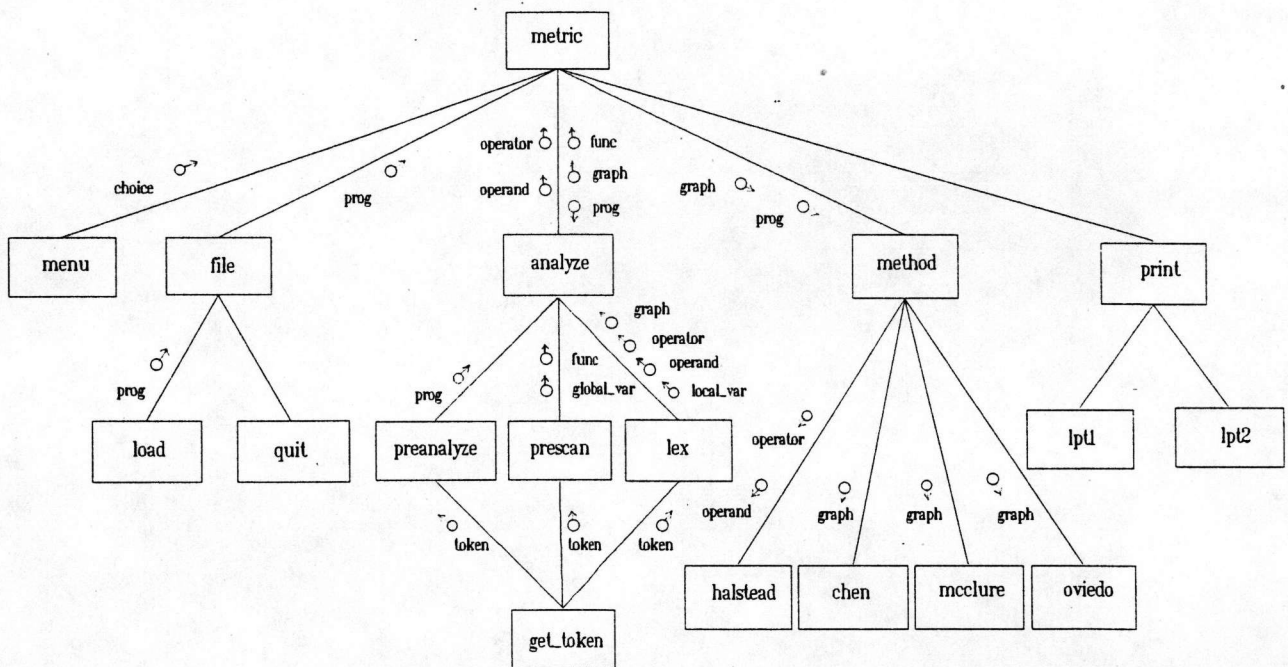


บทที่ 4

การออกแบบโปรแกรม (Program Design)

โปรแกรมวัดความซับซ้อน มีรายละเอียดการออกแบบทั้งหมดทั้งหมดดังนี้

ก. ผังโครงสร้าง (structure chart) ของระบบ



รูปที่ 4.1 แสดงผังโครงสร้าง ของโปรแกรมวัดความซับซ้อนในระดับที่ 1

### พจนานุกรมข้อมูล ของผังโครงสร้างในรูปที่ 4.1

ชื่อ	choice
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	ตัวเลือกการทำงานของโปรแกรม มีค่าในช่วง 1-4
ชื่อ	func
ชนิดข้อมูล	struct
ขนาดข้อมูล	40 ไบต์
อธิบาย	เก็บชื่อและตำแหน่งเริ่มต้นของฟังก์ชัน เพื่อใช้ในการเข้าถึงฟังก์ชันนั้น
ชื่อ	global_var
ชนิดข้อมูล	struct
ขนาดข้อมูล	36 ไบต์
อธิบาย	เก็บตัวแปรโกลบอลทั้งหมดของโปรแกรมนำมาไว้
ชื่อ	local_var
ชนิดข้อมูล	struct
ขนาดข้อมูล	36 ไบต์
อธิบาย	เก็บตัวแปรโลคอลทั้งหมดของโมดูลที่กำลังรัน
ชื่อ	operand
ชนิดข้อมูล	struct
ขนาดข้อมูล	40 ไบต์
อธิบาย	เก็บชื่อและจำนวนตัวถูกดำเนินการทั้งหมดของโมดูลที่กำลังรัน
ชื่อ	operator
ชนิดข้อมูล	struct
ขนาดข้อมูล	40 ไบต์
อธิบาย	เก็บชื่อและจำนวนตัวดำเนินการทั้งหมดของโมดูลที่กำลังรัน

