测试结果展示

测试样例1

Plain Text

```
void main(){
    return 2;
}
```

词法分析结果

lex1.txt Shell

```
void <KW,4>
main <IDN,main>
( <SE,23>
) <SE,24>
{ <SE,25>
return <KW,5>
2 <INT,2>
; <SE,27>
} <SE,26>

```

语法分析结果

syntax1.txt Plain Text

```
0   program#void   reduction
1   compUnit#void  reduction
2   funcDef#void   reduction
3   funcType#void  reduction
4   void#void  move
5   Ident#Ident    move
6   (#(    move
7   funcFParams#)  reduction
8   )#)    move
9   block#{    reduction
10  {#{    move
11  blockItem#return   reduction
12  stmt#return    reduction
13  return#return  move
14  argExp#INT reduction
15  exp#INT    reduction
16  assignExp#INT  reduction
17  INT#INT    move
18  mulExpAtom#;   reduction
19  addExpAtom#;   reduction
20  relExpAtom#;   reduction
21  eqExpAtom#;    reduction
22  assignExpAtom#;    reduction
23  ;#;    move
24  blockItem#}    reduction
25  }#}    move
26  compUnit## reduction
27  ###    accept
```

测试样例2

Plain Text

```
int b = 3;

void main(){
    int a = 4;
    return a + b;
}
```

词法分析结果

lex2.txt Shell

```
int <KW,1>
b <IDN,b>
= <OP,12>
3 <INT,3>
; <SE,27>
void <KW,4>
main <IDN,main>
( <SE,23>
) <SE,24>
{ <SE,25>
int <KW,1>
a <IDN,a>
= <OP,12>
4 <INT,4>
; <SE,27>
return <KW,5>
a <IDN,a>
+ <OP,9>
b <IDN,b>
; <SE,27>
} <SE,26>

```

语法分析结果

syntax2.txt Plain Text

```
0   program#int    reduction
1   compUnit#int   reduction
2   decl#int   reduction
3   varDecl#int    reduction
4   bType#int  reduction
5   int#int    move
6   varDef#Ident   reduction
7   Ident#Ident    move
8   argVarDef#=    reduction
9   =#=    move
10  initVal#INT    reduction
11  exp#INT    reduction
12  assignExp#INT  reduction
13  INT#INT    move
14  mulExpAtom#;   reduction
15  addExpAtom#;   reduction
16  relExpAtom#;   reduction
17  eqExpAtom#;    reduction
18  assignExpAtom#;    reduction
19  argVarDecl#;   reduction
20  ;#;    move
21  compUnit#void  reduction
22  funcDef#void   reduction
23  funcType#void  reduction
24  void#void  move
25  Ident#Ident    move
26  (#(    move
27  funcFParams#)  reduction
28  )#)    move
29  block#{    reduction
30  {#{    move
31  blockItem#int  reduction
32  int#int    move
33  varDef#Ident   reduction
34  Ident#Ident    move
35  argVarDef#=    reduction
36  =#=    move
37  initVal#INT    reduction
38  exp#INT    reduction
39  assignExp#INT  reduction
40  INT#INT    move
41  mulExpAtom#;   reduction
42  addExpAtom#;   reduction
43  relExpAtom#;   reduction
44  eqExpAtom#;    reduction
```

```
45  assignExpAtom#;    reduction
46  argVarDecl#;   reduction
47  ;#;    move
48  blockItem#return   reduction
49  stmt#return    reduction
50  return#return  move
51  argExp#Ident   reduction
52  exp#Ident  reduction
53  assignExp#Ident    reduction
54  Ident#Ident    move
55  callFunc#+ reduction
56  mulExpAtom#+   reduction
57  addExpAtom#+   reduction
58  +#+    move
59  mulExp#Ident   reduction
60  Ident#Ident    move
61  callFunc#; reduction
62  mulExpAtom#;   reduction
63  addExpAtom#;   reduction
64  relExpAtom#;   reduction
65  eqExpAtom#;    reduction
66  assignExpAtom#;    reduction
67  ;#;    move
68  blockItem#}    reduction
69  }#}    move
70  compUnit## reduction
71  ###    accept
```

测试样例3

Plain Text

```
void main(){
    int a, b0, _c;
    a = 1;
    b0 = 2;
    _c = 3;
    return b0 + _c;
}
```

