

# 编译原理实验报告

学院：计算机学院

专业：计算机科学与技术

学号：14051111

姓名：陈存粮

指导老师：谌志群

# 目录

1.实验目的	.....	3
2.实验要求	.....	3
3.实验过程	.....	4
4.实验环境	.....	4
5.实验结果	.....	5
5.1 正确结果	.....	5
5.2 词法报错	.....	7
5.3 语法报错	.....	8
5.4 语义分析及中间代码报错	.....	9
6.实验总结	.....	9

备注：以上目录仅为 S 语言词法分析，递归下降语法分析，语义分析及中间代码三个实验的目录项，PL0 语言实验报告在另外一份 Word 里。

## 1. 实验目的：

### 1.1 词法分析实验：

通过设计、开发一个高级语言的词法分析程序，加深对课堂教学内容（包括正规文法、正规表达式、有限自动机、NFA 到 DFA 的转换、DFA 的最小化）的理解，提高词法分析方法的实践能力。

### 1.2 语法分析实验：

通过设计、开发一个高级语言的递归下降语法分析程序，实现对词法分析程序所提供的单词序列进行语法检查和结构分析，加深对相关课堂教学内容的理解，提高语法分析方法的实践能力。

### 1.3 中间代码生成实验：

通过设计、开发一个高级语言的语义检查和中间代码生成程序，加深对相关课堂教学内容，包括语法制导翻译技术、类型确定、类型检查、常见可执行语句（如赋值语句、条件语句、循环语句）翻译技术的理解。

## 2. 实验要求：

### 2.1 词法分析实验：

- (1) 深入理解、掌握有限自动机及其应用；
- (2) 掌握根据语言的词法规则构造识别其单词的有限自动机的方法；
- (3) 掌握 NFA 到 DFA 的等价变换方法、DFA 最小化的方法；
- (4) 掌握设计、编码、调试词法分析程序的技术与方法，具体实现 S 语言（见附录 A）的词法分析程序。

### 2.2 语法分析实验：

- (1) 理解语法分析在编译程序中的作用，以及它与词法分析程序的关系；
- (2) 掌握递归下降语法分析方法的主要原理；
- (3) 理解递归下降分析法对文法的要求；
- (4) 熟练掌握 Select 集合的求解方法；
- (5) 熟练掌握文法变换方法（消除左递归和提取公因子）。

### 2.3 语义分析与中间代码生成实验：

- (1) 了解语义检查和中间代码生成的目的和意义；
- (2) 掌握语义检查和中间代码生成的一般内容；
- (3) 掌握语法制导翻译技术（特别是针对 S-属性定义的自底向上翻译方法和针对 L-属性定义的深度优先翻译方法）；
- (4) 掌握根据翻译目标编制语义子程序的方法（可模仿教材中的语义规则进行设计）。

### 3. 实验过程：

#### 3.1 词法分析实验：

- (1)根据 S 语言的语法规则，总结 S 语言的单词种类与各类单词的结构特征；
- (2)设计描述 S 语言各类单词结构的状态转换图（即有限自动机 FA）；
- (3)对描述各类单词结构的状态转换图进行合并（将各状态转换图的初始状态合并为一个唯一的初态，然后对状态重新编号），构成一个能识别 S 语言所有单词的状态转换图(NFA)。
- (4)对能识别所有单词的 NFA 进行确定化操作，将其转换成等价的 DFA；
- (5)对 DFA 进行最小化操作；
- (6)编写程序，模拟最小化 DFA 的运行，实现 S 语言的词法分析程序；
- (7)撰写实验报告。

