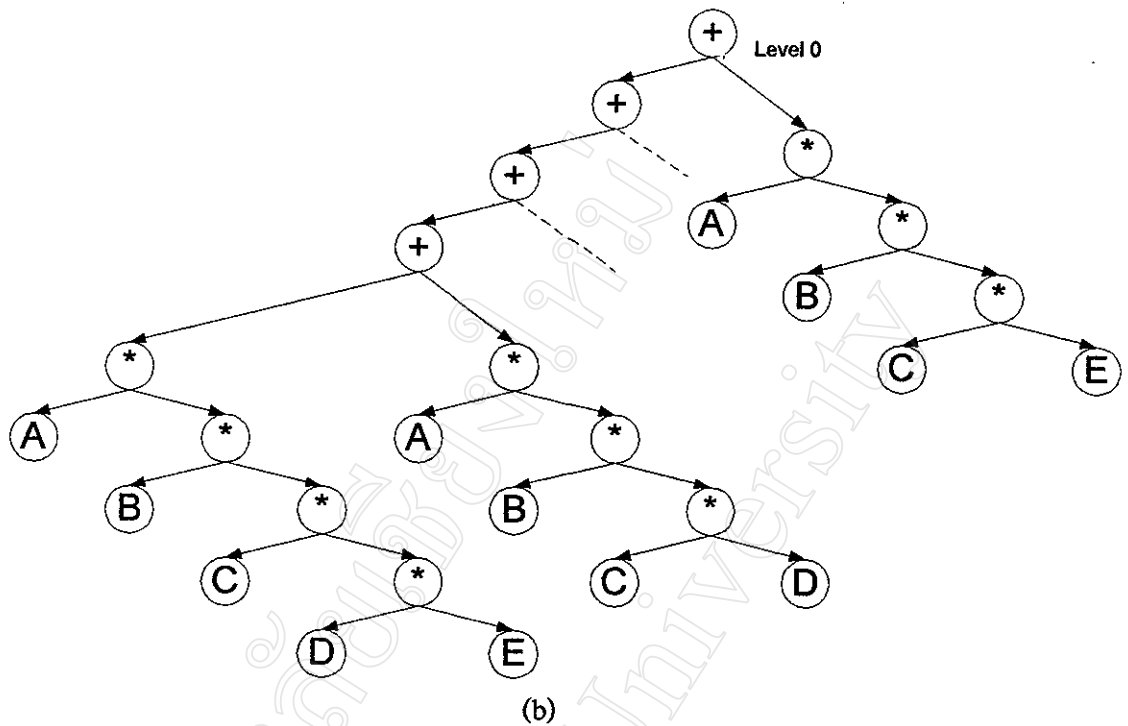


รูปที่ 4.17 (ต่อ) กระบวนการสร้างซับบูลีนเอ็กซ์เพรสชันทรีจากเทอมผลคูณ

- (a) แสดงการสร้างเวอร์เท็กซ์ตัวดำเนินการที่ขยายตัวลงทางด้านซ้าย
 (b) แสดงการติดตั้งเวอร์เท็กซ์ตัวแปรทางด้านขวาของเวอร์เท็กซ์ตัวดำเนินการ
 การ OR



รูปที่ 4.18 กระบวนการสร้างบูลีนเอ็กซ์เพรสชันทรีจากซับบูลีนเอ็กซ์เพรสชันทรี



รูปที่ 4.18 (ต่อ) กระบวนการสร้างบูลีนเอ็กเพรสชันทรีจากซัพบูลีนเอ็กเพรสชันทรี

(a) แสดงเวอร์เท็กซ์ตัวดำเนินการ OR ที่ขยายตัวไปทางซ้าย

(b) แสดงซัพบูลีนที่นำเข้าทางด้านขวาของเวอร์เท็กซ์ตัวดำเนินการ OR

4.2.4 การสร้างสมการตรรกหลายระดับ

วงจรตรรกหลายระดับจากสมการตรรกสองระดับจากความสัมพันธ์ $F=QD+R$ เราใช้วิธีตารางในการค้นหากลุ่มของ Q R และ D เทอมผลคูณดังตารางที่ 4.1 แสดงการค้นหาเพื่อแยก Q D และ R จาก เทอมผลคูณ $BCDE' + A'BCE + AC'DE' + A'B'DE'$