ชื่อ	prog
ชนิดข้อมูล	pointer to char
ขนาดข้อมูล	4 ไบต์
อธิบาย	เป็นตัวชี้ไปยังตำแหน่งใดๆในโปรแกรมที่นำมาวัด

ชื่อ	token
ชนิดข้อมูล	array of char
ขนาดข้อมูล	80
อธิบาย	เก็บโทเคนที่ได้จากโปรแกรมที่นำมาวัดความซับซ้อน

รายละเอียดของโมดูล ผังโครงสร้างของระบบในรูปแบบที่ 4.1

ชื่อ	metric
อินพุต	โปรแกรมภาษาซีที่ต้องการวัดค่าความซับซ้อน
การทำงาน	เป็นฟังก์ชันหลักของโปรแกรมวัดความซับซ้อน ควบคุมการทำงาน รับค่า choice จากเมนูแล้วไปทำงานยังโมดูลที่เลือก
เอาต์พุต	ค่าความซับซ้อนของโปรแกรม

ชื่อ	menu
อินพุต	เลขที่และประเภทของเมนูที่ต้องการทำงาน
การทำงาน	1. สร้างเมนูหลักในแนวระดับและเมนูย่อยในแนวตั้ง 2. รับค่า choice จากหน้าจอแล้วส่งค่าให้ main
เอาต์พุต	ค่าทางเลือกการทำงานของโปรแกรมตัววัด

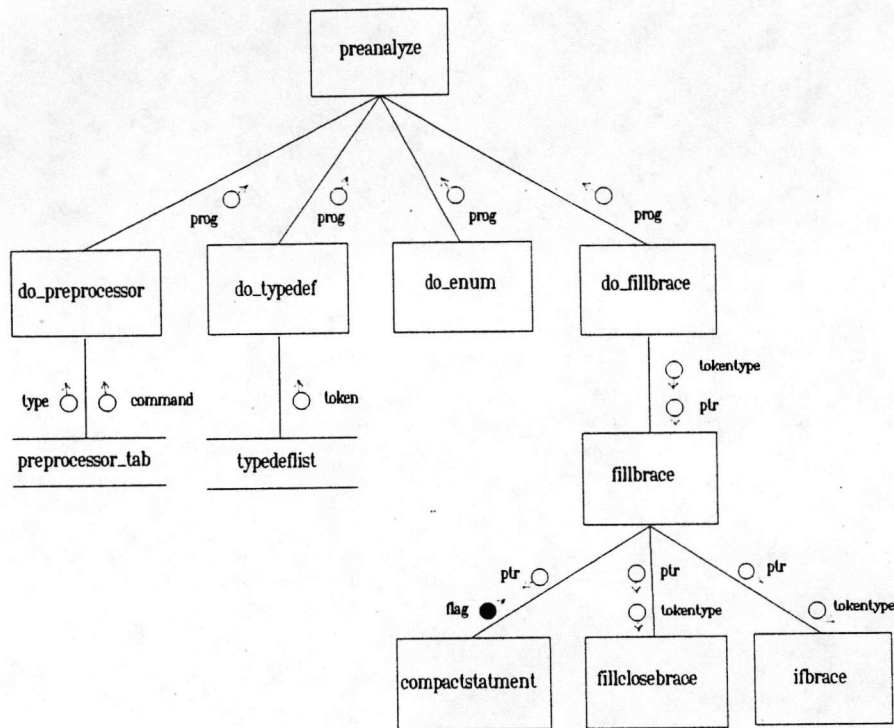
ชื่อ	load
อินพุต	ชื่อไฟล์
การทำงาน	รับชื่อและโหนดไฟล์ของโปรแกรมที่นำมาวัดเก็บไว้ และกรณีที่ทำ ชื่อไฟล์ไม่พบจะรายงานข้อผิดพลาดทางหน้าจอ
เอาต์พุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัดความซับซ้อน