#### 3.2 语法分析实验：

- (1)根据 S 语言 BNF 形式的语法规则（见附件 A），写出 S 语言的上下文无关文法；
- (2)求每个产生式的 Select 集：  
$$\text{Select}(A \rightarrow \beta) = \text{First}(\beta), \text{当 } \epsilon \notin \text{First}(\beta) = (\text{First}(\beta) - \{\epsilon\}) \cup \text{Follow}(A), \text{当 } \epsilon \in \text{First}(\beta)$$
- (3)判断是否满足递归下降法分析条件，若不满足用消除左递归和提取公因子等文法等价变换操作对文法进行变换，使其满足递归下降法的要求；
- (4)构造递归下降语法分析程序，对文法中的每个非终结符号按其产生式结构产生相应的语法分析子程序,完成相应的识别任务。其中终结符号产生匹配命令，非终结符号则产生调用命令。实际的语法分析工作从调用主程序（开始符号 S 对应的程序）开始，根据产生式递归调用各个分析子程序；
- (5)撰写实验报告。

#### 3.3 语义分析与中间代码生成实验：

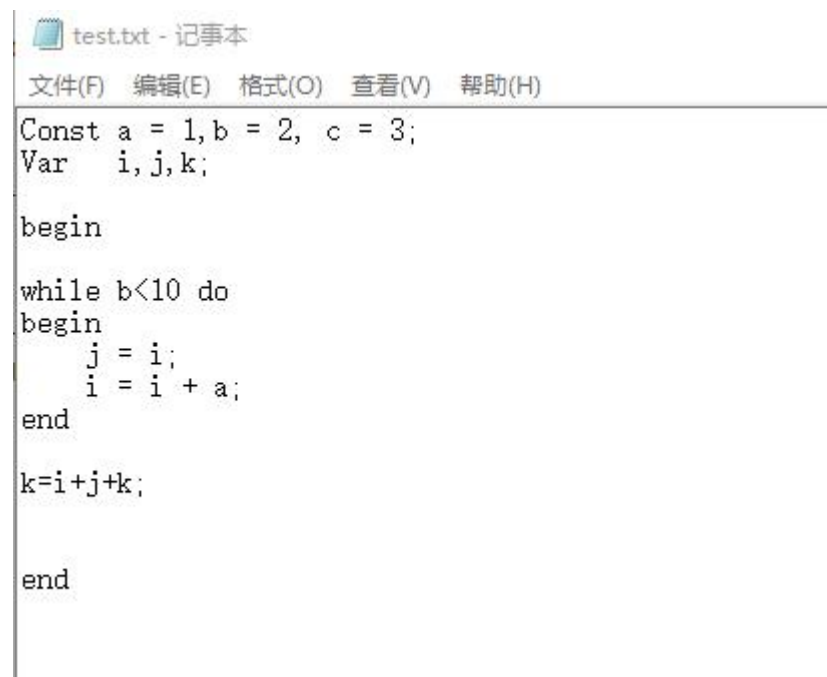
- (1)熟悉 S 语言的语义，了解语义分析阶段需要进行哪些语义检查和需要对哪些可执行语句进行翻译；
- (2)分析 S 语言的文法，必要的话可对文法进行改写（如增加标记非终结符号并为其添加产生式）；
- (3)为相应产生式编写语义子程序（模仿教材中的语义规则进行设计），实现对运算及运算分量进行类型检查（运算的合法性与运算分量类型的一致性或相容性）、变量定义的唯一性检查等语义检查工作；
- (4)理解掌握变量的中间代码、表达式的中间代码、语句的中间代码的结构（见教材）；
- (5)编写语义子程序（模仿教材中的语义规则进行设计），实现 S 语言中各类语句（赋值语句、条件语句、循环语句）的翻译工作，将源程序翻译为中间代码。
- (6)撰写实验报告。