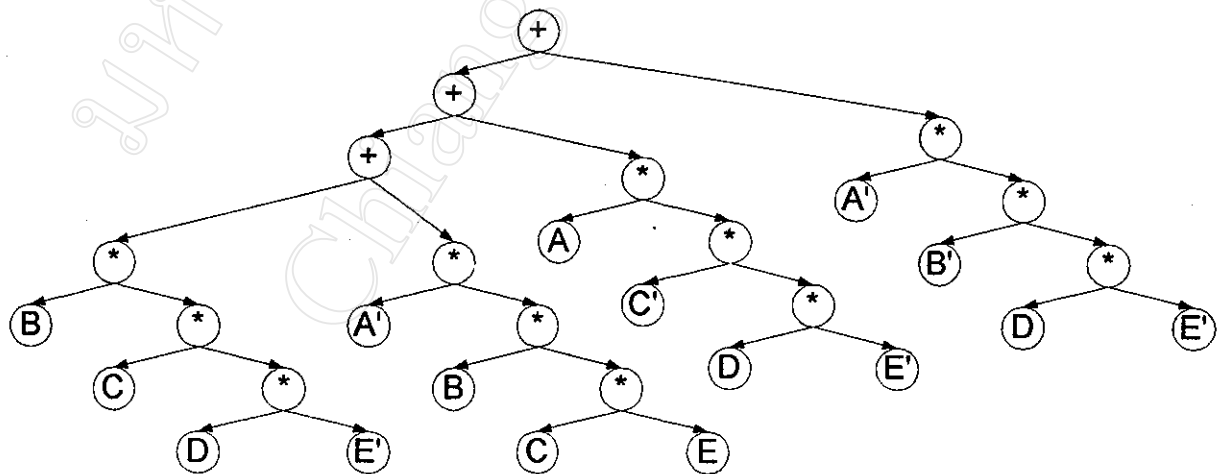
ตารางที่ 4.1 แสดงตัวอย่างการเลือกตัวแปรที่ต้องแยกออก

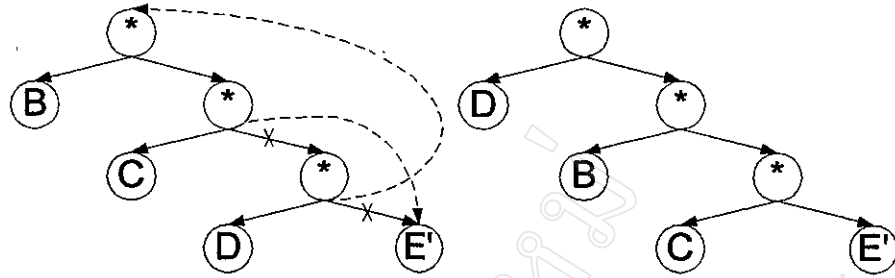
Literal(s)	Product term(s)				Weight
	$BCD.E'$	$A'BCE$	$AC'DE'$	$A'B'DE'$	
A	-	-	✓	-	1
B	✓	✓	-	-	2
C	✓	✓	-	-	2
D	✓	-	✓	✓	3

