# 目录

[1 目录 1](#_Toc513585137)

[2 语法分析单元测试 3](#_Toc513585138)

[2.1 测试环境 3](#_Toc513585139)

[2.1.1 依赖 3](#_Toc513585140)

[2.1.2 调试输出 3](#_Toc513585141)

[2.1.3 获取可执行文件 3](#_Toc513585142)

[2.1.4 输入文件 4](#_Toc513585143)

[2.1.5 输出文件 4](#_Toc513585144)

[2.2 测试计划 4](#_Toc513585145)

[2.3 测试programstruct 4](#_Toc513585146)

[2.3.1 整体程序结束后有多余内容 4](#_Toc513585147)

[2.3.2 主程序头尾部缺少分号 5](#_Toc513585148)

[2.3.3 程序结尾缺少点号 5](#_Toc513585149)

[2.3.4 主程序头书写错误 6](#_Toc513585150)

[2.3.5 主程序体书写错误 6](#_Toc513585151)

[2.3.6 主程序头前有非法字符 7](#_Toc513585152)

[2.3.7 主程序头前有非法记号，且后部缺失分号 7](#_Toc513585153)

[2.3.8 主程序头前有非法记号，且程序尾部缺失分号 8](#_Toc513585154)

[2.3.9 主程序头前包含非法记号，主程序体书写错误 8](#_Toc513585155)

[2.4 测试program\_head 9](#_Toc513585156)

[2.4.1 缺少左括号 9](#_Toc513585157)

[2.4.2 缺少右括号 9](#_Toc513585158)

[2.4.3 缺少主程序名 10](#_Toc513585159)

[2.4.4 主程序头不完整 10](#_Toc513585160)

[2.4.5 主程序头参数列表缺失 11](#_Toc513585161)

[2.5 测试const\_declarations 12](#_Toc513585162)

[2.5.1 常量定义出现错误 12](#_Toc513585163)

[2.5.2 常量定义出现错误的特殊情况 13](#_Toc513585164)

[2.5.3 缺少分号 13](#_Toc513585165)

[2.6 测试const\_declaration 14](#_Toc513585166)

[2.6.1 常数初始化右值缺失 14](#_Toc513585167)

[2.6.2 常量定义缺少分号 15](#_Toc513585168)

[2.6.3 常量定义缺少等号 16](#_Toc513585169)

[2.7 测试var\_declarations 17](#_Toc513585170)

[2.7.1 变量定义出现错误 17](#_Toc513585171)

[2.7.2 变量定义尾部缺少分号 18](#_Toc513585172)

[2.8 测试var\_declaration 18](#_Toc513585173)

[2.8.1 多个变量定义之间缺少分号分隔 18](#_Toc513585174)

[2.8.2 变量定义缺少冒号 19](#_Toc513585175)

[2.8.3 变量定义类型识别失败 20](#_Toc513585176)

[2.9 测试type 20](#_Toc513585177)

[2.9.1 缺少左中括号 20](#_Toc513585178)

[2.9.2 缺少OF关键字 21](#_Toc513585179)

[2.9.3 数组元素识别失败 21](#_Toc513585180)

[2.9.4 不完整的数组类型 22](#_Toc513585181)

[2.10 测试period 23](#_Toc513585182)

[2.10.1 缺少逗号 23](#_Toc513585183)

[2.10.2 缺少双点号 24](#_Toc513585184)

[2.11 测试subprogram\_declarations 24](#_Toc513585185)

[2.11.1 子程序体尾部缺少分号 25](#_Toc513585186)

[2.12 测试subprogram\_head 25](#_Toc513585187)

[2.12.1 函数名缺失 25](#_Toc513585188)

[2.12.2 缺少冒号 26](#_Toc513585189)

[2.12.3 缺少基本类型关键字 26](#_Toc513585190)

[2.12.4 不完整的函数头 27](#_Toc513585191)

[2.12.5 不完整的过程头 27](#_Toc513585192)

[2.13 测试formal\_parameter 27](#_Toc513585193)

[2.13.1 不完整的形参列表 28](#_Toc513585194)

[2.13.2 右括号缺失 28](#_Toc513585195)

[2.14 测试parameter\_list 28](#_Toc513585196)

[2.14.1 缺少分号 29](#_Toc513585197)

[2.15 测试var\_parameter 29](#_Toc513585198)

[2.15.1 不完整的引用参数列表 29](#_Toc513585199)

[2.16 测试value\_parameter 30](#_Toc513585200)

[2.16.1 缺少冒号 30](#_Toc513585201)

[2.16.2 缺少基本类型关键字 30](#_Toc513585202)

[2.17 测试compound\_statement 31](#_Toc513585203)

[2.17.1 缺少end关键字 31](#_Toc513585204)

[2.18 测试statement\_list 31](#_Toc513585205)

[2.18.1 缺少分号 31](#_Toc513585206)

[2.19 测试statement 32](#_Toc513585207)

[2.19.1 IF语句缺少then关键字 32](#_Toc513585208)

[2.19.2 FOR语句缺少赋值号 32](#_Toc513585209)