词法分析结果

lex3.txt Shell

```
void <KW,4>
main <IDN,main>
( <SE,23>
) <SE,24>
{ <SE,25>
int <KW,1>
a <IDN,a>
, <SE,28>
b0 <IDN,b0>
, <SE,28>
_c <IDN,_c>
; <SE,27>
a <IDN,a>
= <OP,12>
1 <INT,1>
; <SE,27>
b0 <IDN,b0>
= <OP,12>
2 <INT,2>
; <SE,27>
_c <IDN,_c>
= <OP,12>
3 <INT,3>
; <SE,27>
return <KW,5>
b0 <IDN,b0>
+ <OP,9>
_c <IDN,_c>
; <SE,27>
} <SE,26>

```

语法分析结果

syntax3.txt Plain Text

```
0   program#void   reduction
1   compUnit#void  reduction
2   funcDef#void   reduction
3   funcType#void  reduction
4   void#void  move
5   Ident#Ident    move
6   (#(    move
7   funcFParams#)  reduction
8   )#)    move
9   block#{    reduction
10  {#{    move
11  blockItem#int  reduction
12  int#int    move
13  varDef#Ident   reduction
14  Ident#Ident    move
15  argVarDef#,    reduction
16  argVarDecl#,   reduction
17  ,#,    move
18  varDef#Ident   reduction
19  Ident#Ident    move
20  argVarDef#,    reduction
21  argVarDecl#,   reduction
22  ,#,    move
23  varDef#Ident   reduction
24  Ident#Ident    move
25  argVarDef#;    reduction
26  argVarDecl#;   reduction
27  ;#;    move
28  blockItem#Ident    reduction
29  stmt#Ident reduction
30  exp#Ident  reduction
31  assignExp#Ident    reduction
32  Ident#Ident    move
33  callFunc#= reduction
34  mulExpAtom#=   reduction
35  addExpAtom#=   reduction
36  relExpAtom#=   reduction
37  eqExpAtom#=    reduction
38  assignExpAtom#=    reduction
39  =#=    move
40  eqExp#INT  reduction
41  INT#INT    move
42  mulExpAtom#;   reduction
43  addExpAtom#;   reduction
44  relExpAtom#;   reduction
```

```
45  eqExpAtom#;    reduction
46  assignExpAtom#;    reduction
47  ;#;    move
48  blockItem#Ident    reduction
49  stmt#Ident reduction
50  exp#Ident  reduction
51  assignExp#Ident    reduction
52  Ident#Ident    move
53  callFunc#= reduction
54  mulExpAtom#=   reduction
55  addExpAtom#=   reduction
56  relExpAtom#=   reduction
57  eqExpAtom#=    reduction
58  assignExpAtom#=    reduction
59  =#=    move
60  eqExp#INT  reduction
61  INT#INT    move
62  mulExpAtom#;   reduction
63  addExpAtom#;   reduction
64  relExpAtom#;   reduction
65  eqExpAtom#;    reduction
66  assignExpAtom#;    reduction
67  ;#;    move
68  blockItem#Ident    reduction
69  stmt#Ident reduction
70  exp#Ident  reduction
71  assignExp#Ident    reduction
72  Ident#Ident    move
73  callFunc#= reduction
74  mulExpAtom#=   reduction
75  addExpAtom#=   reduction
76  relExpAtom#=   reduction
77  eqExpAtom#=    reduction
78  assignExpAtom#=    reduction
79  =#=    move
80  eqExp#INT  reduction
81  INT#INT    move
82  mulExpAtom#;   reduction
83  addExpAtom#;   reduction
84  relExpAtom#;   reduction
85  eqExpAtom#;    reduction
86  assignExpAtom#;    reduction
87  ;#;    move
88  blockItem#return   reduction
89  stmt#return    reduction
90  return#return  move
91  argExp#Ident   reduction
92  exp#Ident  reduction
93
```

```
93  assignExp#Ident    reduction
94  Ident#Ident    move
95  callFunc#+ reduction
96  mulExpAtom#+   reduction
97  addExpAtom#+   reduction
98  +#+    move
99  mulExp#Ident   reduction
100 Ident#Ident    move
101 callFunc#; reduction
102 mulExpAtom#;   reduction
103 addExpAtom#;   reduction
104 relExpAtom#;   reduction
105 eqExpAtom#;    reduction
106 assignExpAtom#;    reduction
107 ;#;    move
108 blockItem#}    reduction
109 }#}    move
110 compUnit## reduction
111 ###    accept
```

