测试样例以及词法、语法分析输出结果

一、文件说明：*.sy 为测试样例源文件，*.tok 为词法分析输出文件，*.spe 为语法分析输出结果。

二、测试结果。

1. accept1，合法样例，主要检测赋值和 if-else 语句。

```
void main() {
    int a;
    int B;
    const float PI = 3.14;

    a = 10;
    B = 5 * a;

    int result_sum;
    result_sum = a + B;

    if (result_sum ⩾ 60) {
        return 1;
    } else {
        int remainder;
        remainder = B % 10;

        float value = a / 2.0;
        if (value ≠ PI) {
            return 0;
        }
    }

    return 0;
}
```

图表 1 accept1.sy

```
void    <KW,2>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
int <KW,1>
a   <IDN,a>
;   <SE,54>
int <KW,1>
B   <IDN,B>
;   <SE,54>
const   <KW,4>
float   <KW,6>
PI  <IDN,PI>
=   <OP,35>
3.14    <FLOAT,3.14>
;   <SE,54>
a   <IDN,a>
=   <OP,35>
10  <INT,10>
;   <SE,54>
B   <IDN,B>
=   <OP,35>
5   <INT,5>
*   <OP,32>
```

图表 2 accept1.tok

```
void#void   move
funcType#Ident  reduction
Ident#main  move
(#( move
)#) move
{#{ move
int#int move
bType#Ident reduction
Ident#a move
varDef#;    reduction
;#; move
varDecl#int reduction
decl#int    reduction
blockItem#int   reduction
int#int move
bType#Ident reduction
Ident#B move
varDef#;    reduction
;#; move
varDecl#const   reduction
decl#const  reduction
blockItem#const reduction
const#const move
float#float move
bType#Ident reduction
```

图表 3 accept1.spe

2. accept2，合法样例，主要检测比较复杂的运算操作。

```
int main() {
    int x;
    int Y_val;
    int z;

    x = 15;
    Y_val = 4;

    z = x + Y_val * 2 - 10 / 5 % 3;

    if (z > 20) {
        return z;
    }

    return 0;
}
```

图表 4 accept2.sy

```
int <KW,1>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
int <KW,1>
x   <IDN,x>
;   <SE,54>
int <KW,1>
Y_val   <IDN,Y_val>
;   <SE,54>
int <KW,1>
z   <IDN,z>
;   <SE,54>
x   <IDN,x>
=   <OP,35>
15  <INT,15>
;   <SE,54>
Y_val   <IDN,Y_val>
```

图表 5 accept2.tok

```
int#int move
bType#Ident reduction
Ident#main  move
(#( move
)#) move
{#{ move
int#int move
bType#Ident reduction
Ident#x move
varDef#;    reduction
;#; move
varDecl#int reduction
decl#int    reduction
blockItem#int   reduction
int#int move
bType#Ident reduction
Ident#Y_val move
varDef#;    reduction
;#; move
varDecl#int reduction
decl#int    reduction
blockItem#int   reduction
```

图表 6 accept2.spe

3. accept3，合法样例，主要检测嵌套 if 语句。

```
void main() {
    const float RATE = 0.05;
    float principal = 1000.0;
    float interest;
    int time = 5;

    interest = principal * RATE * time;

    if (interest == 250.0) {
        float total = principal + interest;

        if (total ⩽ 1250.0) {
            return;
        }
    }

    if (principal < 999.99 || time ≠ 5) {
        return;
    }
}
```

图表 7 accept3.sy

```
void    <KW,2>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
const   <KW,4>
float   <KW,6>
RATE    <IDN,RATE>
=   <OP,35>
0.05    <FLOAT,0.05>
;   <SE,54>
float   <KW,6>
principal   <IDN,principal>
=   <OP,35>
1000.0  <FLOAT,1000.0>
;   <SE,54>
float   <KW,6>
interest    <IDN,interest>
;   <SE,54>
int <KW,1>
time    <IDN,time>
=   <OP,35>
5   <INT,5>
;   <SE,54>
interest    <IDN,interest>
=   <OP,35>
principal   <IDN,principal>
```

