

# 天津大学

## 2022秋季编译原理大作业第六组测试报告

### C- - 编译器测试



学 院	智能与计算学部
专 业	软件工程
年 级	2020级
姓 名	黄丽丽,丁小芮,王悦君,孙思远
学 号	3020244288,3020244201,3020244340,3020244341

2022 年 11 月 13 日

## 第一章 测试整体概述

### 1.1 测试目的

(1) 编写正确的测试样例进行输入以检验词法分析及语法分析代码执行的正确性。

(2) 分别编写有词法错误和语法错误的样例进行输入检验词法分析及语法分析对错误的识别与处理。

### 1.2 测试方法

利用助教给出的6个正确样例以实现第一个测试目的，随后编写错误测试样例，每段错误代码针对其中一个典型的错误，观察代码的运行以及处理方式以实现第二个测试目的。

### 1.3 测试流程图

测试过程的流程图如下：

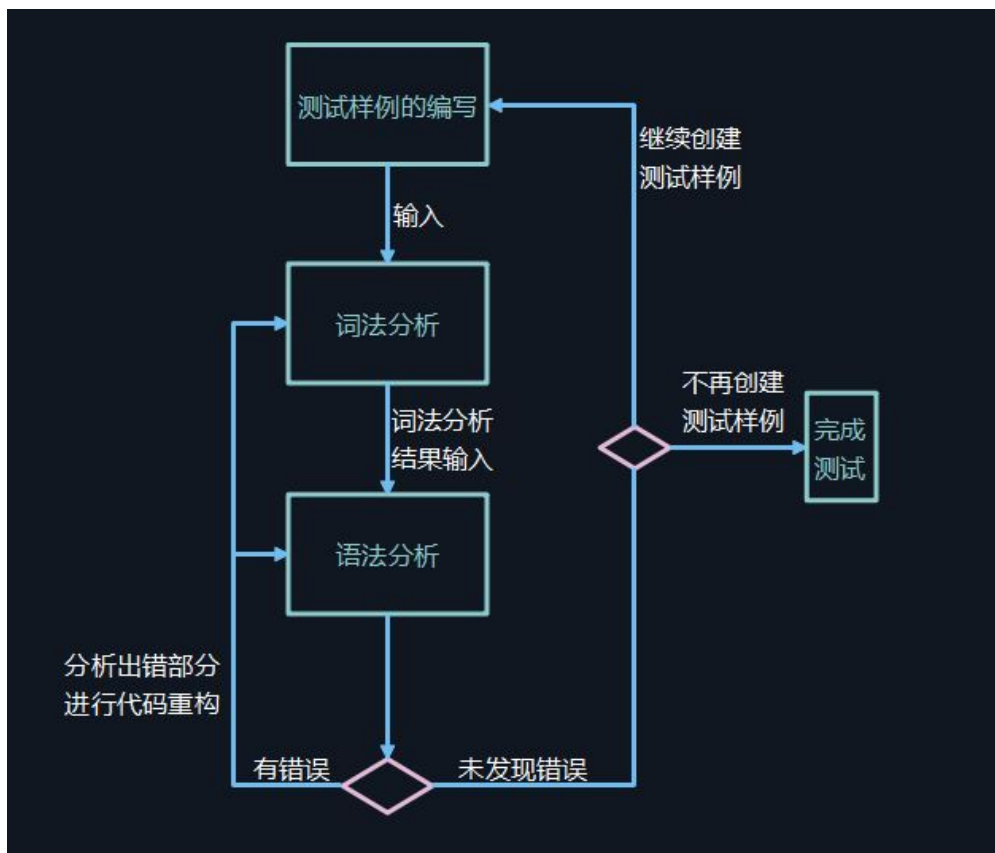


图 1-1 测试过程流程图

## 第二章 代码正确性测试

### 2.1 实验指导书测试样例

#### 2.1.1 代码片段

```
≡ test1.sy
1 //test domain of global var define and local define
2 int a = 10;
3 int main(){
4     int a = 10;
5     return 0;
6 }
```

图 2-1 实验指导书测试样例

#### 2.1.2 词法分析结果

```
int      <KW, 1>
a        <IDN, a>
=        <OP, 11>
10       <INT, 10>
;        <SE, 24>
int      <KW, 1>
main     <KW, 5>
(        <SE, 20>
)        <SE, 21>
{        <SE, 22>
int      <KW, 1>
a        <IDN, a>
=        <OP, 11>
10       <INT, 10>
;        <SE, 24>
return   <KW, 3>
0        <INT, 0>
;        <SE, 24>
}        <SE, 23>
```

图 2-2 实验指导书测试样例词法分析结果

### 2.1.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef##= reduction
- 10 ==# move
- 11 initVal#10 reduction
- 12 exp#10 reduction
- 13 assignExp#10 reduction
- 14 eqExp#10 reduction
- 15 relExp#10 reduction
- 16 addExp#10 reduction
- 17 mulExp#10 reduction
- 18 unaryExp#10 reduction
- 19 primaryExp#10 reduction
- 20 number#10 reduction
- 21 INT#10 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction
- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#int reduction
- 30 decl#int reduction
- 31 varDecl#int reduction
- 32 bType#int reduction
- 33 int#int move
- 34 varDef#main reduction

35 Ident#main move

36 argVarDef##( error

可以看到，在分析到“(”时，语法分析结果错误，原因是原来的文法中

$$funcType \rightarrow void | int$$

而现在的文法为

$$funcType \rightarrow void$$

故根据最新的文法，无法正确解析

## 2.2 已给测试样例1

### 2.2.1 代码片段

```

≡ test1.sy
1  //test domain of global var define and local define
2  int a;
3  void func(){
4      a=10;
5      return 0;
6  }

