

Chapter 4 ABSTRACT SYNTAX

4.2

问题的思考【仅供参考】:

- 1、根据第三章的递归下降的分析方法（P47，P89）写出分析程序。
- 2、将程序 4.4 中定义的文法动作提取出来，嵌入到递归下降的分析程序中，嵌入的方法有三种：分析程序的返回值、函数的附加功能和两者融合的方式（P88，P89）。

【其中：S:stmt L:exps E:exp B:PLUS|MINUS|TIMES|DIV】

```
enum token{ID,INT,PLUS,MINUS,TIMES,DIV, ASSIGN,PRINT,LPAREN,RPAREN,
SEMICOLON, COMMA};
```

```
union tokenval{string id;int num;};
```

```
enum token tok;
```

```
union tokenval tokval;
```

```
typedef struct table *Table_;
```

```
Table_{string id;int value; Table_ tail};
```

```
Table_ Table(string id,int value,struct table *tail);
```

```
Table_ Table=NULL;
```

```
int lookup(Table_ table,string id){
```

```
    assert(table!=NULL);
```

```
    if(id==table.id) return table.value;
```

```
    else return lookup(table.tail,id);
```

```
}
```

```
void update(Table_ *tabptr,string id,int value){
```

```
    *tabptr=Table(id,value,*tabptr);
```

```
}
```

```
int S_FOLLOW[]={SEMICOLON,COMMA};
```

```
void S(void)
```

```
{
```

```
    switch(tok)
```

```
    {
```

```
        case ID:{string id=tokval.id;
```

```
            if(lookahead()==ASSIGN){
```

```
                advance();
```

```
                update(table,id,E());
```

```
            }
```

```
            skipto(S_FOLLOW);
```

```
            break;
```

```
        }
```

```
        case PRINT:{
```

```
            advance();
```

```
        if(lookahead()==LPAREN){
            L();
            eatOrSkipTo(RPAREN,S_FOLLOW);
        }
        break;
    }
default: {
    printf("expected ID or PRINT");
    skipto(S_FOLLOW);
}
}
int L_FOLLOW[]={COMMA,SEMICOLON,RPAREN};
void L(void)
{
    switch(tok)
    {
        case ID:
        case INT:
            {
                printf("%d",E());
                break;
            }
        default:{
            printf("expected ID or INT");
            skipto(L_FOLLOW);
        }
    }
}
int E_FOLLOW[]={SEMICOLON,PLUS,MINUS,TIMES,DIV,RPAREN};
int E()
{
    switch(tok)
    {
        case ID:{
            int i=lookup(table,tokval.id);
            int p=lookahead();
            if(p==PLUS||P==MINUS||P==TIMES||P==DIV){
                advance();
                return B(i);
            }
            advance();
            return i;
        }
        case INT:{
```

```
        int i=tokval.num;
        int p=lookahead();
        if(p==PLUS||P==MINUS||P==TIMES||P==DIV){
            advance();
            return B(i);
        }
        advance();
        return i;
    }
case LPAREN:{
    int i=E();
    if(lookahead()==RPAREN)
        return i;
    else{
        skipto(E_FOLLOW);
        return 0;
    }
}
default:{
    printf("expected ID,INT,LPAREN");
    skipto(E_FOLLOW);
    return 0;
}
}
}
int B_FOLLOW[]={ID,INT,LPAREN};
int B(int a)
{
    switch(tok)
    {
        case PLUS:{
            advance();
            return (a+E());
        }
        case MINUS:{
            advance();
            return (a-E());
        }
        case TIMES:{
            advance();
            return (a*E());
        }
        case DIV:{
            advance();
```

```
        return (a/E());
    }
    Default:{
        Skipto(B_FOLLOW);
        return 0;
    }
}

void eatOrSkipto(int expected,int *stop)
{
    if(tok==expected){
        eat(expected);
    }
    else{
        skipto(stop);
    }
}
```

【注：只要实现出来就行，答案不唯一，很多同学消除了左递归，都是对的】

5.1 a、b （略）

这里主要是数据结构知识，问题不是很大，大家数据结构学的都不错。