图表 8 accept3.tok

```
1   void#void   move
2   funcType#Ident  reduction
3   Ident#main  move
4   (#( move
5   )#) move
6   {#{ move
7   const#const move
8   float#float move
9   bType#Ident reduction
10  Ident#RATE  move
11  =#= move
12  floatConst#0.05 move
13  number#;    reduction
14  primaryExp#;    reduction
15  unaryExp#;  reduction
16  mulExp#;    reduction
17  addExp#;    reduction
18  constExp#;  reduction
19  constInitVal#;  reduction
20  constDef#;  reduction
21  ;#; move
22  constDecl#float reduction
23  decl#float  reduction
```

图表 9 accept3.spe

4. accept4，合法样例，主要检测多层嵌套的 if-else 语句。

```
int main() {
    int status = 1;
    int count = 10;

    if (status == 1 && count ≥ 10) {
        float avg = 5.5;

        if (avg < 5.0 || avg > 6.0) {
            return 0;
        } else {
            return 1;
        }
    } else {
        if (count < 0) {
            return -1;
        }
    }

    return 2;
}
```

图表 10 accept4.sy

```
int <KW,1>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
int <KW,1>
status  <IDN,status>
=   <OP,35>
1   <INT,1>
;   <SE,54>
int <KW,1>
count   <IDN,count>
=   <OP,35>
10  <INT,10>
;   <SE,54>
if  <KW,7>
(   <SE,50>
status  <IDN,status>
==  <OP,38>
1   <INT,1>
&&  <OP,42>
count   <IDN,count>
⩾   <OP,40>
10  <INT,10>
)   <SE,51>
{   <SE,52>
float   <KW,6>
```

图表 11 accept4.tok

```
1   int#int move
2   bType#Ident reduction
3   Ident#main  move
4   (#( move
5   )#) move
6   {#{ move
7   int#int move
8   bType#Ident reduction
9   Ident#status    move
10  =#= move
11  IntConst#1  move
12  number#;    reduction
13  primaryExp#;    reduction
14  unaryExp#;  reduction
15  mulExp#;    reduction
16  addExp#;    reduction
17  relExp#;    reduction
18  eqExp#; reduction
19  lAndExp#;   reduction
20  lOrExp#;    reduction
21  exp#;   reduction
22  initVal#;   reduction
23  varDef#;    reduction
24  ;#; move
25  varDecl#int reduction
```

图表 12 accept4.spe

5. accept5，合法样例，主要检测多个函数声明。

```
int add_numbers(int a, int b) {
    int sum;
    sum = a + b;
    return sum;
}

void main() {
    int x = 5;
    int y = 7;
    int result;

    result = add_numbers(x, y);

    if (result == 12) {
        return;
    }
}
```

图表 13 accept5.sy

```
int <KW,1>
add_numbers <IDN,add_numbers>
(   <SE,50>
int <KW,1>
a   <IDN,a>
,   <SE,55>
int <KW,1>
b   <IDN,b>
)   <SE,51>
{   <SE,52>
int <KW,1>
sum <IDN,sum>
;   <SE,54>
sum <IDN,sum>
=   <OP,35>
a   <IDN,a>
+   <OP,30>
b   <IDN,b>
;   <SE,54>
return  <KW,3>
sum <IDN,sum>
;   <SE,54>
}   <SE,53>
void    <KW,2>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
```

图表 14 accept5.tok

```
int#int move
bType#Ident reduction
Ident#add_numbers   move
(#( move
int#int move
bType#Ident reduction
Ident#a move
funcFParam#,    reduction
funcFParams#,   reduction
,#, move
int#int move
bType#Ident reduction
Ident#b move
funcFParam#)    reduction
funcFParams#)   reduction
)#) move
{#{ move
int#int move
bType#Ident reduction
Ident#sum   move
varDef#;    reduction
;#; move
varDecl#Ident   reduction
decl#Ident  reduction
blockItem#Ident reduction
Ident#sum   move
lVal#=  reduction
=#= move
Ident#a move
lVal#+  reduction
```

图表 15 accept5.spe

6. accept6，合法样例，主要检测全局变量的声明。

```
const float PI = 3.14;

void calculate_area(float radius) {
    float area;
    area = PI * radius * radius;

    if (area ⩾ 3.0 && area ⩽ 3.2) {
        return;
    }
}

void main() {
    float R = 1.0;

    calculate_area(R);

    float large_radius = 10.0;

    if (large_radius ≠ R) {
        calculate_area(large_radius);
    }

    return;
}
```