[2.19.3 FOR语句缺少关键字to 33](#_Toc513585210)

[2.19.4 FOR语句缺少关键字do 33](#_Toc513585211)

[2.19.5 WHILE语句缺少关键字do 34](#_Toc513585212)

[2.19.6 REPEAT语句缺少关键字until 34](#_Toc513585213)

[2.20 测试id\_varpart 35](#_Toc513585214)

[2.20.1 缺少右中括号 35](#_Toc513585215)

[2.20.2 不完整的数组下标列表 36](#_Toc513585216)

[2.21 测试procedure\_call 36](#_Toc513585217)

[2.21.1 缺少右括号 36](#_Toc513585218)

[2.22 测试expression\_list 37](#_Toc513585219)

[2.22.1 缺少逗号 37](#_Toc513585220)

[2.23 测试simple \_expression 37](#_Toc513585221)

[2.23.1 缺少操作数 37](#_Toc513585222)

[2.24 测试 term 38](#_Toc513585223)

[2.24.1 缺少操作数 38](#_Toc513585224)

[2.25 测试Factor 39](#_Toc513585225)

[2.25.1 缺少右括号 39](#_Toc513585226)

[2.25.2 函数调用表达式列表缺失 39](#_Toc513585227)

[2.26 最终黑盒测试 40](#_Toc513585228)

# 语法分析单元测试

## 测试环境

### 依赖

由于语法分析程序的输入是词法分析得到的记号序列，因此我们的语法分析单元测试依赖于完备的词法分析，同时，源程序的预处理函数和记号结点的数据类型定义都在main.cpp、main.h中。

故需要合并编译进行测试的源程序：

* lex.l
* yacc.y
* main.cpp

### 调试输出

如果定义了宏YYDEBUG，那么就可以输出语法分析程序的调试输出，即按顺序输出处理相应记号时的符号栈和状态栈信息以及移进/归约动作。

见main.h：

1. #define YYDEBUG 1

对应main.cpp中的main函数：

1. yydebug=1;

这里的调试信息能够很好兼容我们定义的错误处理机制，足够胜任之后的测试评估工作。

### 获取可执行文件

* Windows端的脚本

1. flex lex.l  //编译lex.l，生成lex.yy.c
2. ren lex.yy.c lex.yy.cpp  //将lex.yy.c更名为lex.yy.cpp
3. bison -vd --debug yacc.y  //编译yacc.y，生成yacc.tab.h和yacc.tab.c
4. ren yacc.tab.c yacc.tab.cpp  //将yacc.tab.c更名为yacc.cpp
5. g++ lex.yy.cpp yacc.tab.cpp main.cpp -o pascal2c.exe
6. //编译生成可执行文件pascal2c.exe

* Mac端的脚本

1. flex lex.l
2. mv lex.yy.c lex.yy.cpp
3. bison -v -d --debug yacc.y
4. mv yacc.tab.c yacc.tab.cpp
5. g++ -o test yacc.tab.cpp lex.yy.cpp main.cpp

### 输入文件

PascalProgram.pas

### 输出文件

preProcessed.pas

## 测试计划

* 白盒测试每个产生式的错误情况。**（语法分析的DEBUG信息较冗长，为了方便阅读，本版本为全黑盒版本）**
* 组合单个产生式的错误为多个错误进行黑盒测试，即另语法分析树有多处错误子树，观察语法分析程序能否成功报告错误信息并回复。
* 以一个较复杂的，没有错误的程序作为最终黑盒测试。

## 测试programstruct

正确产生式为：programstruct ---> program\_head ';' program\_body '.'

### 整体程序结束后有多余内容

programstruct ---> program\_head ';' program\_body '.'

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **function** fun(d,e:**integer**):**integer**;
5. **begin**
6. fun:=d+e;
7. **end**;
8. **begin**
9. c[3,8]:=fun(3,8);
10. **end**.
11. b:=c[1,6];

* 预期结果

在建立完整的语法分析树之后，读取到多余的信息，报告信息冗余，并定位到行号以及具体的符号位。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, redundant content at the **end**!, location: 10.5

* 分析

实际结果与预期一致。程序分析完毕后栈内状态为0，当读取到多余的记号序列时，语法分析程序提示内容冗余。

### 主程序头尾部缺少分号

programstruct ---> program\_head error program\_body '.'

* 测试用例

1. **program** test(input,output)
2. **begin**
4. **end**.

* 预期结果

遇到错误应报告“缺少分号”的错误，并报告行号和具体位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting ';', location: 2.1
3. syntax error, missing a semicolon here, location: 1.27

* 分析

实际结果与预期结果一致，当遇到错误记号时，语法分析程序能正确记录错误信息，并继续分析后续记号串直到读到结束符，在最后报告整体错误信息。

### 程序结尾缺少点号

programstruct ---> program\_head ';' program\_body error

* 测试用例

1. **program** test(input,output);
2. **begin**
4. **end**

* 预期结果

遇到错误应报告“缺少点号”的错误，并报告行号和具体位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected $end, expecting '.', location: 4.3
3. syntax error, missing a dot here, location: 4.4

* 分析

实际结果与预期结果一致，当遇到错误记号时，语法分析程序能正确记录错误信息，并继续分析后续记号串直到读到结束符，在最后报告整体错误信息。

### 主程序头书写错误

programstruct ---> error ';' program\_body '.'