ตารางที่ 4.1 (ต่อ) แสดงตัวอย่างการเลือกตัวแปรที่ต้องแยกออก

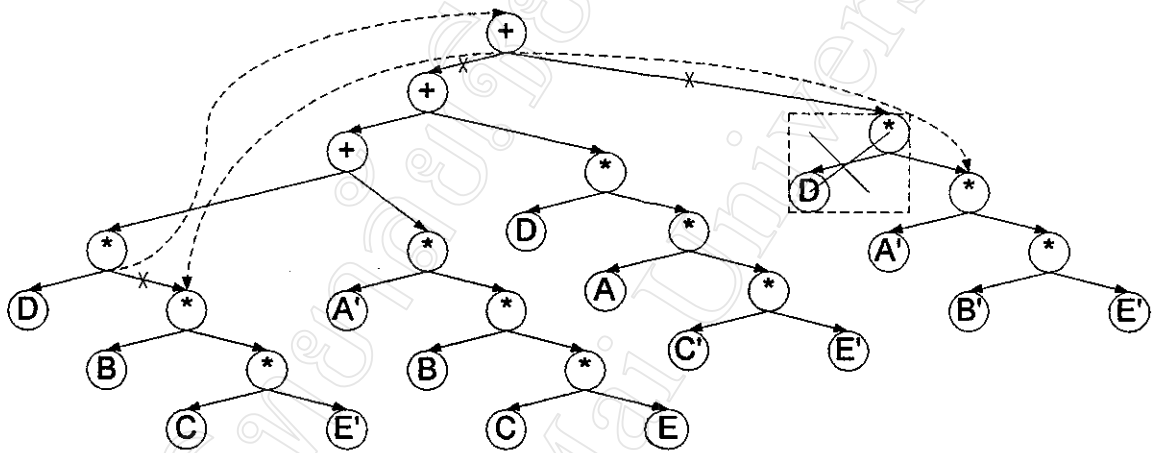
Literal(s)	Product term(s)				Weight
	BCD.E'	A'BCE	AC'DE'	A'B'DE'	
E	-	√	-	-	1
A'	-	√	-	√	2
B'	-	-	-	√	1
C'	-	-	√	-	1
D'	-	-	-	-	0
E'	√	-	√	√	3

จากตารางน้ำหนักของตัวแปรคือความถี่ของตัวแปรแต่ละตัวที่เป็นสมาชิกในเทอมผลคูณ จากตารางตัวแปร D และ E' มีค่าของน้ำหนักเท่ากันคือเท่ากับ 3 ถ้าทุกเทอมผลคูณที่ D เป็นสมาชิกคือทุกเทอมผลคูณที่ E' เป็นสมาชิก ดังนั้น D และ E' คือ Q และ ถ้าไม่เป็นเช่นนั้น D และ E' ตัวใดตัวหนึ่งเป็น Q ในตัวอย่าง Q คือ {D,E'} สำหรับ R คือเทอมที่ DE' ไม่เป็นสมาชิกอยู่คือเทอม A'BCE ดังนั้น $F=DE'(BC+AC'+A'B')+A'BCE$ และ D คือ {BC,AC',A'B'} ซึ่งจัดอยู่ในรูปของ ผลบวกของเทอมผลคูณคือ $BC+AC'+A'B'$