```

图 2-3 已给测试样例1

### 2.2.2 词法分析结果

```

int      <KW, 1>
a        <IDN, a>
;        <SE, 24>
void     <KW, 2>
func     <IDN, func>
(        <SE, 20>
)        <SE, 21>
{        <SE, 22>
a        <IDN, a>
=        <OP, 11>
10       <INT, 10>
;        <SE, 24>
return   <KW, 3>
0        <INT, 0>
;        <SE, 24>
}        <SE, 23>

```

图 2-4 已给测试样例1词法分析结果

### 2.2.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef#; reduction
- 10 argVarDecl#; reduction
- 11 ;#; move
- 12 compUnit#void reduction
- 13 funcDef#void reduction
- 14 funcType#void reduction
- 15 void#void move
- 16 Ident#func move
- 17 #( move
- 18 funcFParams#) reduction
- 19 )#) move
- 20 block# reduction
- 21 # move
- 22 blockItem#a reduction
- 23 stmt#a reduction
- 24 exp#a reduction
- 25 assignExp#a reduction
- 26 eqExp#a reduction
- 27 relExp#a reduction
- 28 addExp#a reduction
- 29 mulExp#a reduction
- 30 unaryExp#a reduction
- 31 Ident#a move
- 32 callFunc#= reduction
- 33 mulExpAtom#= reduction
- 34 addExpAtom#= reduction

35 relExpAtom#= reduction  
36 eqExpAtom#= reduction  
37 assignExpAtom#= reduction  
38 =#= move  
39 eqExp#10 reduction  
40 relExp#10 reduction  
41 addExp#10 reduction  
42 mulExp#10 reduction  
43 unaryExp#10 reduction  
44 primaryExp#10 reduction  
45 number#10 reduction  
46 INT#10 move  
47 mulExpAtom#; reduction  
48 addExpAtom#; reduction  
49 relExpAtom#; reduction  
50 eqExpAtom#; reduction  
51 assignExpAtom#; reduction  
52 ;#; move  
53 blockItem#return reduction  
54 stmt#return reduction  
55 return#return move  
56 argExp#0 reduction  
57 exp#0 reduction  
58 assignExp#0 reduction  
59 eqExp#0 reduction  
60 relExp#0 reduction  
61 addExp#0 reduction  
62 mulExp#0 reduction  
63 unaryExp#0 reduction  
64 primaryExp#0 reduction  
65 number#0 reduction  
66 INT#0 move  
67 mulExpAtom#; reduction  
68 addExpAtom#; reduction  
69 relExpAtom#; reduction  
70 eqExpAtom#; reduction

71 assignExpAtom#; reduction

72 ;#; move

73 blockItem# reduction

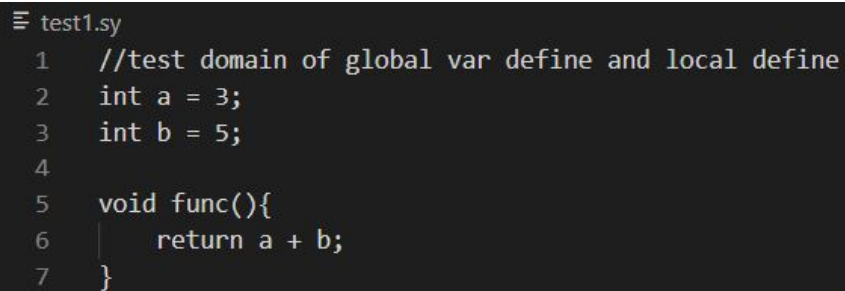
74 # move

75 compUnit## reduction

76 EOF#EOF accept

## 2.3 已给测试样例2

### 2.3.1 代码片段



```
test1.sy
1 //test domain of global var define and local define
2 int a = 3;
3 int b = 5;
4
5 void func(){
6     return a + b;
7 }
```

图 2-5 已给测试样例2

### 2.3.2 词法分析结果

### 2.3.3 语法分析结果

1 program#int reduction

2 compUnit#int reduction

3 decl#int reduction

4 varDecl#int reduction

5 bType#int reduction

6 int#int move

7 varDef#a reduction

8 Ident#a move

9 argVarDef##= reduction

10 =#= move



```
int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
3      <INT, 3>
;      <SE, 24>
int    <KW, 1>
b      <IDN, b>
=      <OP, 11>
5      <INT, 5>
;      <SE, 24>
void   <KW, 2>
func   <IDN, func>
(      <SE, 20>
)      <SE, 21>
{      <SE, 22>
return <KW, 3>
a      <IDN, a>
+      <OP, 6>
b      <IDN, b>
;      <SE, 24>
;      <SE, 23>
```

图 2-6 已给测试样例2词法分析结果

- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction
- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#int reduction
- 30 decl#int reduction

31 varDecl#int reduction  
32 bType#int reduction  
33 int#int move  
34 varDef#b reduction  
35 Ident#b move  
36 argVarDef#= reduction  
37 =#= move  
38 initVal#5 reduction  
39 exp#5 reduction  
40 assignExp#5 reduction  
41 eqExp#5 reduction  
42 relExp#5 reduction  
43 addExp#5 reduction  
44 mulExp#5 reduction  
45 unaryExp#5 reduction  
46 primaryExp#5 reduction  
47 number#5 reduction  
48 INT#5 move  
49 mulExpAtom#; reduction  
50 addExpAtom#; reduction  
51 relExpAtom#; reduction  
52 eqExpAtom#; reduction  
53 assignExpAtom#; reduction  
54 argVarDecl#; reduction  
55 ;#; move  
56 compUnit#void reduction  
57 funcDef#void reduction  
58 funcType#void reduction  
59 void#void move  
60 Ident#func move  
61 #( move  
62 funcFParams#) reduction  
63 )#) move  
64 block# reduction  
65 # move  
66 blockItem#return reduction