ชื่อ	prescan
อินพุต	ptr ซึ่งเก็บตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	แบ่งโปรแกรมที่นำมาวัดเป็นโมดูล โดยเก็บชื่อและตำแหน่งของโมดูลไว้ในโครงสร้าง func และเก็บตัวแปรโกลบอลไว้ในโครงสร้าง global_var
เอาต์พุต	ชื่อและตำแหน่งของโมดูล และตัวแปรโกลบอล
ชื่อ	preanalyze
อินพุต	ตัวแปร ptr ซึ่งชี้ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	เรียกฟังก์ชันที่ทำหน้าที่แปลงโปรแกรมที่นำมาวัด ให้อยู่ในรูปแบบที่สามารถทำการวิเคราะห์ได้
เอาต์พุต	ตำแหน่งเริ่มต้นของโปรแกรมที่ผ่านการเปลี่ยนแปลงแล้ว
ชื่อ	lex
อินพุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	ทำหน้าที่เรียกฟังก์ชันที่ทำการวิเคราะห์ และให้ค่าเริ่มต้นกับตัวแปรโกลบอล และตัวแปรที่ส่งค่าระหว่างโมดูล
เอาต์พุต	กราฟกระแสการควบคุม, ค่าพารามิเตอร์ของตัววัดทั้ง 4 แบบ และตัวแปรโกลบอล
ชื่อ	method
อินพุต	-
การทำงาน	เรียกฟังก์ชันตัววัดทั้ง 4 แบบ
เอาต์พุต	-
ชื่อ	get_token
อินพุต	ptr ซึ่งเป็นตัวแปรชี้ตำแหน่งเริ่มต้นของโทเคน ภายในโปรแกรมที่นำมาวัด
การทำงาน	ตัดโปรแกรมที่นำมาวัดเป็นโทเคน ส่งโทเคนและประเภทของโทเคนนั้นให้กับฟังก์ชันที่เรียก
เอาต์พุต	โทเคน (token), ประเภทของโทเคน (tokentype) คำหลัก (keyword.item) และ กลุ่มของคำหลัก (keyword.group)

ข. ผังโครงสร้างของระบบ ในระดับที่ 2

1. ผังโครงสร้างระดับที่ 2 ในส่วน preanalyze



รูปที่ 4.2 แสดงผังโครงสร้างระดับที่ 2 ในส่วน preanalyze

พจนานุกรมข้อมูล ของผังโครงสร้างในรูปที่ 4.2

ชื่อ	brace
ชนิดข้อมูล	char
ขนาดข้อมูล	1 ไบต์
อธิบาย	เป็นตัวเก็บอักขระ '[' หรือ ']' เพื่อส่งค่าระหว่างฟังก์ชัน
ชื่อ	command
ชนิดข้อมูล	array of char
ขนาดข้อมูล	32 ไบต์
อธิบาย	ค่าคงที่อักขระในตาราง preprocessor_tab ซึ่งเก็บคำสั่ง พรีโพรเซสเซอร์ของภาษาซีทั้งหมดไว้

ชื่อ prog  
ชนิดข้อมูล pointer to char  
ขนาดข้อมูล 4 ไบต์  
อธิบาย ตัวแปรโกลบอล เป็นตัวชี้ไปยังตำแหน่งใดๆในโปรแกรมที่นำมาวัด

ชื่อ ptr  
ชนิดข้อมูล pointer to char  
ขนาดข้อมูล 4 ไบต์  
อธิบาย เก็บตำแหน่งปัจจุบันของโปรแกรมที่นำมาวัด ชั่วคราว