测试样例4

Plain Text

```
void main(){
    const int a = 10, b = 5;
    return b;
}
```

词法分析结果

lex4.txt Shell

```
void <KW,4>
main <IDN,main>
( <SE,23>
) <SE,24>
{ <SE,25>
const <KW,6>
int <KW,1>
a <IDN,a>
= <OP,12>
10 <INT,10>
, <SE,28>
b <IDN,b>
= <OP,12>
5 <INT,5>
; <SE,27>
return <KW,5>
b <IDN,b>
; <SE,27>
} <SE,26>

```

语法分析结果

syntax4.txt Plain Text

```
0   program#void   reduction
1   compUnit#void  reduction
2   funcDef#void   reduction
3   funcType#void  reduction
4   void#void  move
5   Ident#Ident    move
6   (#(    move
7   funcFParams#)  reduction
8   )#)    move
9   block#{    reduction
10  {#{    move
11  blockItem#const    reduction
12  const#const    move
13  bType#int  reduction
14  int#int    move
15  constDef#Ident reduction
16  Ident#Ident    move
17  =#=    move
18  constInitVal#INT   reduction
19  constExp#INT   reduction
20  INT#INT    move
21  mulExpAtom#,   reduction
22  addExpAtom#,   reduction
23  relExpAtom#,   reduction
24  eqExpAtom#,    reduction
25  assignExpAtom#,    reduction
26  argConst#, reduction
27  ,#,    move
28  constDef#Ident reduction
29  Ident#Ident    move
30  =#=    move
31  constInitVal#INT   reduction
32  constExp#INT   reduction
33  INT#INT    move
34  mulExpAtom#;   reduction
35  addExpAtom#;   reduction
36  relExpAtom#;   reduction
37  eqExpAtom#;    reduction
38  assignExpAtom#;    reduction
39  argConst#; reduction
40  ;#;    move
41  blockItem#return   reduction
42  stmt#return    reduction
43  return#return  move
44  argExp#Ident   reduction
```

```
45  exp#Ident  reduction
46  assignExp#Ident    reduction
47  Ident#Ident    move
48  callFunc#; reduction
49  mulExpAtom#;   reduction
50  addExpAtom#;   reduction
51  relExpAtom#;   reduction
52  eqExpAtom#;    reduction
53  assignExpAtom#;    reduction
54  ;#;    move
55  blockItem#}    reduction
56  }#}    move
57  compUnit## reduction
58  ###    accept
```

测试样例5

```
Plain Text
```

```
struct my_struct1 {

  int value;

  void my_func(struct a b, int my_int, struct my_struct ccc) {}

struct my_struct me;
} s1, s2;

struct my_struct2 {

  int value;

  void my_func(struct a b, int my_int, struct my_struct ccc) {}

struct my_struct me;
} s;

struct my_struct3 {

  int value;

  void my_func(struct a b, int my_int, struct my_struct ccc) {}

  struct my_struct me;

union my_union mu;
};
```

词法分析结果

lex5.txt Shell

```
void <KW,4>
main <IDN,main>
( <SE,23>
) <SE,24>
{ <SE,25>
int <KW,1>
x <IDN,x>
= <OP,12>
2 <INT,2>
; <SE,27>
int <KW,1>
y <IDN,y>
= <OP,12>
3 <INT,3>
; <SE,27>
int <KW,1>
= <OP,12>
2 <INT,2>
; <SE,27>
switch <KW,31>
( <SE,23>
x <IDN,x>
) <SE,24>
{ <SE,25>
case <KW,33>
1 <INT,1>
: <SE,35>
{ <SE,25>
y <IDN,y>
= <OP,12>
10 <INT,10>
; <SE,27>
break <KW,34>
; <SE,27>
} <SE,26>
case <KW,33>
2 <INT,2>
: <SE,35>
{ <SE,25>
y <IDN,y>
= <OP,12>
20 <INT,20>
; <SE,27>
func <IDN,func>
( <SE,23>
```

```
y <IDN,y>
) <SE,24>
; <SE,27>
break <KW,34>
; <SE,27>
} <SE,26>
default <KW,32>
: <SE,35>
{ <SE,25>
return <KW,5>
0 <INT,0>
; <SE,27>
} <SE,26>
} <SE,26>
} <SE,26>
```