67 stmt#return reduction  
68 return#return move  
69 argExp#a reduction  
70 exp#a reduction  
71 assignExp#a reduction  
72 eqExp#a reduction  
73 relExp#a reduction  
74 addExp#a reduction  
75 mulExp#a reduction  
76 unaryExp#a reduction  
77 Ident#a move  
78 callFunc#+ reduction  
79 mulExpAtom#+ reduction  
80 addExpAtom#+ reduction  
81 +#+ move  
82 mulExp#b reduction  
83 unaryExp#b reduction  
84 Ident#b move  
85 callFunc#; reduction  
86 mulExpAtom#; reduction  
87 addExpAtom#; reduction  
88 relExpAtom#; reduction  
89 eqExpAtom#; reduction  
90 assignExpAtom#; reduction  
91 ;#; move  
92 blockItem# reduction  
93 # move  
94 compUnit### reduction  
95 EOF#EOF accept

## 2.4 已给测试样例3

### 2.4.1 代码片段

### 2.4.2 词法分析结果

### 2.4.3 语法分析结果

```
test1.sy
1 //test domain of global var define and local define
2 int a = 3;
3 int b = 5;
4
5 void func(){
6     int a = 5;
7     return a + b;
8 }
```

图 2-7 已给测试样例3

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef##= reduction
- 10 ==#= move
- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction

```
int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
3      <INT, 3>
;      <SE, 24>
int    <KW, 1>
b      <IDN, b>
=      <OP, 11>
5      <INT, 5>
;      <SE, 24>
void   <KW, 2>
func   <IDN, func>
(      <SE, 20>
)      <SE, 21>
{      <SE, 22>
int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
5      <INT, 5>
;      <SE, 24>
return <KW, 3>
a      <IDN, a>
+      <OP, 6>
b      <IDN, b>
;      <SE, 24>
}      <SE, 23>
```

图 2-8 已给测试样例3词法分析结果

- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#int reduction
- 30 decl#int reduction
- 31 varDecl#int reduction
- 32 bType#int reduction
- 33 int#int move
- 34 varDef#b reduction
- 35 Ident#b move
- 36 argVarDef# = reduction
- 37 =# = move
- 38 initVal#5 reduction
- 39 exp#5 reduction
- 40 assignExp#5 reduction
- 41 eqExp#5 reduction
- 42 relExp#5 reduction
- 43 addExp#5 reduction
- 44 mulExp#5 reduction

45 unaryExp#5 reduction  
46 primaryExp#5 reduction  
47 number#5 reduction  
48 INT#5 move  
49 mulExpAtom#; reduction  
50 addExpAtom#; reduction  
51 relExpAtom#; reduction  
52 eqExpAtom#; reduction  
53 assignExpAtom#; reduction  
54 argVarDecl#; reduction  
55 ;#; move  
56 compUnit#void reduction  
57 funcDef#void reduction  
58 funcType#void reduction  
59 void#void move  
60 Ident#func move  
61 #( move  
62 funcFParams#) reduction  
63 )#) move  
64 block# reduction  
65 # move  
66 blockItem#int reduction  
67 decl#int reduction  
68 varDecl#int reduction  
69 bType#int reduction  
70 int#int move  
71 varDef#a reduction  
72 Ident#a move  
73 argVarDef#=# reduction  
74 =#=# move  
75 initVal#5 reduction  
76 exp#5 reduction  
77 assignExp#5 reduction  
78 eqExp#5 reduction  
79 relExp#5 reduction  
80 addExp#5 reduction

81 mulExp#5 reduction  
82 unaryExp#5 reduction  
83 primaryExp#5 reduction  
84 number#5 reduction  
85 INT#5 move  
86 mulExpAtom#; reduction  
87 addExpAtom#; reduction  
88 relExpAtom#; reduction  
89 eqExpAtom#; reduction  
90 assignExpAtom#; reduction  
91 argVarDecl#; reduction  
92 ;#; move  
93 blockItem#return reduction  
94 stmt#return reduction  
95 return#return move  
96 argExp#a reduction  
97 exp#a reduction  
98 assignExp#a reduction  
99 eqExp#a reduction  
100 relExp#a reduction  
101 addExp#a reduction  
102 mulExp#a reduction  
103 unaryExp#a reduction  
104 Ident#a move  
105 callFunc#+ reduction  
106 mulExpAtom#+ reduction  
107 addExpAtom#+ reduction  
108 ++ move  
109 mulExp#b reduction  
110 unaryExp#b reduction  
111 Ident#b move  
112 callFunc#; reduction  
113 mulExpAtom#; reduction  
114 addExpAtom#; reduction  
115 relExpAtom#; reduction  
116 eqExpAtom#; reduction

117 assignExpAtom#; reduction

118 ;#; move

119 blockItem# reduction

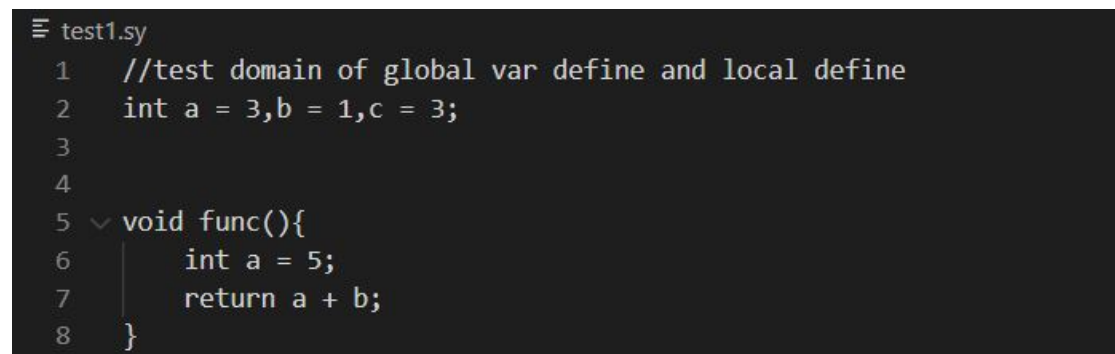
120 # move

121 compUnit## reduction

122 EOF#EOF accept

## 2.5 已给测试样例4

### 2.5.1 代码片段



```

≡ test1.sy
1 //test domain of global var define and local define
2 int a = 3,b = 1,c = 3;
3
4
5 void func(){
6     int a = 5;
7     return a + b;
8 }
  
