

บทที่ 3

ไบซัน

ในงานวิจัยนี้เลือกใช้ภาษาไบซัน (Bison) ซึ่งเป็นตัวสร้างการแจงประโยคชนิดหนึ่ง ในการวิเคราะห์ไวยากรณ์ของภาษาต้นทาง และใช้ภาษา C ในการเขียนฟังก์ชันการเรียงลำดับคำ และการเพิ่มคำต่างๆ ที่จำเป็นในการแปล

```
% {  
  (1) .... ใ้ค้ดภาษา C  
% }  
(2) .... กำหนด non-terminal และ ลำดับ precedence ของ operator ต่างๆ  
%%  
(3) .... กำหนดโครงสร้างของไวยากรณ์ และการทำงาน  
%%  
(4) .... ฟังก์ชันภาษา C
```

รูปที่ 3.1 โครงสร้างโปรแกรมภาษาไบซัน

ส่วนที่ (1) ของภาษา C ที่เกี่ยวกับ header ไฟล์ และการนิยามความหมายด้วย #define

ส่วนที่ (2) การกำหนด terminal (Token), non-terminal และ ลำดับ precedence ของ operator ต่างๆ

ส่วนที่ (3) กำหนดโครงสร้างของไวยากรณ์ที่ต้องการ

ส่วนที่ (4) เขียนฟังก์ชันต่างๆ ที่จำเป็น

3.1 ไวยากรณ์ของภาษาไบซัน

- 1) ใช้ “:” ขึ้นกลางระหว่าง terminal, nonterminal กับ กฎ
- 2) กฎสามารถกระจายย่อยได้ โดยใช้เครื่องหมาย “|” กั้นระหว่างกฎ
- 3) หลังกฎ หรือกฎย่อยสุดท้ายจะต้องลงท้ายด้วย “;”

```
If_stmt : IF '(' exp ')' stmt // ประโยค if จะมี else หรือ ไม่มีก็ได้  
        | IF '(' exp ')' stmt ELSE stmt  
        ;
```

รูปที่ 3.2 ตัวอย่างไวยากรณ์ของภาษาไบซัน

4) เนื้อหาของ action ที่เป็นตัวกำหนดการทำงานจะถูกเขียนไว้ด้านหลังเสมอ

exp : exp '+' term	{ \$\$ = \$1 + \$3; }
--------------------	-----------------------

รูปที่ 3.3 เนื้อหาส่วน action

จากตัวอย่าง \$1 หมายถึง ค่าของสัญลักษณ์ตัวที่ 1 คือ exp ส่วน \$2 หมายถึงค่าของสัญลักษณ์ตัวที่ 3 คือ term และ \$\$ หมายถึง ค่าสุดท้ายที่จะเกิดขึ้นใน action ตัวนั้นๆ ในตัวอย่างนี้คือ ผลลัพธ์ที่เกิดจาก \$1+\$3

3.2 สัญลักษณ์ที่ประกาศในโปรแกรมภาษาไบซัน

- 1) สัญลักษณ์ที่ใช้แสดง Token (terminal)
- 2) ประกาศประเภทของตัวแปรว่าเป็น จำนวนเต็ม ทศนิยม หรือ สตริง ฯลฯ
- 3) Operator และลำดับของ precedence
- 4) สัญลักษณ์แสดงจุดเริ่มต้นของไวยากรณ์

3.3 การประกาศสัญลักษณ์ที่เป็น Token (terminal)

Token สามารถประกาศได้ดังนี้

% token	สัญลักษณ์
---------	-----------

รูปที่ 3.4 การประกาศสัญลักษณ์ Token

% token	IF ELSE WHILE
---------	---------------

รูปที่ 3.5 ตัวอย่างการประกาศสัญลักษณ์ Token

มีความหมายว่า IF, ELSE และ WHILE เป็น token ที่จะนำไปใช้ในโปรแกรมต่อ

3.4 การกำหนดประเภทของตัวแปร (Data Type)

ในภาษาไบซัน ประเภทของตัวแปรต่างๆ จะถูกกำหนดไว้ใน YYSTYPE ซึ่งมีค่า default เป็น int ในกรณีที่ไวยากรณ์นั้นมีตัวแปรหลายประเภท เช่น มีทั้งตัวเลขจำนวนเต็ม เลขทศนิยม และ สตริง เราสามารถนิยาม Data type ใหม่ได้โดยใช้ union ซึ่งมีโครงสร้างดังนี้

```
%union {
  Data_type  สัญลักษณ์ที่ใช้แทน
  .....
}
```

รูปที่ 3.6 การกำหนดประเภทของตัวแปร

ในกรณีที่ต้องการบอก Data type ของ Token ประกาศได้ดังรูปที่ 3.7

```
%type < สัญลักษณ์ที่ใช้แทน data type> Token
```

รูปที่ 3.7 การประกาศ Data type

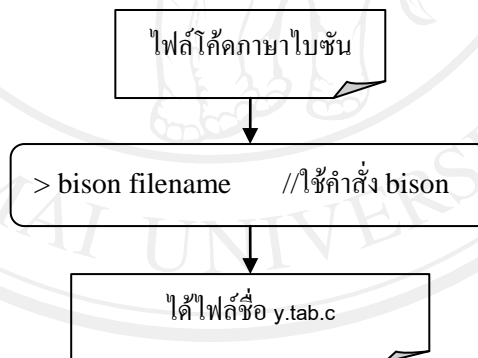
3.5 การประกาศจุดเริ่มต้นของไวยากรณ์

การวิเคราะห์ไวยากรณ์จะเริ่มต้นจากสัญลักษณ์ที่ประกาศเป็นจุดเริ่มต้น

```
%start <สัญลักษณ์แสดงจุดเริ่มต้น>
```

รูปที่ 3.8 การประกาศจุดเริ่มต้นของไวยากรณ์

3.6 วิธีกรใช้คำสั่ง bison



รูปที่ 3.9 การเรียกใช้คำสั่ง bison