ชื่อ tokentype  
ชนิดข้อมูล int  
ขนาดข้อมูล 2 ไบต์  
อธิบาย ตัวเลขบอกชนิดของโทเคน มีค่าตามตารางที่ 3.1

ชื่อ type  
ชนิดข้อมูล int  
ขนาดข้อมูล 2 ไบต์  
อธิบาย ตัวเลขบอกประเภทของคำสั่งในตาราง preprocessor\_tab

รายละเอียดของโมดูล ฝังโครงสร้างในส่วน preanalyze

ชื่อ do\_fillbrace  
อินพุต ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด  
การทำงาน ฟังก์ชันหลักที่เรียกฟังก์ชันที่ทำหน้าที่แปลงโปรแกรมนำเข้า ในส่วนของ การบอกขอบเขตของฟังก์ชันด้วยวงเล็บปีกกา  
เอาต์พุต โปรแกรมที่ผ่านการเติมวงเล็บในคำสั่งควบคุมสายงานแล้ว

ชื่อ compactstatement  
อินพุต ตำแหน่งของคำสั่งควบคุมสายงานในโปรแกรม  
การทำงาน ตรวจสอบว่าคำสั่งควบคุมสายงานเป็น compactstatement หรือไม่ เช่น for (;;) หรือ while (x)  
เอาต์พุต ผลลัพธ์ว่าคำสั่งควบคุมสายงานนั้นเป็น compactstatement หรือไม่

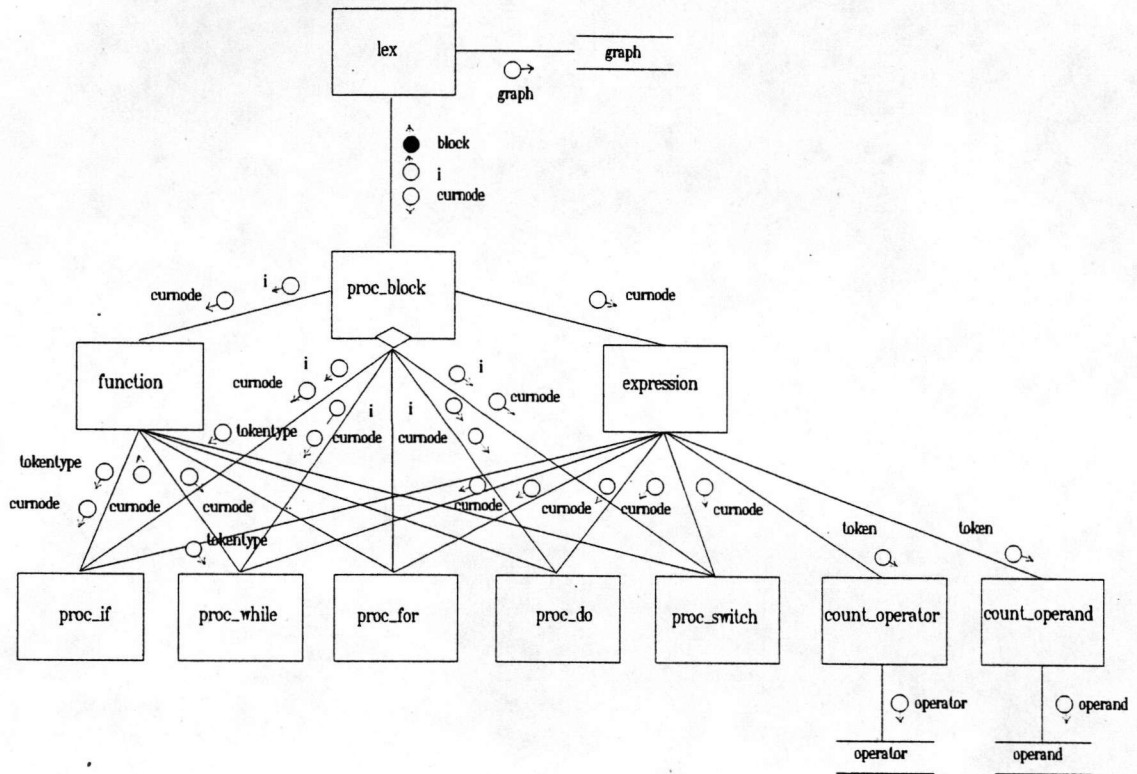
ชื่อ	ifbrace
อินพุต	ตำแหน่งในโปรแกรมหลังคำสั่ง if ที่ต้องการใส่เครื่องหมายวงเล็บ
การทำงาน	ตรวจสอบว่าคำสั่ง if นั้นมี else หรือไม่ และเติมเครื่องหมายวงเล็บเปิดหรือปิด
เอาต์พุต	โปรแกรมที่ผ่านการเติมวงเล็บหลังคำสั่ง if และ/หรือ else
ชื่อ	fillbrace
อินพุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	ทำหน้าที่แปลงโปรแกรมนำเข้า ในส่วนของกรบอกรขอบเขตของฟังก์ชันด้วยวงเล็บปีกกา
เอาต์พุต	โปรแกรมที่ผ่านการเติมวงเล็บปีกกา ในส่วนของคำสั่งควบคุมสายงาน
ชื่อ	fillclosebrace
อินพุต	ตำแหน่งในโปรแกรมที่ต้องการใส่เครื่องหมายวงเล็บปิด
การทำงาน	เติมเครื่องหมายวงเล็บปิดในโปรแกรม
เอาต์พุต	โปรแกรมที่ผ่านการเติมวงเล็บปิดแล้ว
ชื่อ	do_preprocessor
อินพุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	ทำหน้าที่แปลงโปรแกรมนำเข้า แทนที่โปรแกรมในส่วนที่ได้นิยามไว้ล่วงหน้าโดยพรีโพรเซสเซอร์
เอาต์พุต	โปรแกรมที่ผ่านการแปลงในส่วนของพรีโพรเซสเซอร์แล้ว
ชื่อ	do_typedef
อินพุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	เก็บชนิดของข้อมูลที่ผู้เขียนโปรแกรมได้กำหนดขึ้นไว้ในรายการเชื่อมโยง typedef_list
เอาต์พุต	รายการเชื่อมโยง typedef_list
ชื่อ	do_enum
อินพุต	ตำแหน่งเริ่มต้นของโปรแกรมที่นำมาวัด
การทำงาน	ทำหน้าที่แปลงโปรแกรมนำเข้า โดยแทนที่โปรแกรมในส่วนที่ได้นิยามไว้ล่วงหน้าโดยใช้คำสั่ง enumerate