```

图 2-9 已给测试样例4

### 2.5.2 词法分析结果

### 2.5.3 语法分析结果

1 program#int reduction

2 compUnit#int reduction

3 decl#int reduction

4 varDecl#int reduction

5 bType#int reduction

6 int#int move

7 varDef#a reduction

8 Ident#a move

9 argVarDef#-= reduction

10 -=#-= move

11 initVal#3 reduction

12 exp#3 reduction

13 assignExp#3 reduction



```

int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
3      <INT, 3>
,      <SE, 25>
b      <IDN, b>
=      <OP, 11>
1      <INT, 1>
,      <SE, 25>
c      <IDN, c>
=      <OP, 11>
3      <INT, 3>
;      <SE, 24>
void   <KW, 2>
func   <IDN, func>
(      <SE, 20>
)      <SE, 21>
{      <SE, 22>
int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
5      <INT, 5>
;      <SE, 24>
return <KW, 3>
a      <IDN, a>
+      <OP, 6>
b      <IDN, b>
;      <SE, 24>
}      <SE, 23>

```

图 2-10 已给测试样例4词法分析结果

- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#, reduction
- 23 addExpAtom#, reduction
- 24 relExpAtom#, reduction
- 25 eqExpAtom#, reduction
- 26 assignExpAtom#, reduction
- 27 argVarDecl#, reduction
- 28 ,#, move
- 29 varDef#b reduction
- 30 Ident#b move
- 31 argVarDef#= reduction

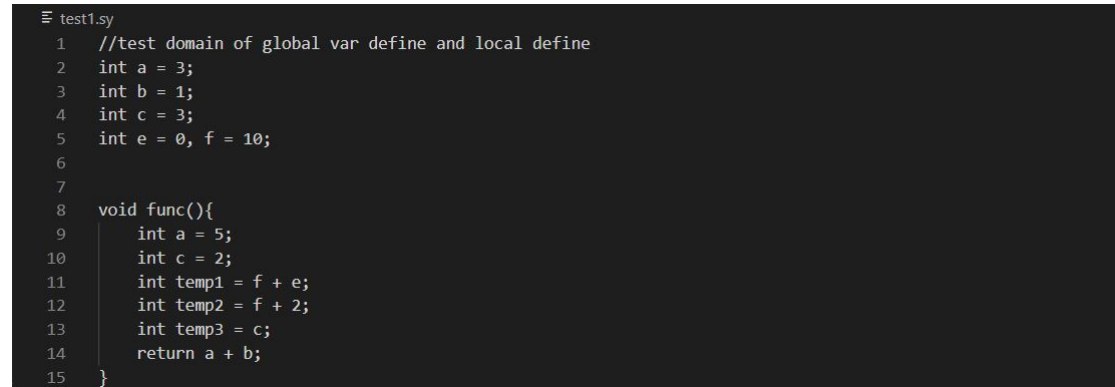
32 =#= move  
33 initVal#1 reduction  
34 exp#1 reduction  
35 assignExp#1 reduction  
36 eqExp#1 reduction  
37 relExp#1 reduction  
38 addExp#1 reduction  
39 mulExp#1 reduction  
40 unaryExp#1 reduction  
41 primaryExp#1 reduction  
42 number#1 reduction  
43 INT#1 move  
44 mulExpAtom#, reduction  
45 addExpAtom#, reduction  
46 relExpAtom#, reduction  
47 eqExpAtom#, reduction  
48 assignExpAtom#, reduction  
49 argVarDecl#, reduction  
50 ,#, move  
51 varDef#c reduction  
52 Ident#c move  
53 argVarDef#= reduction  
54 =#= move  
55 initVal#3 reduction  
56 exp#3 reduction  
57 assignExp#3 reduction  
58 eqExp#3 reduction  
59 relExp#3 reduction  
60 addExp#3 reduction  
61 mulExp#3 reduction  
62 unaryExp#3 reduction  
63 primaryExp#3 reduction  
64 number#3 reduction  
65 INT#3 move  
66 mulExpAtom#; reduction  
67 addExpAtom#; reduction

68 relExpAtom#; reduction  
69 eqExpAtom#; reduction  
70 assignExpAtom#; reduction  
71 argVarDecl#; reduction  
72 ;#; move  
73 compUnit#void reduction  
74 funcDef#void reduction  
75 funcType#void reduction  
76 void#void move  
77 Ident#func move  
78 (#( move  
79 funcFParams#) reduction  
80 )#) move  
81 block# reduction  
82 # move  
83 blockItem#int reduction  
84 decl#int reduction  
85 varDecl#int reduction  
86 bType#int reduction  
87 int#int move  
88 varDef#a reduction  
89 Ident#a move  
90 argVarDef#= reduction  
91 =#= move  
92 initVal#5 reduction  
93 exp#5 reduction  
94 assignExp#5 reduction  
95 eqExp#5 reduction  
96 relExp#5 reduction  
97 addExp#5 reduction  
98 mulExp#5 reduction  
99 unaryExp#5 reduction  
100 primaryExp#5 reduction  
101 number#5 reduction  
102 INT#5 move  
103 mulExpAtom#; reduction