### 4. 实验环境：

Windows10 平台，VisioStdio 2013，  
CPU: i5-4200h 2.8GHz 8GB 内存

## 5. 实验结果：

正确测试代码，包含复合语句，while 语句，和赋值语句，常量声明，变量声明。



```
test.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Const a = 1,b = 2, c = 3;
Var   i,j,k;

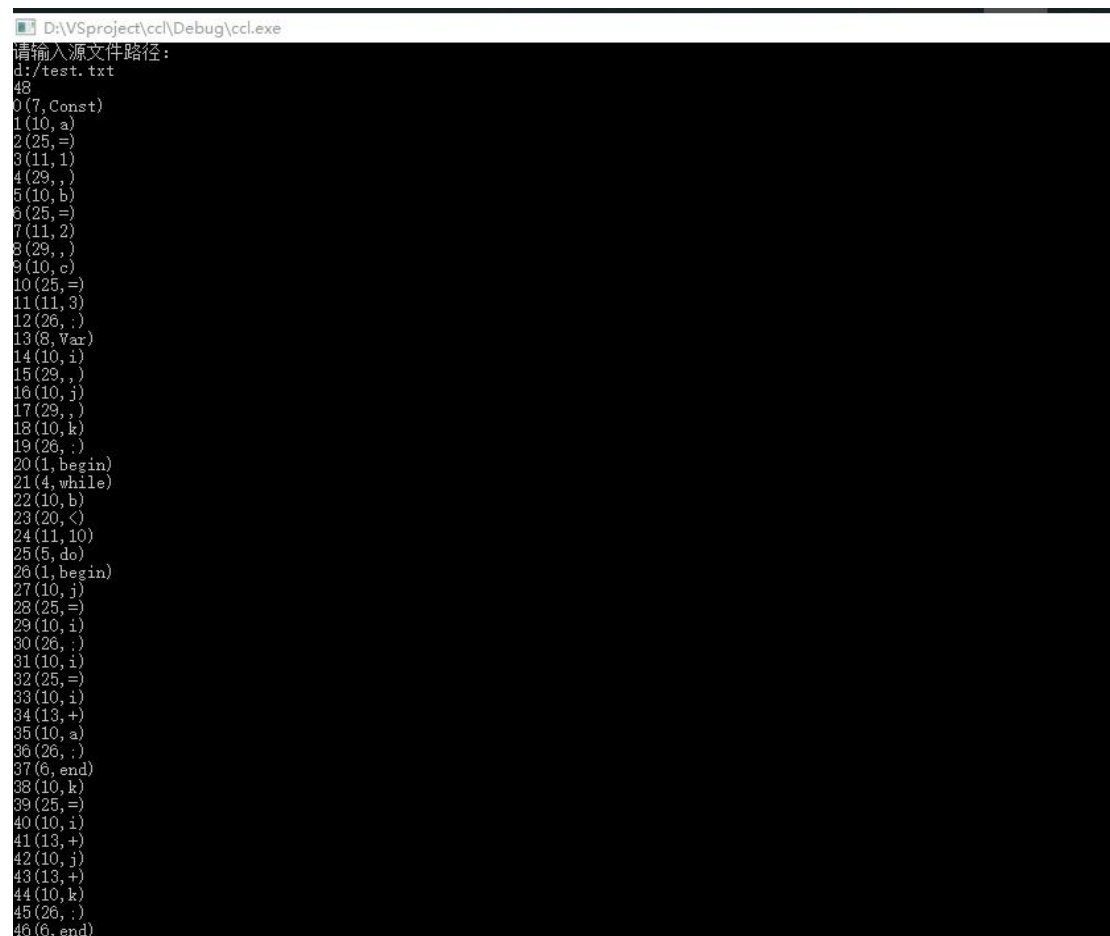
begin

while b<10 do
begin
    j = i;
    i = i + a;
end

k=i+j+k;

end
```

