

## วิชาการแปลภาษาโปรแกรม

### Symbol table สำหรับคอมไพเลอร์ CSubset

ในการทำ semantics analysis หรือการวิเคราะห์เกี่ยวกับความหมายของโปรแกรมนั้น เราจำเป็นต้องมีการเก็บข้อมูลเกี่ยวกับ ตัวแปร หรือ type หรือ ฟังก์ชัน เพื่อประโยชน์ในการทำ semantics analysis และ symbol table เป็นองค์ประกอบสำคัญในคอมไพเลอร์ที่จะมาทำหน้าที่ตรงนี้ ต่อไปเราจะแนะนำการสร้าง symbol table ในคอมไพเลอร์ CSubset

เริ่มจากการสร้าง struct ที่ใช้บรรจุข้อมูลของสัญลักษณ์ (เรียกสัญลักษณ์เหล่านี้ว่า node) และข้อมูลของ type ดังต่อไปนี้

```
/* mode & class */
enum {CSGVar, CSGConst, CSGFld, CSGTyp, CSGProc, CSGSProc, CSGAddr,
CSGInst};

/* form */
enum {CSGInteger, CSGBoolean, CSGArray, CSGStruct};

typedef struct CSGTypeDesc *CSGType;
typedef struct CSGNodeDesc *CSGNode;

typedef struct CSGTypeDesc {
    signed char form; // integer, array, struct
    CSGNode fields; // linked list of the fields in a struct
    CSGType base; // base type (array element type)
    int size; // total size of the type
    int len; // number of array elements
} CSGTypeDesc;

typedef struct CSGNodeDesc {
    signed char class; // Var, Const, Field, Type, Proc, SProc, Inst
    signed char lev; // 0 = global, 1 = local
    CSGNode next; // linked list of all objects in same scope
    CSGNode dsc; // Proc: link to procedure scope (head)
    CSGType type; // type
    CSSIdent name; // name
    long long val; // Const: value; Var: address; SProc: number;
                //Type: size
    char op; // operation of instruction
    CSGNode x, y; // the two operands
    CSGNode prv, nxt; // previous and next instruction
    int line; // line number for printing purposes
    CSGNode true, false; // Jmp: true and false chains; Proc: entry
    point;
} CSGNodeDesc;
```

พิจารณา declaration ในภาษา CSubset ดังต่อไปนี้

```
struct R {  
    long f, g;  
} S;  
long x, a[10];  
const long q = 123;
```

ข้อมูลเกี่ยวกับ declaration ด้านบน จะถูกนำมาสร้างเป็น symbol table ของ CSubset คอมไพเลอร์ได้ตามด้านล่างนี้