104 addExpAtom#; reduction  
105 relExpAtom#; reduction  
106 eqExpAtom#; reduction  
107 assignExpAtom#; reduction  
108 argVarDecl#; reduction  
109 ;#; move  
110 blockItem#return reduction  
111 stmt#return reduction  
112 return#return move  
113 argExp#a reduction  
114 exp#a reduction  
115 assignExp#a reduction  
116 eqExp#a reduction  
117 relExp#a reduction  
118 addExp#a reduction  
119 mulExp#a reduction  
120 unaryExp#a reduction  
121 Ident#a move  
122 callFunc#+ reduction  
123 mulExpAtom#+ reduction  
124 addExpAtom#+ reduction  
125 ++ move  
126 mulExp#b reduction  
127 unaryExp#b reduction  
128 Ident#b move  
129 callFunc#; reduction  
130 mulExpAtom#; reduction  
131 addExpAtom#; reduction  
132 relExpAtom#; reduction  
133 eqExpAtom#; reduction  
134 assignExpAtom#; reduction  
135 ;#; move  
136 blockItem# reduction  
137 # move  
138 compUnit## reduction  
139 EOF#EOF accept

## 2.6 已给测试样例5

### 2.6.1 代码片段



```
test1.sy
1 //test domain of global var define and local define
2 int a = 3;
3 int b = 1;
4 int c = 3;
5 int e = 0, f = 10;
6
7
8 void func(){
9     int a = 5;
10    int c = 2;
11    int temp1 = f + e;
12    int temp2 = f + 2;
13    int temp3 = c;
14    return a + b;
15 }
```

图 2-11 已给测试样例5

### 2.6.2 词法分析结果

### 2.6.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef##= reduction
- 10 ==#= move
- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction
- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#int reduction
- 30 decl#int reduction
- 31 varDecl#int reduction
- 32 bType#int reduction
- 33 int#int move
- 34 varDef#b reduction

35 Ident#b move  
36 argVarDef##= reduction  
37 ==#= move  
38 initVal#1 reduction  
39 exp#1 reduction  
40 assignExp#1 reduction  
41 eqExp#1 reduction  
42 relExp#1 reduction  
43 addExp#1 reduction  
44 mulExp#1 reduction  
45 unaryExp#1 reduction  
46 primaryExp#1 reduction  
47 number#1 reduction  
48 INT#1 move  
49 mulExpAtom#; reduction  
50 addExpAtom#; reduction  
51 relExpAtom#; reduction  
52 eqExpAtom#; reduction  
53 assignExpAtom#; reduction  
54 argVarDecl#; reduction  
55 ;#; move  
56 compUnit#int reduction  
57 decl#int reduction  
58 varDecl#int reduction  
59 bType#int reduction  
60 int#int move  
61 varDef#c reduction  
62 Ident#c move  
63 argVarDef##= reduction  
64 ==#= move  
65 initVal#3 reduction  
66 exp#3 reduction  
67 assignExp#3 reduction  
68 eqExp#3 reduction  
69 relExp#3 reduction  
70 addExp#3 reduction

71 mulExp#3 reduction  
72 unaryExp#3 reduction  
73 primaryExp#3 reduction  
74 number#3 reduction  
75 INT#3 move  
76 mulExpAtom#; reduction  
77 addExpAtom#; reduction  
78 relExpAtom#; reduction  
79 eqExpAtom#; reduction  
80 assignExpAtom#; reduction  
81 argVarDecl#; reduction  
82 ;#; move  
83 compUnit#int reduction  
84 decl#int reduction  
85 varDecl#int reduction  
86 bType#int reduction  
87 int#int move  
88 varDef#e reduction  
89 Ident#e move  
90 argVarDef#=# reduction  
91 =#=# move  
92 initVal#0 reduction  
93 exp#0 reduction  
94 assignExp#0 reduction  
95 eqExp#0 reduction  
96 relExp#0 reduction  
97 addExp#0 reduction  
98 mulExp#0 reduction  
99 unaryExp#0 reduction  
100 primaryExp#0 reduction  
101 number#0 reduction  
102 INT#0 move  
103 mulExpAtom#, reduction  
104 addExpAtom#, reduction  
105 relExpAtom#, reduction  
106 eqExpAtom#, reduction



107 assignExpAtom#, reduction  
108 argVarDecl#, reduction  
109 #, move  
110 varDef##f reduction  
111 Ident#f move  
112 argVarDef##= reduction  
113 ==#= move  
114 initVal#10 reduction  
115 exp#10 reduction  
116 assignExp#10 reduction  
117 eqExp#10 reduction  
118 relExp#10 reduction  
119 addExp#10 reduction  
120 mulExp#10 reduction  
121 unaryExp#10 reduction  
122 primaryExp#10 reduction  
123 number#10 reduction  
124 INT#10 move  
125 mulExpAtom#; reduction  
126 addExpAtom#; reduction  
127 relExpAtom#; reduction  
128 eqExpAtom#; reduction  
129 assignExpAtom#; reduction  
130 argVarDecl#; reduction  
131 ;#; move  
132 compUnit#void reduction  
133 funcDef#void reduction  
134 funcType#void reduction  
135 void#void move  
136 Ident#func move  
137 #( move  
138 funcFParams#) reduction  
139 )#) move  
140 block# reduction  
141 # move  
142 blockItem#int reduction

143 decl#int reduction  
144 varDecl#int reduction  
145 bType#int reduction  
146 int#int move  
147 varDef#a reduction  
148 Ident#a move  
149 argVarDef##= reduction  
150 ==#= move  
151 initVal#5 reduction  
152 exp#5 reduction  
153 assignExp#5 reduction  
154 eqExp#5 reduction  
155 relExp#5 reduction  
156 addExp#5 reduction  
157 mulExp#5 reduction  
158 unaryExp#5 reduction  
159 primaryExp#5 reduction  
160 number#5 reduction  
161 INT#5 move  
162 mulExpAtom#; reduction  
163 addExpAtom#; reduction  
164 relExpAtom#; reduction  
165 eqExpAtom#; reduction  
166 assignExpAtom#; reduction  
167 argVarDecl#; reduction  
168 ;#; move  
169 blockItem#int reduction  
170 decl#int reduction  
171 varDecl#int reduction  
172 bType#int reduction  
173 int#int move  
174 varDef#c reduction  
175 Ident#c move  
176 argVarDef##= reduction  
177 ==#= move  
178 initVal#2 reduction