* 测试用例

1. ;
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别主程序头失败，应能报告确实主程序头的错误信息，并准确定位。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting PROGRAM, location: 1.1
3. syntax error, fatal error **in** **program** head, maybe missing keyword "**program**", location: 1.1-1.1

* 分析

实际结果与预期结果一致。语法分析程序开始状态为0，此状态下读到’;’时，会移进error符号，从而触发相关报错动作，并继续分析后续记号串。

### 主程序体书写错误

programstruct ---> program\_head ';' error '.'

* 测试用例

1. **program** test(input,output);
3. .

* 预期结果

语法分析程序识别主程序体失败，应在不影响其他记号串分析的情况下，报告主程序体的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected '.', location: 3.1
3. fatal error **in** **program** body, location: 3.1

* 分析

实际结果与预期结果一致。当语法分析程序状态为11时，读取到’.’时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 主程序头前有非法字符

programstruct ---> error program\_head ';' program\_body '.'

* 测试用例

1. 1 **program** test(input,output);
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头前的非法字符，应在不影响其他记号串分析的情况下，报告主程序体的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting PROGRAM, location: 1.1
3. syntax error, invalid symbol before **program** head, location: 1.1-1.2

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态0时，读取到非法记号例如UINUM时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 主程序头前有非法记号，且后部缺失分号

programstruct ---> error program\_head error program\_body '.'

* 测试用例

1. 1 **program** test(input,output)
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头前的非法字符，后部缺失分号，应在不影响其他记号串分析的情况下，报告主程序体的所有错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting PROGRAM, location: 1.1
3. syntax error, unexpected \_BEGIN, expecting ';', location: 2.1
4. syntax error, invalid token before **program** head, maybe missing keyword "**program**", location: 1.1-1.2
5. syntax error, missing a semicolon here, location: 1.29

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态0时，读取到非法记号例如UINUM时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。随后在状态6未遇到’;’就读到”begin”时，也是同样的处理。

### 主程序头前有非法记号，且程序尾部缺失分号

programstruct ---> error program\_head ';' program\_body error

* 测试用例

1. 1 **program** test(input,output);
2. **begin**
4. **end**

* 预期结果

语法分析程序识别到主程序头前的非法字符，程序尾部缺失点号，应在不影响其他记号串分析的情况下，报告主程序体的所有错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting PROGRAM, location: 1.1
3. syntax error, unexpected $end, expecting '.', location: 4.3
4. syntax error, invalid token before **program** head, maybe missing keyword "**program**", location: 1.1-1.2
5. syntax error, missing a dot here, location: 4.4

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态0时，读取到非法记号例如UINUM时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。随后在状态31未遇到’.’就读到文本结束符时，也是同样的处理。

### 主程序头前包含非法记号，主程序体书写错误

programstruct ---> error program\_head ';' error '.'

* 测试用例

1. 1 **program** test(input,output);
3. .

* 预期结果

语法分析程序识别到主程序头前的非法字符，主程序体书写错误，应在不影响其他记号串分析的情况下，报告主程序的所有错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting PROGRAM, location: 1.1
3. syntax error, unexpected '.', location: 3.1
4. syntax error, invalid token before **program** head, maybe missing keyword "**program**", location: 1.1-1.2
5. syntax error, fatal error **in** **program** body, location: 1.30-3.0

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态0时，读取到非法记号例如UINUM时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。随后在状态16未遇到program\_body就读到点号时，也是同样的处理。

## 测试program\_head

正确产生式为：program\_head ---> PROGRAM IDENTIFIER '(' idlist ')'

### 缺少左括号

program\_head ---> PROGRAM IDENTIFIER error idlist ')'

* 测试用例

1. **program** test input,ouput);
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头缺失左括号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting '(', location: 1.14
3. syntax error, missing a left bracket here, location: 1.13

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态8时，未读取到’(‘就读取到标识符记号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 缺少右括号

program\_head ---> PROGRAM IDENTIFIER '(' idlist error

* 测试用例

1. **program** test (input,output;
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头缺失右括号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ')' **or** ',', location: 1.27
3. syntax error, missing a right bracket here, location: 1.27

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态36时，未读取到’)‘就读取到分号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 缺少主程序名

program\_head ---> PROGRAM error '(' idlist ')'

* 测试用例

1. **program** (input,output);
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头缺失名字，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected '(', expecting IDENTIFIER, location: 1.9
3. syntax error, missing **program** name here, location: 1.8

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态2时，未读取到’)‘就读取到分号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 主程序头不完整

program\_head ---> PROGRAM error

* 测试用例

1. **program**
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头缺失两个或两个以上的必要记号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting IDENTIFIER, location: 2.1
3. syntax error, **program** head imcomplete, location: 1.1-1.7