เอาต์พุต

ตำแหน่งของโปรแกรมที่ผ่านการแปลงแล้ว

2. ฟังก์ชันโครงสร้างระดับที่ 2 ในส่วน lex



รูปที่ 4.3 แสดงฟังก์ชันโครงสร้างระดับที่ 2 ในส่วน lex

พจนานุกรมข้อมูล ของฟังก์ชันโครงสร้างในส่วน lex

ชื่อ block  
 ชนิดข้อมูล enum  
 ขนาดข้อมูล 2 ไบต์  
 อธิบาย เป็น flag เพื่อบอกว่าขณะนี้ตำแหน่งของตัวชี้โปรแกรมยังอยู่ในขอบเขตของบล็อกหรือไม่ มีค่าเป็น TRUE หรือ FALSE

ชื่อ cumode  
 ชนิดข้อมูล int  
 ขนาดข้อมูล 2 ไบต์  
 อธิบาย เลขที่ปัจจุบันของโหนด มีค่าอยู่ในช่วง 0-MAX\_NODE





ชื่อ	tokentype
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	ตัวเลขบอกชนิดของโทเคน มีค่าตามตารางที่ 3.1
ชื่อ	i
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	เลขที่ของโหนดที่ต่อกับโหนดปัจจุบัน มีค่าอยู่ในช่วง 0-MAX_NODE

รายละเอียดของโมดูล ผังโครงสร้างในส่วน lex

ชื่อ	proc_while
อินพุต	i และ curnode
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในโหนด while
เอาต์พุต	กราฟกระแสการควบคุม และค่าของพารามิเตอร์ในโหนด while