179 exp#2 reduction  
180 assignExp#2 reduction  
181 eqExp#2 reduction  
182 relExp#2 reduction  
183 addExp#2 reduction  
184 mulExp#2 reduction  
185 unaryExp#2 reduction  
186 primaryExp#2 reduction  
187 number#2 reduction  
188 INT#2 move  
189 mulExpAtom#; reduction  
190 addExpAtom#; reduction  
191 relExpAtom#; reduction  
192 eqExpAtom#; reduction  
193 assignExpAtom#; reduction  
194 argVarDecl#; reduction  
195 ;#; move  
196 blockItem#int reduction  
197 decl#int reduction  
198 varDecl#int reduction  
199 bType#int reduction  
200 int#int move  
201 varDef#temp1 reduction  
202 Ident#temp1 move  
203 argVarDef##= reduction  
204 ==#= move  
205 initVal#f reduction  
206 exp#f reduction  
207 assignExp#f reduction  
208 eqExp#f reduction  
209 relExp#f reduction  
210 addExp#f reduction  
211 mulExp#f reduction  
212 unaryExp#f reduction  
213 Ident#f move  
214 callFunc#+ reduction

215 mulExpAtom#+ reduction  
216 addExpAtom#+ reduction  
217 +##+ move  
218 mulExp#e reduction  
219 unaryExp#e reduction  
220 Ident#e move  
221 callFunc#; reduction  
222 mulExpAtom#; reduction  
223 addExpAtom#; reduction  
224 relExpAtom#; reduction  
225 eqExpAtom#; reduction  
226 assignExpAtom#; reduction  
227 argVarDecl#; reduction  
228 ;#; move  
229 blockItem#int reduction  
230 decl#int reduction  
231 varDecl#int reduction  
232 bType#int reduction  
233 int#int move  
234 varDef#temp2 reduction  
235 Ident#temp2 move  
236 argVarDef##= reduction  
237 ==#= move  
238 initVal#f reduction  
239 exp#f reduction  
240 assignExp#f reduction  
241 eqExp#f reduction  
242 relExp#f reduction  
243 addExp#f reduction  
244 mulExp#f reduction  
245 unaryExp#f reduction  
246 Ident#f move  
247 callFunc#+ reduction  
248 mulExpAtom#+ reduction  
249 addExpAtom#+ reduction  
250 +##+ move

251 mulExp#2 reduction  
252 unaryExp#2 reduction  
253 primaryExp#2 reduction  
254 number#2 reduction  
255 INT#2 move  
256 mulExpAtom#; reduction  
257 addExpAtom#; reduction  
258 relExpAtom#; reduction  
259 eqExpAtom#; reduction  
260 assignExpAtom#; reduction  
261 argVarDecl#; reduction  
262 ;#; move  
263 blockItem#int reduction  
264 decl#int reduction  
265 varDecl#int reduction  
266 bType#int reduction  
267 int#int move  
268 varDef#temp3 reduction  
269 Ident#temp3 move  
270 argVarDef#=# reduction  
271 =#=# move  
272 initVal#c reduction  
273 exp#c reduction  
274 assignExp#c reduction  
275 eqExp#c reduction  
276 relExp#c reduction  
277 addExp#c reduction  
278 mulExp#c reduction  
279 unaryExp#c reduction  
280 Ident#c move  
281 callFunc#; reduction  
282 mulExpAtom#; reduction  
283 addExpAtom#; reduction  
284 relExpAtom#; reduction  
285 eqExpAtom#; reduction  
286 assignExpAtom#; reduction

287 argVarDecl#; reduction  
288 ;#; move  
289 blockItem#return reduction  
290 stmt#return reduction  
291 return#return move  
292 argExp#a reduction  
293 exp#a reduction  
294 assignExp#a reduction  
295 eqExp#a reduction  
296 relExp#a reduction  
297 addExp#a reduction  
298 mulExp#a reduction  
299 unaryExp#a reduction  
300 Ident#a move  
301 callFunc#+ reduction  
302 mulExpAtom#+ reduction  
303 addExpAtom#+ reduction  
304 ++ move  
305 mulExp#b reduction  
306 unaryExp#b reduction  
307 Ident#b move  
308 callFunc#; reduction  
309 mulExpAtom#; reduction  
310 addExpAtom#; reduction  
311 relExpAtom#; reduction  
312 eqExpAtom#; reduction  
313 assignExpAtom#; reduction  
314 ;#; move  
315 blockItem# reduction  
316 # move  
317 compUnit## reduction  
318 EOF#EOF accept

```

int      <KW, 1>
a        <IDN, a>
=        <OP, 11>
3        <INT, 3>
;        <SE, 24>
int      <KW, 1>
b        <IDN, b>
=        <OP, 11>
1        <INT, 1>
;        <SE, 24>
int      <KW, 1>
c        <IDN, c>
=        <OP, 11>
3        <INT, 3>
;        <SE, 24>
int      <KW, 1>
e        <IDN, e>
=        <OP, 11>
0        <INT, 0>
,        <SE, 25>
f        <IDN, f>
=        <OP, 11>
10       <INT, 10>
;        <SE, 24>
void     <KW, 2>
func     <IDN, func>
(        <SE, 20>
)        <SE, 21>
{        <SE, 22>
int      <KW, 1>
a        <IDN, a>
=        <OP, 11>
5        <INT, 5>
;        <SE, 24>
int      <KW, 1>
c        <IDN, c>
=        <OP, 11>
2        <INT, 2>
;        <SE, 24>
int      <KW, 1>
temp1    <IDN, temp1>
=        <OP, 11>
f        <IDN, f>
+        <OP, 6>
e        <IDN, e>
;        <SE, 24>
int      <KW, 1>
temp2    <IDN, temp2>
=        <OP, 11>
f        <IDN, f>
+        <OP, 6>
2        <INT, 2>
;        <SE, 24>
int      <KW, 1>
temp3    <IDN, temp3>
=        <OP, 11>
c        <IDN, c>
;        <SE, 24>
return   <KW, 3>
a        <IDN, a>
+        <OP, 6>
b        <IDN, b>
;        <SE, 24>
}        <SE, 23>