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态2时，未读取到主程序头所需后续记号就读取到主程序体信息时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 主程序头参数列表缺失

1： program\_head ---> PROGRAM IDENTIFIER '(' error ')'

2： program\_head ---> PROGRAM IDENTIFIER error

3： program\_head ---> PROGRAM IDENTIFIER '(' error

* 测试用例1

1. **program** test (1,2);
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头参数列表缺失，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting IDENTIFIER, location: 1.15
3. syntax error, **program** identifier list missing **or** imcomplete, location: 1.1-1.12
4. syntax error, missing a semicolon here, location: 1.16

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态4时，读取到参数列表错误时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头参数列表缺失，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting '(', location: 2.1
3. syntax error, **program** identifier list missing **or** imcomplete, location: 1.1-1.12

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态2时，读取到参数列表错误时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例3

1. **program** test(
2. **begin**
4. **end**.

* 预期结果

语法分析程序识别到主程序头参数列表缺失，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting IDENTIFIER, location: 2.1
3. syntax error, **program** identifier list missing **or** imcomplete, location: 1.1-1.12

* 分析

实际结果与预期结果一致。当语法分析程序在开始状态19时，读取到参数列表错误时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试const\_declarations

正确的表达式：const\_declarations ---> CONST const\_declaration ';' | empty

### 常量定义出现错误

const\_declarations ---> CONST error

* 测试用例

1. **program** test(input,output);
2. **const** 1
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义出现错误，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting IDENTIFIER, location: 2.7

* 分析

实际结果与预期结果一致。当语法分析程序在状态12，读取到非法字符（例如测试用例中的UINUM）时会进行丢弃并移进error，从而触发相关报错动作，并继续分析后续记号串。

### 常量定义出现错误的特殊情况

const\_declarations ---> CONST error ';'

* 测试用例

1. **program** test(input,output);
2. **const** 1;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义出现错误，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting IDENTIFIER, location: 2.7
3. syntax error, fatal error **in** **const** declarations, location: 2.7-2.7

* 分析

实际结果与预期结果一致。当语法分析程序在状态12，读取到非法字符（例如测试用例中的UINUM）时会进行丢弃并移进error，从而触发相关报错动作，并继续分析后续记号串。

### 缺少分号

const\_declarations ---> CONST const\_declaration error

* 测试用例

1. **program** test(input,output);
2. **const** a=3;b=5
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义缺少分号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting ';', location: 3.1
3. syntax error, missing a semicolon here, location: 2.7-2.13

* 分析

实际结果与预期结果一致。当语法分析程序在状态25时，未读到分号便读到了“beigin”时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试const\_declaration

正确的产生式为：const\_declaration ---> const\_declaration ';' IDENTIFIER '=' const\_value

| IDENTIFIER '=' const\_value

### 常数初始化右值缺失

1: const\_declaration ---> const\_declaration ';' IDENTIFIER '=' error

2: const\_declaration ---> IDENTIFIER '=' error

* 测试用例1

1. **program** test(input,output);
2. **const** a=1;b=;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常数初始化右值缺失，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', location: 2.13
3. syntax error, constant definition missing initial r-value, location: 2.12-2.12

* 分析

实际结果与预期结果一致。当语法分析程序在状态91时，读取到常数初始化右值缺失时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test(input,output);
2. **const** a=;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常数初始化右值缺失，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', location: 2.9
3. syntax error, constant definition missing initial r-value, location: 2.9-2.9

* 分析

实际结果与预期结果一致。当语法分析程序在状态43时，读取到常数初始化右值缺失时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 常量定义缺少分号

const\_declaration ---> const\_declaration error IDENTIFIER '=' const\_value

* 测试用例

1. **program** test(input,output);
2. **const** a=1 b=a;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义缺少分号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting ';', location: 2.11
3. syntax error, missing a semicolon here, location: 2.7-2.10

* 分析

实际结果与预期结果一致。当语法分析程序在状态25时，读取到常量定义缺少分号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 常量定义缺少等号

1： const\_declaration ---> const\_declaration ';' IDENTIFIER error const\_value

2： const\_declaration ---> IDENTIFIER error const\_value

* 测试用例1

1. **program** test(input,output);
2. **const** a=1;b 2;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义缺少等号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting '=', location: 2.13
3. syntax error, missing a equal sign here, location: 2.11-2.11

* 分析

实际结果与预期结果一致。当语法分析程序在状态70时，读取到常量定义缺少等号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test(input,output);
2. **const** a 1;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到常量定义缺少等号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting '=', location: 2.9
3. syntax error, missing a equal sign here, location: 2.9-2.9

* 分析

实际结果与预期结果一致。当语法分析程序在状态24时，读取到常量定义缺少等号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试var\_declarations

var\_declarations ---> VAR var\_declaration ';'

| empty

### 变量定义出现错误

* 测试用例1

1. **program** test(input,output);
2. **var** ;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到变量定义出现错误，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting IDENTIFIER, location: 2.5
3. syntax error, fatal error **in** variant declarations, location: 2.1-2.3