下面为上面测试代码的词法分析结果



```
D:\VSproject\ccl\Debug\ccl.exe
请输入源文件路径:
d:/test.txt
48
0 (7, Const)
1 (10, a)
2 (25, =)
3 (11, 1)
4 (29, ,)
5 (10, b)
6 (25, =)
7 (11, 2)
8 (29, ,)
9 (10, c)
10 (25, =)
11 (11, 3)
12 (26, :)
13 (8, Var)
14 (10, i)
15 (29, ,)
16 (10, j)
17 (29, ,)
18 (10, k)
19 (26, :)
20 (1, begin)
21 (4, while)
22 (10, b)
23 (20, <)
24 (11, 10)
25 (5, do)
26 (1, begin)
27 (10, j)
28 (25, =)
29 (10, i)
30 (26, :)
31 (10, i)
32 (25, =)
33 (10, i)
34 (13, +)
35 (10, a)
36 (26, :)
37 (6, end)
38 (10, k)
39 (25, =)
40 (10, i)
41 (13, +)
42 (10, j)
43 (13, +)
44 (10, k)
45 (26, :)
46 (6, end)
```

## 语法分析输出结果：

```

CONST SUCCESS!!
常量说明
Const
常量定义
    标识符 a
    =
    无符号整数 1
常量定义
    标识符 b
    =
    无符号整数 2
常量定义
    标识符 c
    =
    无符号整数 3
:
CONST SUCCESS!!
变量说明
Var
变量定义
    标识符 i
变量定义
    标识符 j
变量定义
    标识符 k
:
VAR SUCCESS
语句
    语句
        复合语句
            begin
                语句
                    循环语句
                        while
                            条件
                                表达式
                                    项
                                        因子
                                            标识符 b
                                            < | > | =
                                            表达式
                                                项
                                                    因子
                                                        无符号整数 10
do
    语句
        复合语句
            begin
                语句
                    赋值语句
                        标识符 j
                        =
                        表达式
                            项
                                因子
                                    标识符 i
:
VARIABLE SUCCESS

```

```

语句
    赋值语句
        标识符 i
        =
        表达式
            项
                因子
                    标识符 i
                    + | -
                    项
                        因子
                            标识符 a
:
VARIABLE SUCCESS
语句
    end
COMPLEX SUCCESS
LOOP SUCCESS
语句
    赋值语句
        标识符 k
        =
        表达式
            项
                因子
                    标识符 i
                    + | -
                    项
                        因子
                            标识符 j
                    + | -
                    项
                        因子
                            标识符 k
:
VARIABLE SUCCESS
语句
    end
COMPLEX SUCCESS

```

变量常量及其初始值的输出结果:

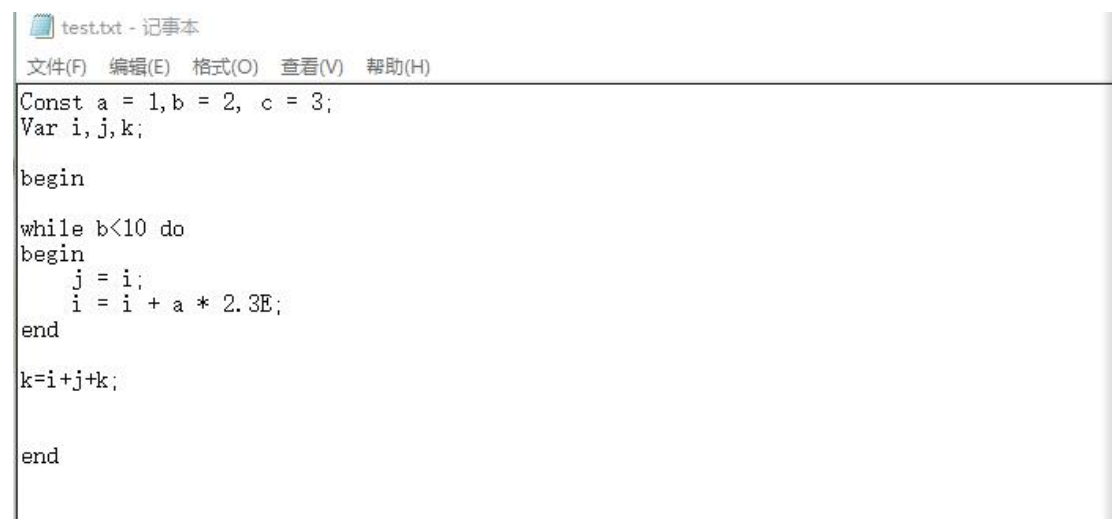
```
7  a  1
7  b  2
7  c  3
10 i  0
10 j  0
10 k  0
```