```

图 2-12 已给测试样例5词法分析结果

## 第三章 代码错误识别及处理测试

### 3.1 错误测试样例1

错误产生原因：标识符非法：以数字开头

#### 3.1.1 代码片段

```
≡ errortest1.sy
1  int 3a = 3;
2  int b = 5;
3
4  void func(){
5      int a = 5;
6      return a + b;
7  }
```

图 3-1 标识符非法测试样例

#### 3.1.2 词法分析结果

```
int <KW,1>
Error keyword or identifier beginning with a number!
```

图 3-2 标识符非法测试样例词法分析结果

输入的标识符“3a”以数字3开头，非法，因此检测到的时候报出异常。

#### 3.1.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef## error



## 3.2 错误测试样例2

错误产生原因：数字以0开头

### 3.2.1 代码片段

```
≡ errortest1.sy
1  int a = 3;
2  int b = 5;
3
4  ✓ void func(){
5      |    int c = 05;
6      |    return a + b;
7      |}
```

图 3-3 标识符非法测试样例

### 3.2.2 词法分析结果

```
int    <KW, 1>
a      <IDN, a>
=      <OP, 11>
3      <INT, 3>
;      <SE, 24>
int    <KW, 1>
b      <IDN, b>
=      <OP, 11>
5      <INT, 5>
;      <SE, 24>
void   <KW, 2>
func   <IDN, func>
(      <SE, 20>
)      <SE, 21>
{      <SE, 22>
int    <KW, 1>
c      <IDN, c>
=      <OP, 11>
A number beginning with 0!
```

图 3-4 标识符非法测试样例词法分析结果

对标识符c的赋值为05，数字以0开头，造成异常。

### 3.2.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef#= reduction
- 10 =#= move
- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction
- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#int reduction
- 30 decl#int reduction
- 31 varDecl#int reduction
- 32 bType#int reduction
- 33 int#int move

34 varDef#b reduction  
35 Ident#b move  
36 argVarDef#= reduction  
37 =#= move  
38 initVal#5 reduction  
39 exp#5 reduction  
40 assignExp#5 reduction  
41 eqExp#5 reduction  
42 relExp#5 reduction  
43 addExp#5 reduction  
44 mulExp#5 reduction  
45 unaryExp#5 reduction  
46 primaryExp#5 reduction  
47 number#5 reduction  
48 INT#5 move  
49 mulExpAtom#; reduction  
50 addExpAtom#; reduction  
51 relExpAtom#; reduction  
52 eqExpAtom#; reduction  
53 assignExpAtom#; reduction  
54 argVarDecl#; reduction  
55 ;#; move  
56 compUnit#void reduction  
57 funcDef#void reduction  
58 funcType#void reduction  
59 void#void move  
60 Ident#func move  
61 (#( move  
62 funcFParams#) reduction  
63 )#) move  
64 block# reduction  
65 # move  
66 blockItem#int reduction  
67 decl#int reduction  
68 varDecl#int reduction  
69 bType#int reduction

70 int#int move  
71 varDef#c reduction  
72 Ident#c move  
73 argVarDef##= reduction  
74 =#= move  
75 initVal## error

### 3.3 错误测试样例3

错误产生原因：输入非法字符

#### 3.3.1 代码片段

观察代码可发现，输入了不可识别字符：|

```
1 int a = 3;  
2 void func(){  
3     int b = |a;  
4 }
```

图 3-5 非法输入测试样例

#### 3.3.2 词法分析结果

```
int    <KW,1>  
a      <IDN,a>  
=      <OP,11>  
3      <INT,3>  
;      <SE,24>  
void   <KW,2>  
func   <IDN,func>  
(      <SE,20>  
)      <SE,21>  
{      <SE,22>  
int    <KW,1>  
b      <IDN,b>  
=      <OP,11>  
The input: | out of range!
```

图 3-6 非法输入测试样例

### 3.3.3 语法分析结果

- 1 program#int reduction
- 2 compUnit#int reduction
- 3 decl#int reduction
- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef##= reduction
- 10 ==#= move
- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction
- 16 addExp#3 reduction
- 17 mulExp#3 reduction
- 18 unaryExp#3 reduction
- 19 primaryExp#3 reduction
- 20 number#3 reduction
- 21 INT#3 move
- 22 mulExpAtom#; reduction
- 23 addExpAtom#; reduction
- 24 relExpAtom#; reduction
- 25 eqExpAtom#; reduction
- 26 assignExpAtom#; reduction
- 27 argVarDecl#; reduction
- 28 ;#; move
- 29 compUnit#void reduction
- 30 funcDef#void reduction
- 31 funcType#void reduction
- 32 void#void move
- 33 Ident#func move
- 34 #( move

35 funcFParams#) reduction  
36 )#) move  
37 block# reduction  
38 # move  
39 blockItem#int reduction  
40 decl#int reduction  
41 varDecl#int reduction  
42 bType#int reduction  
43 int#int move  
44 varDef##b reduction  
45 Ident#b move  
46 argVarDef##= reduction  
47 =##= move  
48 initVal## error

### 3.4 错误测试样例4

错误产生原因：输入非法字符

#### 3.4.1 代码片段

观察代码可发现，输入了不可识别字符：？

```
1 int a? = 3;  
2 void func(){  
3     int b = a;  
4 }
```