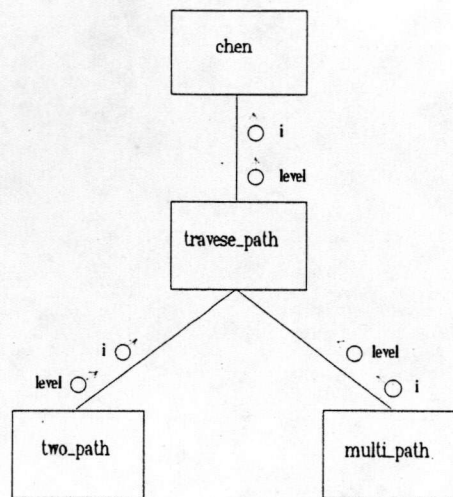
ชื่อ	proc_if
อินพุต	i และ curnode
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในโหนด if
เอาต์พุต	กราฟกระแสการควบคุม และค่าของพารามิเตอร์ในโหนด if

ชื่อ	proc_for
อินพุต	i และ curnode
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในโหนด for
เอาต์พุต	กราฟกระแสการควบคุม และค่าของพารามิเตอร์ในโหนด for

ชื่อ	proc_do
อินพุต	i และ curnode
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในโหนด do
เอาต์พุต	กราฟกระแสการควบคุม และค่าของพารามิเตอร์ในโหนด do

ชื่อ	proc_switch
อินพุต	i และ curnode
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในโหนด switch
เอาต์พุต	กราฟกระแสการควบคุม และค่าพารามิเตอร์ในโหนด switch
ชื่อ	proc_block
อินพุต	i, curnode และ block
การทำงาน	สร้างกราฟกระแสการควบคุม และหาค่าของพารามิเตอร์ในแต่ละบล็อก
เอาต์พุต	กราฟกระแสการควบคุม และค่าของพารามิเตอร์ในโมดูล
ชื่อ	expression
อินพุต	โหนดปัจจุบัน
การทำงาน	หาค่าของพารามิเตอร์ในคำสั่งของภาษาซี
เอาต์พุต	พารามิเตอร์ของตัววัดแบบต่างๆใน expression ใดๆ
ชื่อ	count_operand
อินพุต	item, tokentype, curnode และ ovido_type
การทำงาน	นับจำนวนตัวถูกดำเนินการ และเก็บค่าและตัวแปรไว้ในรายการ
เชื่อมโยง	
เอาต์พุต	operand_list
	ตัวถูกดำเนินการในรายการเชื่อมโยง operand_list
ชื่อ	count_operator
อินพุต	item
การทำงาน	นับจำนวนตัวดำเนินการ และเก็บค่าและตัวแปรไว้ในรายการ
	เชื่อมโยง operator_list
เอาต์พุต	ตัวดำเนินการในรายการเชื่อมโยง operator_list
ชื่อ	function
อินพุต	curnode และ tokentype
การทำงาน	หาค่าของพารามิเตอร์ในคำสั่งเรียกฟังก์ชันย่อยของโปรแกรม
เอาต์พุต	ค่าของพารามิเตอร์ของตัววัดแบบต่างๆ

### 3. ผังโครงสร้างระดับที่ 2 ในส่วน chen



รูปที่ 4.4 แสดงผังโครงสร้างระดับ 2 ในส่วน chen

พจนานุกรมข้อมูล ของผังโครงสร้างในส่วน chen

ชื่อ	i
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	เก็บเลขที่ของโหนดปัจจุบันที่กำลังคำนวณค่าความซับซ้อน
ชื่อ	level
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	เลขที่แสดงระดับของโครงสร้างการซ้อนใน (nesting structure)

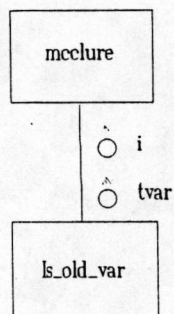
รายละเอียดของโมดูล ผังโครงสร้างในส่วน chen