* 分析

实际结果与预期结果一致。当语法分析程序在状态27时，读取到变量定义出现错误时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test(input,output);
2. **var** x;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到变量定义出现错误，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ',' **or** ':', location: 2.6

* 分析

实际结果与预期结果一致。当语法分析程序在状态47时，读取到变量定义出现错误时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 变量定义尾部缺少分号

var\_declarations ---> VAR var\_declaration error

* 测试用例

1. **program** test(input,output);
2. **var** x:**integer**; y:**real**
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到变量定义尾部缺少分号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting ';', location: 3.1
3. syntax error, missing a semicolon here, location: 2.22

* 分析

实际结果与预期结果一致。当语法分析程序在状态48时，读取到变量定义尾部缺少分号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试var\_declaration

var\_declaration ---> var\_declaration ';' idlist ':' type

| idlist ':' type

### 多个变量定义之间缺少分号分隔

var\_declaration ---> var\_declaration error idlist ':' type

* 测试用例

1. **program** test(input,output);
2. **var** x:**integer** y:**real**;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到多个变量定义之间缺少分号分隔，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting ';', location: 2.15
3. syntax error, missing a semicolon here, location: 2.14

* 分析

实际结果与预期结果一致。当语法分析程序在状态48时，读取到多个变量定义之间缺少分号分隔时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 变量定义缺少冒号

1： var\_declaration ---> var\_declaration ';' idlist error type

2： var\_declaration ---> idlist error type

* 测试用例1

1. **program** test(input,output);
2. **var** x:**integer**;y **real**;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到多个变量定义缺少冒号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting ',' **or** ':', location: 2.17
3. syntax error, missing a colon here, location: 2.16

* 分析

实际结果与预期结果一致。当语法分析程序在状态98时，读取到变量定义缺少冒号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test(input,output);
2. **var** b **real**;
3. **procedure** pro;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

语法分析程序识别到多个变量定义缺少冒号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting ',' **or** ':', location: 2.7
3. syntax error, missing a colon here, location: 2.6

* 分析

实际结果与预期结果一致。当语法分析程序在状态47时，读取到变量定义缺少冒号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 变量定义类型识别失败

var\_declaration ---> var\_declaration ';' idlist ':' error

* 测试用例1

1. **program** test(input,output);
2. **var** x:**integer**;y:;
3. **begin**
5. **end**.

* 预期结果

遇到错误时报告“缺少变量定义类型”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ARRAY **or** TYPE, location: 2.17
3. syntax error, missing a **type** here, location: 2.17

* 分析

测试结果与预期相同。可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试type

正确的表达式为：type ---> TYPE

| ARRAY '[' period ']' OF TYPE

### 缺少左中括号

type ---> ARRAY error period ']' OF TYPE

* 测试用例

1. **program** test(input,output);
2. **var** a: **array** 1..3] **of** **integer**;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到数组的类型声明缺少左中括号，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting '[', location: 2.14
3. syntax error, missing a left square bracket here, location: 2.13

* 分析

实际结果与预期结果一致。当语法分析程序在状态92时，读取到变量定义类型识别失败时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 缺少OF关键字

type ---> ARRAY '[' period ']' error TYPE

* 测试用例

1. **program** test(input,output);
2. **var** a: **array**[1..3] **integer**;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到数组的类型声明缺少OF关键字，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting OF, location: 2.20
3. syntax error, missing keyword "OF" here, location: 2.19-2.19

* 分析

实际结果与预期结果一致。当语法分析程序在状态203时，读取到数组的类型声明缺少OF关键字时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 数组元素识别失败

type ---> ARRAY '[' period ']' OF error

* 测试用例

1. **program** test(input,output);
2. **var** a: **array**[1..2] **of** 123;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别数组元素类型失败，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting TYPE, location: 2.23
3. syntax error, missing a base **type** keyword here, location: 2.22
4. syntax error, missing a semicolon here, location: 2.26
5. fatal error **in** **program** body, location: 5.4

* 分析

实际结果与预期结果一致。当语法分析程序在状态246时，读取到错误的数组元素类型名时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 不完整的数组类型

1： type ---> ARRAY error

2： type ---> ARRAY '[' error

3： type ---> ARRAY '[' period error

* 测试用例1

1. **program** test(input,output);
2. **var** a: **array**;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到不完整的数据类型，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting '[', location: 2.13
3. syntax error, incomplete **array** **type**, location: 2.8-2.13

* 分析

实际结果与预期结果一致。当语法分析程序在状态92时，读取到不完整的数组类型时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例2

1. **program** test(input,output);
2. **var** a: **array** [a;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到不完整的数据类型，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting UINUM, location: 2.15
3. syntax error, incomplete **array** **type**, location: 2.8-2.15
4. syntax error, missing a semicolon here, location: 2.16
5. fatal error **in** **program** body, location: 5.4

* 分析

实际结果与预期结果一致。当语法分析程序在状态122时，读取到不完整的数组类型时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

