

Lab03.Semantic Analysis

담당조교 : 박성환



Semantic Analysis

- 실습 목표
 - 컴파일러의 기호 테이블 구현
 - constant symbol을 테이블에 삽입 하는것 까지 실습 목표
 - 이후 구현 사항은 과제로 출제

Symbol Types

- Symbol은 다음과 같은 속성을 가짐
 - type
 - variable, function, parameter, constant
 - name
 - 심볼의 이름을 나타냄
 - value
 - 심볼이 const type이라면 해당 심볼의 값을 나타냄
 - parameter count
 - 심볼의 타입이 함수라면 받는 인자의 개수가 저장됨
 - Address
 - 해당 심볼의 위치
- Address는 다음과 같은 속성을 가짐
 - level
 - 현재 심볼이 속한 영역의 중첩 수준을 나타냄
 - address
 - 속한 영역에서의 주소를 나타냄

Symboltable Functions

- **blockBegin**
- **blockEnd**
- enterTfunc
- enterTval
- enterTpar
- **enterTconst**
- searchT
- printTable

Function Details

- `blockBegin`
 - 블록의 시작시 현재 중첩 수준을 관리하며 이에 따라 초기화 수행
- `blockEnd`
 - 블록이 종료될 때 중첩 수준을 관리
- `enterTfunc`
- `enterTval`
- `enterTpar`
- `enterTconst`
 - 각 심볼의 타입에 따라 symbol table에 symbol 삽입
- `searchT`
 - `symbol name`과 `symbol kind`를 통해 심볼 탐색
 - 탐색에 실패할 경우 에러 출력

Data Structure

- Symbol type

- hw_table.h 파일 내에 열거형을 활용해 타입 정의

```
1  /*****hw_table.h*****/
2
3  typedef enum kindT {
4      varId, funcId, parId, constId
5  }KindT;
6
```

- Symbol struct

- 구조체 활용해 hw_table.c 파일내에 Symbol의 구조 정의

```
7  typedef struct symbol {
8      KindT type;
9      char name[MAXNAME];
10     int val;
11     int level;
12     int addr;
13 }Symbol;
```

- 이외 필요한 변수들 선언

```
15 Symbol symbolTable[MAXTABLE];
16 int tIndex = 0;
17 int cLevel = -1;
18 int localAddr = 0;
```

blockBegin

- 블록의 시작을 알리는 함수
 - 현재 블록의 level과 주소를 설정

```
21 void blockBegin()  
22 {  
23     if(cLevel == -1) {  
24         tIndex = 0;  
25     }  
26     localAddr = 0;  
27     cLevel ++;  
28  
29     return ;  
30 }
```

- 블록 시작시 해당 함수 수행하기 위해 yacc file 수정

```
program : { blockBegin(FIRSTADDR); }  
        block '.'  
        ;
```

blockEnd

- 블록의 레벨을 낮추어줌

```
32 void blockEnd( )  
33 {  
34     cLevel --;  
35  
36     return ;  
37 }
```

- yacc file 수정

```
25 block      :      { }  
26           declList      { }  
27           statement      { blockEnd( ); }  
28
```


enterTconst

- constant type의 심볼을 테이블에 저장

```
51 int enterTconst(char *id, int v)
52 {
>> 53     strcpy(symbolTable[tIndex].name, id);
54     symbolTable[tIndex].type = constId;
55     symbolTable[tIndex].val = v;
56     return tIndex++;
57 }
```

- yacc file 수정

```
42 numberList : IDENT EQ NUMBER      { enterTconst($1, $3); }
43           | numberList COMMA IDENT EQ NUMBER
44                               { enterTconst($3, $5); }
45           ;
```

과제 요구 사항

- 심볼 테이블의 생성 및 출력
 - 심볼의 인덱스와 속성값들을 출력할 수 있어야함
 - 해당 심볼의 레벨과 주소값을 출력할 수 있어야함
 - 해당 레벨에서 주소는 0부터 시작해 1씩 증가
 - 함수와 변수는 각각 다른 주소를 가짐
 - 함수 파라미터의 경우 음수 값의 주소를 가짐
- 심볼 테이블에서 필요한 심볼 검색
 - 검색 실패시 에러 처리
 - 검색된 심볼에대한 정보 출력
 - Index, Type, Id

Result

```
Index : 1      kind : par      Id:x
Index : 2      kind : par      Id:y
Index : 4      kind : var      Id:b
Index : 4      kind : var      Id:b
Index : 5      kind : var      Id:c
Index : 3      kind : var      Id:a
Index : 3      kind : var      Id:a
Index : 4      kind : var      Id:b
Index : 5      kind : var      Id:c
Index : 7      kind : par      Id:x
Index : 8      kind : par      Id:y
Index : 9      kind : var      Id:a
Index : 10     kind : var      Id:b
Index : 11     kind : var      Id:c
Index : 12     kind : const     Id:m
Index : 13     kind : const     Id:n
Index : 14     kind : var      Id:x
Index : 15     kind : var      Id:y
Index  kindT   name    value  level  addr  pars
[0]    func   multiply          0      0      2
[1]    par    x          1      -2
[2]    par    y          1      -1
[3]    var    a          1      0
[4]    var    b          1      1
[5]    var    c          1      2
[6]    func   add          0      1      2
[7]    par    x          1      -2
[8]    par    y          1      -1
[9]    var    a          1      0
[10]   var    b          1      1
[11]   var    c          1      2
[12]   const  m          7
[13]   const  n         85
[14]   var    x          0      0
[15]   var    y          0      1
```

```
Index : 1      kind : par      Id:x
Index : 2      kind : par      Id:y
Index : 4      kind : var      Id:b
Index : 4      kind : var      Id:b
Index : 5      kind : var      Id:c
Index : 3      kind : var      Id:a
Index : 3      kind : var      Id:a
Index : 4      kind : var      Id:b
Index : 5      kind : var      Id:c
Index : 7      kind : const     Id:n
undef
Can't find symbol (z, var) in the table
Index : 6      kind : const     Id:m
Index : 7      kind : const     Id:n
Index : 8      kind : var      Id:x
Index : 9      kind : var      Id:y
Index : 8      kind : var      Id:x
Index : 9      kind : var      Id:y
```

과제 안내 사항

- 제출 마감 : 11월 14일 23시 59분
- 배포된 파일을 작성하여 Lab03_<학번>.tar.gz 형태로 압축하여 제출할 것
 - 제출 전 make clean 실행하여 소스 코드만 제출 할 것
- Makefile 활용하여 간단하게 compile 가능
 - make <- 실행 파일 생성
 - make clean <- 컴파일 과정에서 생성된 파일 삭제