语法分析结果

syntax_struct.txt Plain Text

```
0   program#struct reduction
1   compUnit#struct    reduction
2   structDef#struct   reduction
3   struct#struct  move
4   structType#Ident   reduction
5   Ident#Ident    move
6   {#{    move
7   structBlockElem#int    reduction
8   int#int    move
9   varDef#Ident   reduction
10  Ident#Ident    move
11  argVarDef#;    reduction
12  argVarDecl#;   reduction
13  ;#;    move
14  structBlockElem#void   reduction
15  void#void  move
16  Ident#Ident    move
17  (#(    move
18  funcFParams#struct reduction
19  funcFParam#struct  reduction
20  struct#struct  move
21  structType#Ident   reduction
22  Ident#Ident    move
23  Ident#Ident    move
24  argFunctionF#, reduction
25  ,#,    move
26  funcFParam#int reduction
27  int#int    move
28  Ident#Ident    move
29  argFunctionF#, reduction
30  ,#,    move
31  funcFParam#struct  reduction
32  struct#struct  move
33  structType#Ident   reduction
34  Ident#Ident    move
35  Ident#Ident    move
36  argFunctionF#) reduction
37  )#)    move
38  block#{    reduction
39  {#{    move
40  blockItem#}    reduction
41  }#}    move
42  structBlockElem#struct reduction
43  structDecl#struct  reduction
44  struct#struct  move
```

```
45  structType#Ident   reduction
46  Ident#Ident    move
47  Ident#Ident    move
48  ;#;    move
49  structBlockElem#}  reduction
50  }#}    move
51  structVar#Ident    reduction
52  Ident#Ident    move
53  argStructVar#, reduction
54  ,#,    move
55  Ident#Ident    move
56  argStructVar#; reduction
57  ;#;    move
58  compUnit#struct    reduction
59  structDef#struct   reduction
60  struct#struct  move
61  structType#Ident   reduction
62  Ident#Ident    move
63  {#{    move
64  structBlockElem#int    reduction
65  int#int    move
66  varDef#Ident   reduction
67  Ident#Ident    move
68  argVarDef#;    reduction
69  argVarDecl#;   reduction
70  ;#;    move
71  structBlockElem#void   reduction
72  void#void  move
73  Ident#Ident    move
74  (#(    move
75  funcFParams#struct reduction
76  funcFParam#struct  reduction
77  struct#struct  move
78  structType#Ident   reduction
79  Ident#Ident    move
80  Ident#Ident    move
81  argFunctionF#, reduction
82  ,#,    move
83  funcFParam#int reduction
84  int#int    move
85  Ident#Ident    move
86  argFunctionF#, reduction
87  ,#,    move
88  funcFParam#struct  reduction
89  struct#struct  move
90  structType#Ident   reduction
91  Ident#Ident    move
92  Ident#Ident    move
```

```
93  argFunctionF#) reduction
94  )#)    move
95  block#{    reduction
96  {#{    move
97  blockItem#}    reduction
98  }#}    move
99  structBlockElem#struct reduction
100 structDecl#struct  reduction
101 struct#struct  move
102 structType#Ident   reduction
103 Ident#Ident    move
104 Ident#Ident    move
105 ;#;    move
106 structBlockElem#}  reduction
107 }#}    move
108 structVar#Ident    reduction
109 Ident#Ident    move
110 argStructVar#; reduction
111 ;#;    move
112 compUnit#struct    reduction
113 structDef#struct   reduction
114 struct#struct  move
115 structType#Ident   reduction
116 Ident#Ident    move
117 {#{    move
118 structBlockElem#int    reduction
119 int#int    move
120 varDef#Ident   reduction
121 Ident#Ident    move
122 argVarDef#;    reduction
123 argVarDecl#;   reduction
124 ;#;    move
125 structBlockElem#void   reduction
126 void#void  move
127 Ident#Ident    move
128 (#(    move
129 funcFParams#struct reduction
130 funcFParam#struct  reduction
131 struct#struct  move
132 structType#Ident   reduction
133 Ident#Ident    move
134 Ident#Ident    move
135 argFunctionF#, reduction
136 ,#,    move
137 funcFParam#int reduction
138 int#int    move
139 Ident#Ident    move
140 argFunctionF#, reduction
141
```

```
141 ,#,    move
142 funcFParam#struct  reduction
143 struct#struct  move
144 structType#Ident   reduction
145 Ident#Ident    move
146 Ident#Ident    move
147 argFunctionF#) reduction
148 )#)    move
149 block#{    reduction
150 {#{    move
151 blockItem#}    reduction
152 }#}    move
153 structBlockElem#struct reduction
154 structDecl#struct  reduction
155 struct#struct  move
156 structType#Ident   reduction
157 Ident#Ident    move
158 Ident#Ident    move
159 ;#;    move
160 structBlockElem#union  reduction
161 unionDecl#union    reduction
162 union#union    move
163 unionType#Ident    reduction
164 Ident#Ident    move
165 Ident#Ident    move
166 ;#;    move
167 structBlockElem#}  reduction
168 }#}    move
169 structVar#;    reduction
170 ;#;    move
171 compUnit## reduction
172 ###    accept
```