中间代码生成的输出结果:

```
if @2 < 10 goto l1
l1
j=s0;s0=i;
l1
i=s1;s1=i+@1;
goto l1
l2
k=s2;s2=i+s3;s3=j+k;
```

## 词法报错

测试代码中包含不允许出现的 2.3E, 不应出现项和项



test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
Const a = 1,b = 2, c = 3;
Var i,j,k;

begin

while b<10 do
begin
    j = i;
    i = i + a * 2.3E;
end

k=i+j+k;

end
```

报错如下

```
31(10,i)
32(25,=)
33(10,i)
34(13,+)
35(10,a)
36(15,*)
37(11,2)
38(10,.)
(-1,3E) Error!
40(26,:)
41(6,end)
42(10,k)
43(25,=)
44(10,i)
```

# 语法报错

下图示例测试代码中 while 循环中缺少 do，while 的第一条语句后面缺少分号

test.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Const a = 1,b = 2, c = 3;  
Var i,j,k;  
  
begin  
  
while b<10  
begin  
  j = i  
  i = i + a;  
end  
  
k=i+j+k;  
  
end

下图实验结果表明缺少分号；

标识符 j  
= 表达式  
项  
因子

标识符 i  
ERROR BEFOR =  
VARIABLE ERROR  
  
ERROR BEFOR i  
COMPLEX ERROR  
LOOP SUCCESS  
语句  
赋值语句  
标识符 i  
  
ERROR BEFOR a  
VARIABLE ERROR  
语句  
赋值语句  
标识符 a  
  
error at aConst 类型数据不能赋值  
VARIABLE ERROR  
语句  
  
ERROR BEFOR end  
COMPLEX ERROR

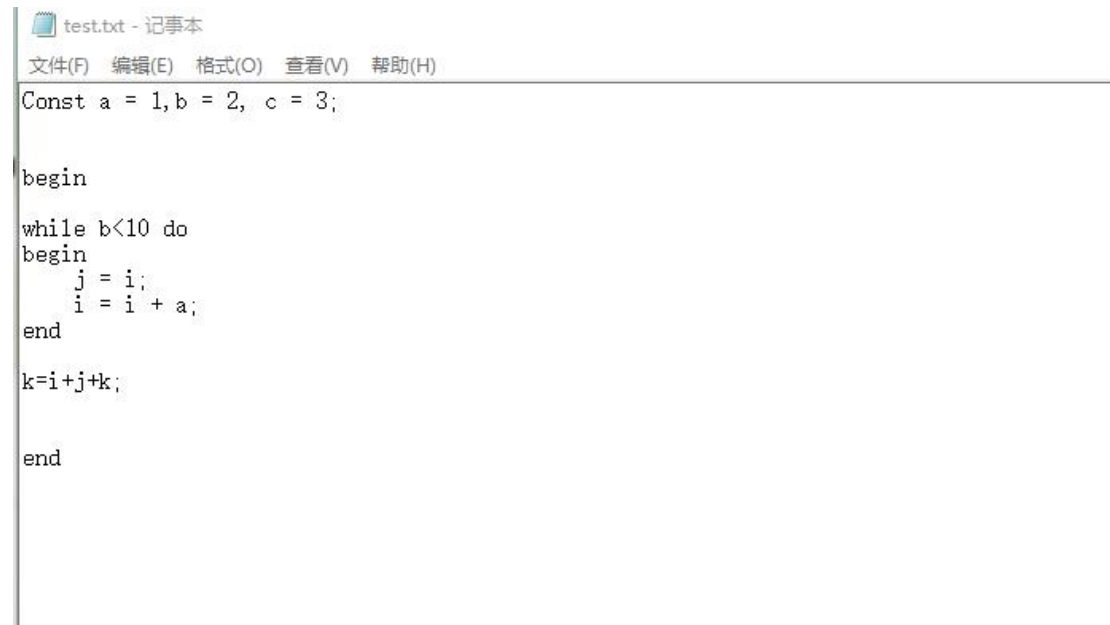
下图实验结果表明缺少 do。

变量定义  
标识符 j  
变量定义  
标识符 k  
j  
VAR SUCCESS  
语句  
语句  
复合语句  
begin  
语句  
循环语句  
while  
条件  
表达式  
项  
因子  
标识符 b  
< | > | =  
表达式  
项  
因子  
无符号整数 10  
  
ERROR BEFOR j  
LOOP ERROR  
语句  
赋值语句  
标识符 j  
= 表达式  
项  
因子  
标识符 i  
  
ERROR BEFOR =  
VARIABLE ERROR  
语句  
ERROR BEFOR i  
COMPLEX ERROR



## 语义分析及中间代码生成报错

下图测试代码没有定义变量 i, j, k。



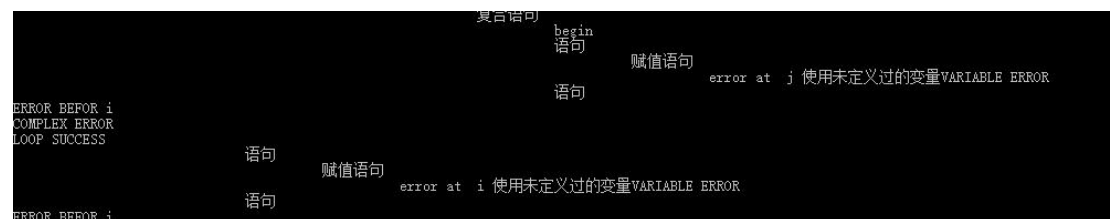
```
test.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Const a = 1, b = 2, c = 3;