* 测试用例3

1. **program** test(input,output);
2. **var** a: **array** [1..3,4..6,7..9;
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到不完整的数据类型，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ',' **or** ']', location: 2.29
3. syntax error, incomplete **array** **type**, location: 2.8-2.29

* 分析

实际结果与预期结果一致。当语法分析程序在状态158时，读取到不完整的数组类型时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试period

正确的产生式为：period ---> period ',' UINUM RANGEDOT UINUM

| UINUM RANGEDOT UINUM

### 缺少逗号

period ---> period error UINUM RANGEDOT UINUM

* 测试用例

1. **program** test(input,output);
2. **var** a: **array** [1..3 4..6 7..9];
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到缺少逗号的错误时，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting ',' **or** ']', location: 2.20
3. syntax error, missing a comma here, location: 2.19
4. syntax error, unexpected UINUM, expecting ',' **or** ']', location: 2.25
5. syntax error, missing a comma here, location: 2.24
6. syntax error, unexpected ';', expecting OF, location: 2.30

* 分析

实际结果与预期结果一致。当语法分析程序在状态158时，未读取到逗号就读取到数字时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

### 缺少双点号

period ---> period ',' UINUM error UINUM

* 测试用例

1. **program** test(input,output);
2. **var** a: **array** [1..3,4 6,7 9];
3. **begin**
5. **end**.

* 预期结果

语法分析程序识别到缺少双点号时，应在不影响其他记号串分析的情况下，报告主程序的错误信息，并报告位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting RANGEDOT, location: 2.22
3. syntax error, missing range dot .. here, location: 2.21
4. syntax error, unexpected UINUM, expecting RANGEDOT, location: 2.26
5. syntax error, missing range dot .. here, location: 2.25

* 分析

实际结果与预期结果一致。当语法分析程序在状态243时，读取到缺少双点号时，会移进error记号，从而触发相关报错动作，并继续分析后续记号串。

## 测试subprogram\_declarations

正确产生式为：

subprogram\_declarations ---> subprogram\_declarations subprogram ';' | empty

### 子程序体尾部缺少分号

subprogram\_declarations ---> subprogram\_declarations subprogram error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误应报告“缺少分号”，并报告错误出现的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting ';', location: 4.1
3. syntax error, missing a semicolon here, location: 3.14

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试subprogram\_head

正确产生式为：

subprogram\_head ---> PROCEDURE IDENTIFIER formal\_parameter | FUNCTION IDENTIFIER formal\_parameter ':' TYPE

### 函数名缺失

subprogram\_head ---> FUNCTION error formal\_parameter ':' TYPE

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **function** :**integer**;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误应报告“缺少函数名”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ':', expecting IDENTIFIER, location: 3.10
3. syntax error, missing **function** name, location: 3.9

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 缺少冒号

subprogram\_head ---> FUNCTION IDENTIFIER formal\_parameter error TYPE

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **function** fun(a:**integer**)**integer**;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误应报告“缺少冒号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting ':', location: 3.24
3. syntax error, missing a colon here, location: 3.23

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 缺少基本类型关键字

subprogram\_head ---> FUNCTION IDENTIFIER formal\_parameter ':' error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **function** fun;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时应报告 “缺少基本类型关键字”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ':', location: 3.13
3. syntax error, missing a base **type** keyword here, location: 3.13

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 不完整的函数头

subprogram\_head ---> FUNCTION error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **function**;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“函数头不完整”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting IDENTIFIER, location: 3.9
3. syntax error, incomplete **function** head, location: 3.1-3.9

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 不完整的过程头

subprogram\_head ---> PROCEDURE error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure**;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“过程头不完整”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting IDENTIFIER, location: 3.10
3. syntax error, incomplete **procedure** head, location: 3.1-3.10

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试formal\_parameter

正确表达式为：formal\_parameter ---> '(' parameter\_list ')' | empty

### 不完整的形参列表

formal\_parameter ---> '(' error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro();
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“形参列表不完整”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting VAR **or** IDENTIFIER, location: 3.15
3. syntax error, incomplete formal parameter list, location: 3.14-3.15

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 右括号缺失

formal\_parameter ---> '(' parameter\_list error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro(a:**integer**;
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告 “缺少右括号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected \_BEGIN, expecting VAR **or** IDENTIFIER, location: 4.1
3. syntax error, missing a right bracket here, location: 3.24

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试parameter\_list

正确表达式为：parameter\_list ---> parameter\_list ';' parameter | parameter

### 缺少分号

parameter\_list ---> parameter\_list error parameter

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro(**var** a:**integer** b:**real**);
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“缺少分号，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting ';' **or** ')', location: 3.29
3. syntax error, missing a semicolon here, location: 3.28

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试var\_parameter

正确表达式为：var\_parameter ---> VAR value\_parameter

### 不完整的引用参数列表

var\_parameter ---> VAR error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro(**var** );
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“不完整参数列表”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ')', expecting IDENTIFIER, location: 3.19
3. syntax error, incomplete refereced parameter list, location: 3.15-3.19

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试value\_parameter

正确表达式为：value\_parameter ---> idlist ':' TYPE

### 缺少冒号

value\_parameter ---> idlist 'error TYPE

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro(a **integer**);
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“缺少分号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting ',' **or** ':', location: 3.17
3. syntax error, missing a colon here, location: 3.16

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 缺少基本类型关键字

value\_parameter ---> idlist ':' error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro(a:);
4. **begin**
5. **end**;
6. **begin**
8. **end**.