ชื่อ	chen
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนตามวิธีของเซน
เอาต์พุต	ค่าความซับซ้อนของโปรแกรมตามวิธีของเซน

ชื่อ	two_path
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนตามวิธีของเซนในบริเวณย่อย (region) ของคำสั่งควบคุมสายงาน if-else, while, do, for
เอาต์พุต	ค่าความซับซ้อนของโปรแกรมตามวิธีของเซน
ชื่อ	multi_path
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนตามวิธีของเซนในบริเวณย่อย (region) ของคำสั่งควบคุมสายงาน switch
เอาต์พุต	ค่าความซับซ้อนของเซนในคำสั่งควบคุมสายงาน switch
ชื่อ	traverse_path
อินพุต	i และ level
การทำงาน	คำนวณหาค่า maximum path ของแต่ละบริเวณโดยคำนวณจากโหนดแรกไล่ไปตามมูลกษณะการเชื่อมโยงของโหนดจนถึงโหนดสุดท้าย
เอาต์พุต	ค่าความซับซ้อนของโปรแกรมตามวิธีของเซน



## 4. ผังโครงสร้างระดับที่ 2 ในส่วน McClure



รูปที่ 4.5 แสดงผังโครงสร้างระดับที่ 2 ในส่วน mcclure

หจนาุกรมข้อมูล ของผังโครงสร้างใน ส่วน mcclure

ชื่อ	i
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	เลขที่ของโหนดปัจจุบันที่กำลังตรวจสอบตัวแปรควบคุมอยู่

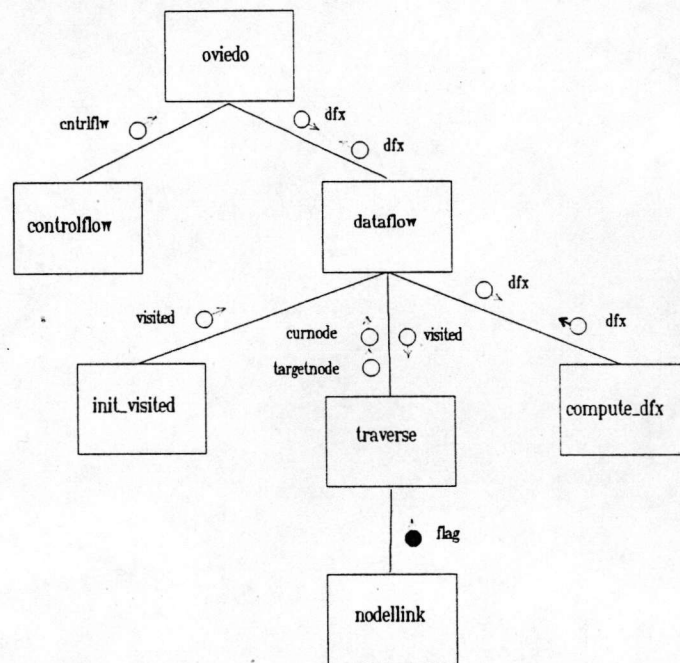
ชื่อ	tvar
ชนิดข้อมูล	array of char
ขนาดข้อมูล	32 ไบต์
อธิบาย	เก็บค่าตัวแปรควบคุมชั่วคราว เพื่อใช้ในการเปรียบเทียบกับตัวแปรควบคุมในโหนดอื่น

รายละเอียดของโมดูล ผังโครงสร้างในส่วน chen

ชื่อ	mcclure
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนตามวิธีของแมคคูลู
เอาต์พุต	ค่าความซับซ้อนของโปรแกรมตามวิธีของแมคคูลู

ชื่อ	Is_old_var
อินพุต	กราฟกระแสการควบคุม
การทำงาน	ตรวจสอบตัวแปรควบคุม ในคำสั่งควบคุมสายงานในโหนดปัจจุบัน ว่าเป็นตัวแปรที่เคยใช้ในคำสั่งควบคุมสายงานในโหนดก่อนหน้าหรือไม่
เอาต์พุต	ผลลัพธ์ว่าเป็น TRUE หรือ FALSE