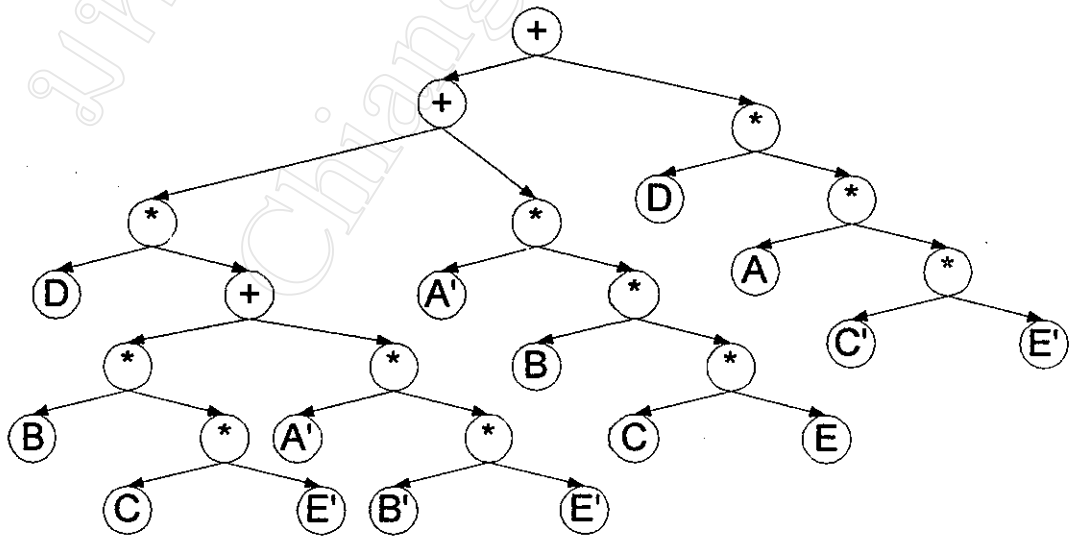
รูปที่ 4.19 แสดงมูลลินเอ็กซ์เพรสชันทรีของสมการ $BCDE'+A'BCE+AC'DE'+A'B'DE'$



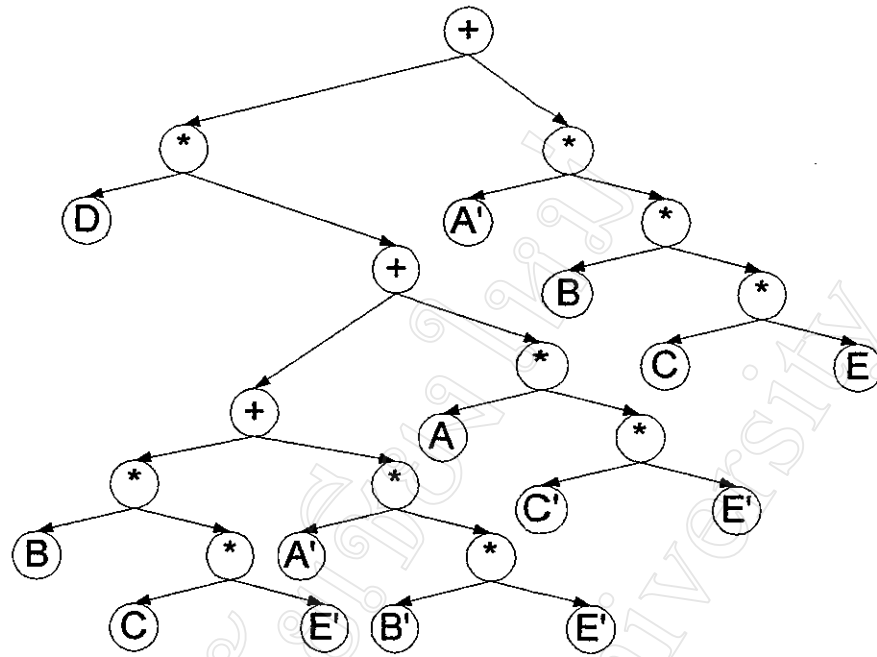
รูปที่ 4.20 แสดงการปรับโครงสร้าง



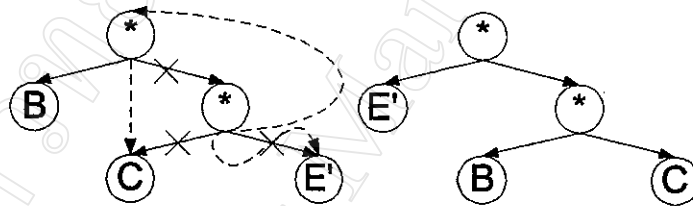
รูปที่ 4.21 แสดงทอมผลคูณที่ถูกเลือกเป็นทอมแรก โดยย้ายเวอร์เท็กซ์ตัวแปรมาไว้ทางด้านซ้าย