* 预期结果

遇到错误时报告“缺少基本类型关键字”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected TYPE, expecting ',' **or** ':', location: 3.17
3. syntax error, missing a colon here, location: 3.16

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试compound\_statement

正确表达式为：compound\_statement ---> \_BEGIN statement\_list END

### 缺少end关键字

compound\_statement ---> \_BEGIN statement\_list error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **begin**
5. .

* 预期结果

遇到错误时报告“缺少END关键字”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected '.', expecting END **or** ';', location: 5.1
3. syntax error, missing keyword "**end**", location: 3.6

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试statement\_list

正确表达式为：statement\_list ---> statement\_list ';' statement | statement

### 缺少分号

statement\_list ---> statement\_list error statement

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. b:=b;
7. b:=b
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少分号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting END **or** ';', location: 8.5
3. syntax error, missing a semicolon here, location: 7.9

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试statement

正确表达式为：

statement ---> variable ASSIGNOP expression | procedure\_call |

compound\_statement | IF expression THEN statement else\_part |

FOR IDENTIFIER ASSIGNOP expression TO expression DO statement |

WHILE expression DO statement | REPEAT statement UNTIL expression | empty

### IF语句缺少then关键字

statement ---> IF expression error statement else\_part

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **if** a>b
7. b:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少then关键字”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting THEN, location: 7.9
3. syntax error, missing keyword "**then**", location: 6.11

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### FOR语句缺少赋值号

statement ---> FOR IDENTIFIER error expression TO expression DO statement

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **for** a 1 **to** 2 **do**
7. b:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少赋值号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting ASSIGNOP, location: 6.11
3. syntax error, missing assignop ":=", location: 6.10

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### FOR语句缺少关键字to

statement ---> FOR IDENTIFIER ASSIGNOP expression error expression DO statement

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **for** a:=1  2 **do**
7. b:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少关键字to”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected UINUM, expecting TO, location: 6.15
3. syntax error, missing keywrod "**to**", location: 6.13

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### FOR语句缺少关键字do

statement ---> FOR IDENTIFIER ASSIGNOP expression TO expression error statement

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **for** a:=1 **to** 2
7. b:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少关键字do”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting DO, location: 7.9
3. syntax error, missing keywrod "**do**", location: 6.13

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### WHILE语句缺少关键字do

statement ---> WHILE expression error statement

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **while** a=1
7. b:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少关键字do”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting DO, location: 7.9
3. syntax error, missing keywrod "**do**", location: 6.14

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### REPEAT语句缺少关键字until

statement ---> REPEAT statement error expression

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. **procedure** pro;
4. **var** a:**integer**;
5. **begin**
6. **repeat**
7. b:=1
8. a=1;
9. **end**;
10. **begin**
12. **end**.

* 预期结果

遇到错误时报告“缺少关键字until”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting UNTIL, location: 8.5
3. syntax error, missing keywrod "**until**", location: 8.5

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试id\_varpart

正确表达式为：id\_varpart ---> '[' expression\_list ']' | empty

### 缺少右中括号

id\_varpart ---> '[' expression\_list error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **procedure** pro;
5. **var** a:**integer**;
6. **begin**
7. **repeat**
8. c[1,6:=100
9. **until** a=1;
10. **end**;
11. **begin**
13. **end**.

* 预期结果

遇到错误时报告“缺少右中括号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ASSIGNOP, expecting ',' **or** ']', location: 8.14
3. syntax error, missing a right square bracket here, location: 8.14

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 不完整的数组下标列表

id\_varpart ---> '[' error

* 测试用例：

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **procedure** pro;
5. **var** a:**integer**;
6. **begin**
7. **repeat**
8. c[:=100
9. **until** a=1;
10. **end**;
11. **begin**
13. **end**.

* 预期结果

遇到错误时报告“不完整的数组下表列表”，并报告错误所在的具体位置。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ASSIGNOP, location: 8.11
3. syntax error, incomplete expression list **of** **array** subindex, location: 8.10-8.12

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试procedure\_call

正确表达式：procedure\_call ---> IDENTIFIER | IDENTIFIER '(' expression\_list ')'

### 缺少右括号

procedure\_call ---> IDENTIFIER '(' expression\_list error

* 测试用例

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **procedure** pro;
5. **var** a:**integer**;
6. **begin**
7. pro(a;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告“缺少右括号”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ')' **or** ',', location: 7.10
3. syntax error, missing a right bracket here, location: 7.10

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试expression\_list

正确表达式：expression\_list ---> expression\_list ',' expression | expression

### 缺少逗号

expression\_list ---> expression\_list error expression

* 测试用例：

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **procedure** pro;
5. **var** a:**integer**;
6. **begin**
7. c[a+b a\*b]:=1;
8. **end**;
9. **begin**
11. **end**.