测试样例6

Plain Text

```
void main() {
    int x = 2;
    int y = 3;
    int =2;
    switch (x) {
        case 1: {
            y = 10;
            break;
        }
        case 2: {
            y = 20;
            func(y);
            break;
        }
        default: {
            return 0;
        }
    }
}
```

词法分析结果

lex6.txt Shell

```
struct <KW,29>
my_struct1 <IDN,my_struct1>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
void <KW,4>
my_func <IDN,my_func>
( <SE,23>
struct <KW,29>
a <IDN,a>
b <IDN,b>
, <SE,28>
int <KW,1>
my_int <IDN,my_int>
, <SE,28>
struct <KW,29>
my_struct <IDN,my_struct>
ccc <IDN,ccc>
) <SE,24>
{ <SE,25>
} <SE,26>
struct <KW,29>
my_struct <IDN,my_struct>
me <IDN,me>
; <SE,27>
} <SE,26>
s1 <IDN,s1>
, <SE,28>
s2 <IDN,s2>
; <SE,27>
struct <KW,29>
my_struct2 <IDN,my_struct2>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
void <KW,4>
my_func <IDN,my_func>
( <SE,23>
struct <KW,29>
a <IDN,a>
b <IDN,b>
, <SE,28>
int <KW,1>
```

```
my_int <IDN,my_int>
, <SE,28>
struct <KW,29>
my_struct <IDN,my_struct>
ccc <IDN,ccc>
) <SE,24>
{ <SE,25>
} <SE,26>
struct <KW,29>
my_struct <IDN,my_struct>
me <IDN,me>
; <SE,27>
} <SE,26>
s <IDN,s>
; <SE,27>
struct <KW,29>
my_struct3 <IDN,my_struct3>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
void <KW,4>
my_func <IDN,my_func>
( <SE,23>
struct <KW,29>
a <IDN,a>
b <IDN,b>
, <SE,28>
int <KW,1>
my_int <IDN,my_int>
, <SE,28>
struct <KW,29>
my_struct <IDN,my_struct>
ccc <IDN,ccc>
) <SE,24>
{ <SE,25>
} <SE,26>
struct <KW,29>
my_struct <IDN,my_struct>
me <IDN,me>
; <SE,27>
union <KW,30>
my_union <IDN,my_union>
mu <IDN,mu>
; <SE,27>
} <SE,26>
; <SE,27>
```

语法分析结果

syntax_switch.txt Plain Text

```
0   program#void   reduction
1   compUnit#void  reduction
2   funcDef#void   reduction
3   funcType#void  reduction
4   void#void  move
5   Ident#Ident    move
6   (#(    move
7   funcFParams#)  reduction
8   )#)    move
9   block#{    reduction
10  {#{    move
11  blockItem#int  reduction
12  int#int    move
13  varDef#Ident   reduction
14  Ident#Ident    move
15  argVarDef#=    reduction
16  =#=    move
17  initVal#INT    reduction
18  exp#INT    reduction
19  assignExp#INT  reduction
20  INT#INT    move
21  mulExpAtom#;   reduction
22  addExpAtom#;   reduction
23  relExpAtom#;   reduction
24  eqExpAtom#;    reduction
25  assignExpAtom#;    reduction
26  argVarDecl#;   reduction
27  ;#;    move
28  blockItem#int  reduction
29  int#int    move
30  varDef#Ident   reduction
31  Ident#Ident    move
32  argVarDef#=    reduction
33  =#=    move
34  initVal#INT    reduction
35  exp#INT    reduction
36  assignExp#INT  reduction
37  INT#INT    move
38  mulExpAtom#;   reduction
39  addExpAtom#;   reduction
40  relExpAtom#;   reduction
41  eqExpAtom#;    reduction
42  assignExpAtom#;    reduction
43  argVarDecl#;   reduction
44  ;#;    move
```

```
45  blockItem#int  reduction
46  int#int    move
47  varDef#=   error
```