#### 5. ผังโครงสร้างระดับที่ 2 ในส่วน oviedo



รูปที่ 4.6 แสดงผังโครงสร้างระดับที่ 2 ในส่วน oviedo

พจนานุกรมข้อมูล ของผังโครงสร้างในส่วน oviedo

ชื่อ	ctrlflw
ชนิดข้อมูล	int
ขนาดข้อมูล	2 ไบต์
อธิบาย	เก็บค่าความซับซ้อนเชิงกระแสการควบคุมของโอวิโด

ชื่อ dfx  
ชนิดข้อมูล int  
ขนาดข้อมูล 2 ไบต์  
อธิบาย เก็บค่าความซับซ้อนเชิงกระแสข้อมูลของโอวิโด

ชื่อ targetnode  
ชนิดข้อมูล int  
ขนาดข้อมูล 2 ไบต์  
อธิบาย ตัวเลขบอกเลขที่ของโหนดเป้าหมายที่ต้องการคำนวณค่าความซับซ้อน

ชื่อ visited  
ชนิดข้อมูล array of int  
ขนาดข้อมูล 2 ไบต์  
อธิบาย ตัวแปรชุดที่ใช้เก็บค่า flag เพื่อบอกว่าโหนดใดๆเคยได้รับการคำนวณค่าความซับซ้อนตามวิธีของโอวิโดหรือไม่

รายละเอียดของโมดูล ผังโครงสร้างในส่วน oviedo

ชื่อ oviedo  
อินพุต กราฟกระแสการควบคุม  
การทำงาน เรียกฟังก์ชันคำนวณหาค่าความซับซ้อนทั้งกระแสข้อมูล และ  
กระแสการควบคุมตามวิธีของโอวิโด  
เอาต์พุต ค่าความซับซ้อนของโปรแกรมตามวิธีของโอวิโด

ชื่อ controlflow  
อินพุต กราฟกระแสการควบคุม  
การทำงาน คำนวณหาค่าความซับซ้อนทางกระแสการควบคุมตามวิธีของโอวิโด โดย  
การหาจำนวนเส้นเชื่อม (edge) ของกราฟกระแสการควบคุม  
เอาต์พุต ค่าความซับซ้อนทางกระแสการควบคุมของโปรแกรมตามวิธีของโอวิโด

ชื่อ	dataflow
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนทางกระแสข้อมูลตามวิธีของโอวีโอด
เอาต์พุต	ค่าความซับซ้อนทางกระแสข้อมูลของโปรแกรมตามวิธีของโอวีโอด
ชื่อ	traverse
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนตามวิธีของโอวีโอด โดยคำนวณจากโหนดแรกไล่ไปตามลักษณะการเชื่อมโยงของโหนดจนถึงโหนดสุดท้าย
เอาต์พุต	-
ชื่อ	compute_dfx
อินพุต	กราฟกระแสการควบคุม
การทำงาน	คำนวณหาค่าความซับซ้อนทางกระแสข้อมูลตามวิธีของโอวีโอด โดยค่า dfx ย่อยในแต่ละโหนด คือจำนวนตัวแปรที่แฝงเฉพาะถิ่นของแต่ละโหนดที่ได้รับการนิยามมาก่อน
เอาต์พุต	ค่าความซับซ้อนทางกระแสข้อมูลของโปรแกรมตามวิธีของโอวีโอด
ชื่อ	node_link
อินพุต	กราฟกระแสการควบคุม
การทำงาน	ตรวจสอบการเชื่อมโยงระหว่างโหนด 2 โหนด
เอาต์พุต	ผลตรวจสอบการเชื่อมโยงระหว่างโหนด 2 โหนด ว่าเป็น TRUE หรือ FALSE
ชื่อ	init_visited
อินพุต	ตัวแปรชุด visited
การทำงาน	ให้ค่าเริ่มต้นกับตัวแปรชุด visited เป็น TRUE
เอาต์พุต	ตัวแปรชุด visied ที่มีค่าเริ่มต้น TRUE