图 3-7 非法输入测试样例

#### 3.4.2 词法分析结果

```
int    <KW,1>  
a      <IDN,a>  
The input: ? out of range!
```

图 3-8 非法输入测试样例

### 3.4.3 语法分析结果

1 program#int reduction  
2 compUnit#int reduction  
3 decl#int reduction  
4 varDecl#int reduction  
5 bType#int reduction  
6 int#int move  
7 varDef#a reduction  
8 Ident#a move  
9 argVarDef## reduction  
10 argVarDecl## reduction  
11 ;## error

### 3.5 错误测试样例5

代码片段

```
1    a = 3;  
2    void func(){  
3        |    int b = a;  
4    }
```

图 3-9 错误文法测试样例

词法分析结果

语法分析结果

### 3.6 错误测试样例6

代码片段

词法分析结果

语法分析结果

1 program#int reduction  
2 compUnit#int reduction  
3 decl#int reduction

```
a    <IDN,a>
=    <OP,11>
3    <INT,3>
;    <SE,24>
void  <KW,2>
func  <IDN,func>
(     <SE,20>
)     <SE,21>
{     <SE,22>
int   <KW,1>
b     <IDN,b>
=     <OP,11>
a     <IDN,a>
;     <SE,24>
}     <SE,23>
```

图 3-10 词法分析结果

```
1  program#a  error
```

图 3-11 语法分析结果

- 4 varDecl#int reduction
- 5 bType#int reduction
- 6 int#int move
- 7 varDef#a reduction
- 8 Ident#a move
- 9 argVarDef#-= reduction
- 10 =#-= move
- 11 initVal#3 reduction
- 12 exp#3 reduction
- 13 assignExp#3 reduction
- 14 eqExp#3 reduction
- 15 relExp#3 reduction



```
int a = 3;  
void func(){  
    int b = a;
```

图 3-12 错误文法测试样例

16 addExp#3 reduction  
17 mulExp#3 reduction  
18 unaryExp#3 reduction  
19 primaryExp#3 reduction  
20 number#3 reduction  
21 INT#3 move  
22 mulExpAtom#; reduction  
23 addExpAtom#; reduction  
24 relExpAtom#; reduction  
25 eqExpAtom#; reduction  
26 assignExpAtom#; reduction  
27 argVarDecl#; reduction  
28 ;#; move  
29 compUnit#void reduction  
30 funcDef#void reduction  
31 funcType#void reduction  
32 void#void move  
33 Ident#func move  
34 #( move  
35 funcFParams#) reduction  
36 )#) move  
37 block# reduction  
38 # move  
39 blockItem#int reduction  
40 decl#int reduction  
41 varDecl#int reduction  
42 bType#int reduction

```
int <KW,1>
a <IDN,a>
= <OP,11>
3 <INT,3>
; <SE,24>
void <KW,2>
func <IDN,func>
( <SE,20>
) <SE,21>
{ <SE,22>
int <KW,1>
b <IDN,b>
= <OP,11>
a <IDN,a>
; <SE,24>
```

图 3-13 词法分析结果

43 int#int move  
44 varDef#b reduction  
45 Ident#b move  
46 argVarDef# = reduction  
47 =# = move  
48 initVal#a reduction  
49 exp#a reduction  
50 assignExp#a reduction  
51 eqExp#a reduction  
52 relExp#a reduction  
53 addExp#a reduction  
54 mulExp#a reduction  
55 unaryExp#a reduction  
56 Ident#a move  
57 callFunc#; reduction

58 mulExpAtom#; reduction  
59 addExpAtom#; reduction  
60 relExpAtom#; reduction  
61 eqExpAtom#; reduction  
62 assignExpAtom#; reduction  
63 argVarDecl#; reduction  
64 ;#; move  
65 blockItem## reduction  
66 ## error

### 3.7 错误测试样例7

代码片段

```
int a = 3;
void func(){
    int b = a
}
```

图 3-14 错误文法测试样例

词法分析结果

文法分析结果

1 program#int reduction  
2 compUnit#int reduction  
3 decl#int reduction  
4 varDecl#int reduction  
5 bType#int reduction  
6 int#int move  
7 varDef#a reduction  
8 Ident#a move  
9 argVarDef##= reduction  
10 =#= move  
11 initVal#3 reduction

```
int <KW,1>
a <IDN,a>
= <OP,11>
3 <INT,3>
; <SE,24>
void <KW,2>
func <IDN,func>
( <SE,20>
) <SE,21>
{ <SE,22>
int <KW,1>
b <IDN,b>
= <OP,11>
a <IDN,a>
} <SE,23>
```

图 3-15 词法分析结果

12 exp#3 reduction  
13 assignExp#3 reduction  
14 eqExp#3 reduction  
15 relExp#3 reduction  
16 addExp#3 reduction  
17 mulExp#3 reduction  
18 unaryExp#3 reduction  
19 primaryExp#3 reduction  
20 number#3 reduction  
21 INT#3 move  
22 mulExpAtom#; reduction  
23 addExpAtom#; reduction  
24 relExpAtom#; reduction  
25 eqExpAtom#; reduction

26 assignExpAtom#; reduction  
27 argVarDecl#; reduction  
28 ;#; move  
29 compUnit#void reduction  
30 funcDef#void reduction  
31 funcType#void reduction  
32 void#void move  
33 Ident#func move  
34 #( move  
35 funcFParams#) reduction  
36 )#) move  
37 block# reduction  
38 # move  
39 blockItem#int reduction  
40 decl#int reduction  
41 varDecl#int reduction  
42 bType#int reduction  
43 int#int move  
44 varDef#b reduction  
45 Ident#b move  
46 argVarDef#=# reduction  
47 =#=# move  
48 initVal#a reduction  
49 exp#a reduction  
50 assignExp#a reduction  
51 eqExp#a reduction  
52 relExp#a reduction  
53 addExp#a reduction  
54 mulExp#a reduction  
55 unaryExp#a reduction  
56 Ident#a move  
57 callFunc# error