测试样例7

Plain Text

```
union my_union1 {

int value;
float weight;
} u1, u2;


union my_union2 {

int value;

union my_union1 my_union;

struct my_struct1 my_struct;
} u;

union my_union3 {

int value;
char id;
};
```

词法分析结果

lex7.txt Shell

```
union <KW,30>
my_union1 <IDN,my_union1>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
float <KW,2>
weight <IDN,weight>
; <SE,27>
} <SE,26>
u1 <IDN,u1>
, <SE,28>
u2 <IDN,u2>
; <SE,27>
union <KW,30>
my_union2 <IDN,my_union2>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
union <KW,30>
my_union1 <IDN,my_union1>
my_union <IDN,my_union>
; <SE,27>
struct <KW,29>
my_struct1 <IDN,my_struct1>
my_struct <IDN,my_struct>
; <SE,27>
} <SE,26>
u <IDN,u>
; <SE,27>
union <KW,30>
my_union3 <IDN,my_union3>
{ <SE,25>
int <KW,1>
value <IDN,value>
; <SE,27>
char <KW,3>
id <IDN,id>
; <SE,27>
} <SE,26>
; <SE,27>

```

语法分析结果

syntax_union.txt Plain Text

```
0   program#union  reduction
1   compUnit#union reduction
2   unionDef#union reduction
3   union#union    move
4   unionType#Ident    reduction
5   Ident#Ident    move
6   {#{    move
7   unionBlockElem#int reduction
8   int#int    move
9   varDef#Ident   reduction
10  Ident#Ident    move
11  argVarDef#;    reduction
12  argVarDecl#;   reduction
13  ;#;    move
14  unionBlockElem#float   reduction
15  float#float    move
16  varDef#Ident   reduction
17  Ident#Ident    move
18  argVarDef#;    reduction
19  argVarDecl#;   reduction
20  ;#;    move
21  unionBlockElem#}   reduction
22  }#}    move
23  unionVar#Ident reduction
24  Ident#Ident    move
25  argUnionVar#,  reduction
26  ,#,    move
27  Ident#Ident    move
28  argUnionVar#;  reduction
29  ;#;    move
30  compUnit#union reduction
31  unionDef#union reduction
32  union#union    move
33  unionType#Ident    reduction
34  Ident#Ident    move
35  {#{    move
36  unionBlockElem#int reduction
37  int#int    move
38  varDef#Ident   reduction
39  Ident#Ident    move
40  argVarDef#;    reduction
41  argVarDecl#;   reduction
42  ;#;    move
43  unionBlockElem#union   reduction
44  unionDecl#union    reduction
```

```
union#union    move
unionType#Ident    reduction
Ident#Ident    move
Ident#Ident    move
;#;    move
unionBlockElem#struct  reduction
struct#struct  move
structType#Ident   reduction
Ident#Ident    move
Ident#Ident    move
;#;    move
unionBlockElem#}   reduction
}#}    move
unionVar#Ident reduction
Ident#Ident    move
argUnionVar#;  reduction
;#;    move
compUnit#union reduction
unionDef#union reduction
union#union    move
unionType#Ident    reduction
Ident#Ident    move
{#{    move
unionBlockElem#int reduction
int#int    move
varDef#Ident   reduction
Ident#Ident    move
argVarDef#;    reduction
argVarDecl#;   reduction
;#;    move
unionBlockElem#char    reduction
char#char  move
varDef#Ident   reduction
Ident#Ident    move
argVarDef#;    reduction
argVarDecl#;   reduction
;#;    move
unionBlockElem#}   reduction
}#}    move
unionVar#; reduction
;#;    move
compUnit## reduction
###    accept
```