* 预期结果

遇到错误时报告：“缺少逗号”，并报告错误所在的具体位置，行号和列号。

* 测试结果：

1. Here **is** the syntax error information
2. syntax error, unexpected IDENTIFIER, expecting ',' **or** ']', location: 7.11
3. syntax error, missing a comma here, location: 7.10

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试simple \_expression

正确表达式：

simple\_expression ---> simple\_expression ADDOP term| simple\_expression '-' term| term

### 缺少操作数

simple\_expression ---> simple\_expression ADDOP error term %prec ADD

| simple\_expression '-' error term %prec ADD

* 测试用例

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. **procedure** pro;
4. **var** c:**integer**;
5. **begin**
6. a:=1;
7. b:=2;
8. c:=a-+b
9. **end**;
10. **begin**
12. **end**.

* 预期结果

遇到错误时报告“缺少操作数”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ADDOP, location: 8.14
3. syntax error, missing operand, location: 8.14

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试 term

正确表达式：term ---> term MULOP factor| factor

### 缺少操作数

term ---> term MULOP error factor %prec MUL

* 测试用例

1. **program** test(input,output);
2. **var** a,b:**integer**;
3. **procedure** pro;
4. **var** c:**integer**;
5. **begin**
6. a:=1;
7. b:=2;
8. c:=a\*+b
9. **end**;
10. **begin**
12. **end**.

* 预期结果

遇到错误时报告“缺少操作数”，并报告错误所在的具体位置，行号和列号。

* 测试结果

1. Here **is** the syntax error information
2. syntax error, unexpected ADDOP, location: 8.14
3. syntax error, missing operand, location: 8.14

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 测试Factor

正确表达式：factor ---> UINUM | UFNUM | variable | IDENTIFIER '(' expression\_list ')' | '(' expression ')' | NOT factor | '-' factor | CHAR

### 缺少右括号

factor ---> IDENTIFIER '(' expression\_list error

* 测试用例：

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **function** fun(d,e:**integer**):**integer**;
5. **begin**
6. fun:=d+e;
7. **end**;
8. **begin**
9. c[3,8]:=fun(b;
10. **end**.

* 预期结果：

遇到错误时报告“缺少右括号”，并报告错误的具体位置，行号和列号。

* 测试结果：

1. Here **is** the syntax error information
2. syntax error, unexpected ';', expecting ')' **or** ',', location: 9.18
3. syntax error, missing a right bracket here, location: 9.18

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

### 函数调用表达式列表缺失

factor ---> IDENTIFIER '(' error

* 测试用例：

1. **program** test(input,output);
2. **var** b:**real**;
3. c:**array**[1..5,6..10] **of** **integer**;
4. **function** fun(d,e:**integer**):**integer**;
5. **begin**
6. fun:=d+e;
7. **end**;
8. **begin**
9. c[3,8]:=fun(;
10. **end**.

* 预期结果：

遇到错误时报告“缺失函数调用表达式列表”，并报告错误所在的具体位置，行号和列号。

* 测试结果：

1. Here **is** the syntax error information
2. syntax error, unexpected ';', location: 9.17
3. syntax error, missing actual parameter list **of** **function** call, location: 9.17

测试结果与预期一致，可以看到错误被成功识别并报告了正确位置，第一行为yacc自带的yyerror函数结果，第二行为自行使用函数重载进行错误处理的结果。

## 最终黑盒测试

此时我们使用复杂的PASCAL源程序，对语法分析程序进行黑盒测试。

注释掉main.cpp里的DEBUG信息显示：

1. //yydebug=1;

* 测试用例

快速排序：

1. **program** quicksort(input,output);
2. **var**
3. n,i:**integer**;
4. a:**array**[0..100000] **of** **integer**;
5. b:**char**;
7. **procedure** kp(l,r:**integer**);
8. **var**
9. i,j,mid:**integer**;
10. **begin**
11. b:='a';
12. **if** l>=r **then** exit;
13. i:=l;j:=r;mid:=a[(l+r) **div** 2];
14. **repeat**
15. **begin**
16. **while** a[i]<mid **do** inc(i);
17. **while** a[j]>mid **do** dec(j);
18. **if** i<=j **then**
19. **begin**
20. a[0]:=a[i];a[i]:=a[j];a[j]:=a[0];
21. inc(i);dec(j);
22. **end**
23. **end**
24. **until** i>j;
25. kp(l,j);
26. kp(i,r)
27. **end**;
29. **begin**
30. readln(n);
31. **for** i:=1 **to** n **do**
32. read(a[i]);
33. kp(1,n);
34. **for** i:=1 **to** n **do**
35. **write**(a[i]);
36. **end**.

* 预期结果

语法分析程序顺利分析到文本尾部，无任何错误信息，且输出文本与输入文本内容一致。

* 测试结果

1. -----**begin** parsing /Users/mac/yacc\_and\_lex\_repository/lex\_and\_yacc/preProcessed.pas
2. -----**end** parsing

* 分析

实际结果与预期结果一致。