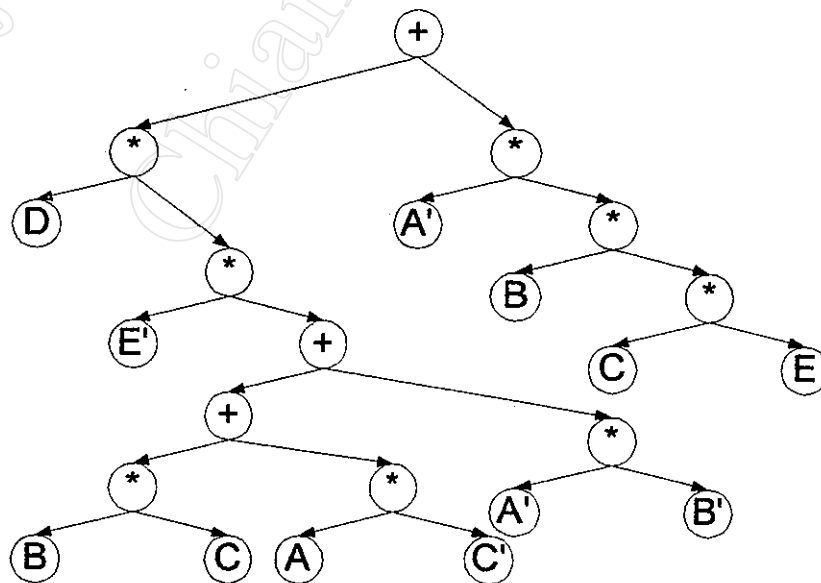
รูปที่ 4.22 แสดงการปรับโครงสร้าง



รูปที่ 4.23 แสดงการปรับโครงสร้าง



รูปที่ 4.24 แสดงการปรับโครงสร้าง



รูปที่ 4.25 แสดงโครงสร้างใหม่หลังการปรับโครงสร้างเสร็จแล้ว

4.2.5 วิธีการแปลคำสั่งมูลเงินจากมูลเงินอิเล็กทรอนิกส์พรสขันธ์

โครงสร้างข้อมูล มูลเงินอิเล็กทรอนิกส์พรสขันธ์หนึ่งโครงสร้างคือเอาต์พุตฟังก์ชันหนึ่งเอาต์พุตการค้นหาคำสั่งมูลเงินจากมูลเงินอิเล็กทรอนิกส์พรสขันธ์ใช้วิธีการทอ้งเข้าไปในทริแบบโพสออเคอร์ (Post Order) คือการเข้าถึงเวอร์เท็กซ์ทางด้านซ้ายสุดของโครงสร้างข้อมูลเป็นลำดับแรกและต่อมาจึงเข้าไปยังเวอร์เท็กซ์ทางด้านขวาของโครงสร้างข้อมูล สุดท้ายจึงกลับมายังรูตเวอร์เท็กซ์ เหมือนกันทุก ๆ โครงสร้างข้อมูลย่อยในมูลเงินอิเล็กทรอนิกส์พรสขันธ์ ตามนิยาม 3.1 ในบทที่ 3 มูลเงินอิเล็กทรอนิกส์พรสขันธ์ประกอบด้วยเวอร์เท็กซ์ทั้งหมด 3 เวอร์เท็กซ์คือ เวอร์เท็กซ์ตัวดำเนินการ เวอร์เท็กซ์ตัวแปร และ เวอร์เท็กซ์ปลายทางในการทอ้งเข้าไปในโครงสร้างข้อมูลมูลเงินอิเล็กทรอนิกส์พรสขันธ์นี้เราจะเข้าไปถึงแค่เวอร์เท็กซ์ตัวดำเนินการเท่านั้นแล้วทำการหาคำสั่งมูลเงินตามกฎในนิยาม 3.3 ในบทที่ 3 กระบวนการการแปลคำสั่งมูลเงินจากโครงสร้างข้อมูลมูลเงินอิเล็กทรอนิกส์พรสขันธ์