begin
while b<10 do
begin
    j = i;
    i = i + a;
end
k=i+j+k;

end
```

下图报错指出 i 未定义。



```
复合语句
begin
语句
赋值语句
error at j 使用未定义过的变量 VARIABLE ERROR
语句
error at i 使用未定义过的变量 VARIABLE ERROR
语句
赋值语句
语句
ERROR BEFOR i
COMPLEX ERROR
LOOP SUCCESS
ERROR BEFOR i
```

## 6. 实验总结：

### 6.1 词法分析实验：

通过词法分析器实验，对整个词法分析过程有了很大的了解，对有限自动机有了深刻的了解。从构建 NFA 到 DFA 的转化，DFA 的最小化过程，程序的实现，对 S 语言分析的技术和测试，都有了很大的认识。在编代码的时候出现了很多的问题，但通过解决，这样对整个分析过程的理解更加深刻了。通过整个实验，让我明白了很多知识。

### 6.2 语法分析实验：

通过递归下降子程序法进行语法分析，对递归下降子程序法的理解更加深刻，从对一个程序-语句串-语句-各种语句-表达式-项-因子的一个递归过程，熟悉掌握递归的整个过程，对整个语法分析的理解，有着很大的帮助。First 集以及 Follow 集和 Select 集的求解，有了理解，对做题目有很大的帮助。总之，通过

语法分析器的实验，让我对整个编译器的实现有了很大的理解，对后续的语义分析，中间代码生成，代码优化有了进一步的认识。

### **6.3 语义分析与中间代码生成实验：**

这次实验加深了我对编译的整个过程的理解，无论是整体还是细节都有了更深入的体会和感悟。活动记录、符号表、中间过程生成等，强化了这些的基础知识的理解。这次实验让我更好的理解了语法分析，词法分析的联系和区别，以及语义分析，中间代码生产的联系。我们要在实际的编程中理解，有问题，解决问题，这样对一个知识点的理解会更加的深刻。