图表 16 accept6.sy

```
const    <KW,4>
float    <KW,6>
PI   <IDN,PI>
=    <OP,35>
3.14     <FLOAT,3.14>
;    <SE,54>
void     <KW,2>
calculate_area   <IDN,calculate_area>
(    <SE,50>
float    <KW,6>
radius   <IDN,radius>
)    <SE,51>
{    <SE,52>
float    <KW,6>
area     <IDN,area>
;    <SE,54>
area     <IDN,area>
=    <OP,35>
PI   <IDN,PI>
*    <OP,32>
radius   <IDN,radius>
*    <OP,32>
radius   <IDN,radius>
;    <SE,54>
if   <KW,7>
(    <SE,50>
area     <IDN,area>
```

图表 17 accept6.tok

```
const#const move
float#float move
bType#Ident reduction
Ident#PI    move
=#= move
floatConst#3.14 move
number#;    reduction
primaryExp#;    reduction
unaryExp#;  reduction
mulExp#;    reduction
addExp#;    reduction
constExp#;  reduction
constInitVal#;  reduction
constDef#;  reduction
;#; move
constDecl#void  reduction
decl#void   reduction
compUnit#void   reduction
void#void   move
funcType#Ident  reduction
Ident#calculate_area    move
(#( move
float#float move
bType#Ident reduction
Ident#radius    move
funcFParam#)    reduction
funcFParams#)   reduction
)#) move
{#{ move
float#float move
```

图表 18 accept6.spe

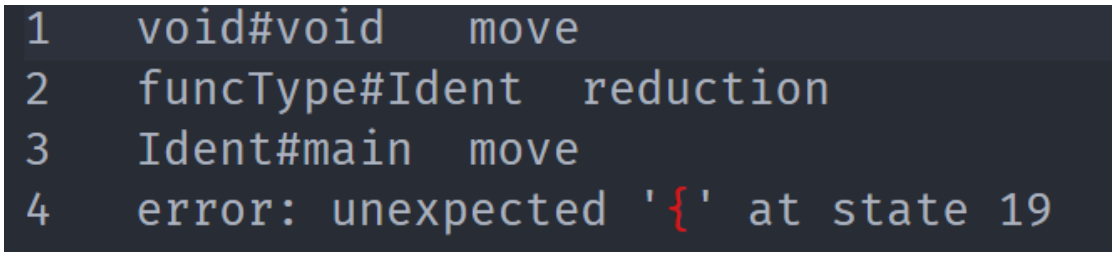
7. refuse1，非法样例，主要检测变量的合法命名和条件语句。

```
void main {
    int 9test = 10;

    int result = 5 * 2

    float PI = 3.14;
    PI = 4.0;

    if (result &&) {
        return 0;
    }

    return 1;
}
```

图表 19 refuse1.sy

```
void    <KW,2>
main    <IDN,main>
{   <SE,52>
int <KW,1>
9   <INT,9>
test    <IDN,test>
=   <OP,35>
10  <INT,10>
;   <SE,54>
int <KW,1>
result  <IDN,result>
=   <OP,35>
5   <INT,5>
*   <OP,32>
2   <INT,2>
float   <KW,6>
PI  <IDN,PI>
=   <OP,35>
3.14    <FLOAT,3.14>
;   <SE,54>
PI  <IDN,PI>
=   <OP,35>
4.0 <FLOAT,4.0>
;   <SE,54>
if  <KW,7>
(   <SE,50>
result  <IDN,result>
&&  <OP,42>
)   <SE,51>
```

图表 20 refuse1.tok



图表 21 refuse1.spe

8. refuse2，非法样例，主要检测变量的合法命名。

```
void main() {
    int 1alpha = 5;
    int salary@ = 100;
    float value = 5.;

    if (1alpha == 5) {
        return 0;
    }
}
```

图表 22 refuse2.sy

```
void    <KW,2>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
int <KW,1>
1   <INT,1>
alpha   <IDN,alpha>
=   <OP,35>
5   <INT,5>
;   <SE,54>
int <KW,1>
salary  <IDN,salary>
@   <ERROR,201>
=   <OP,35>
100 <INT,100>
;   <SE,54>
float   <KW,6>
value   <IDN,value>
=   <OP,35>
5   <INT,5>
.   <ERROR,201>
;   <SE,54>
if  <KW,7>
(   <SE,50>
1   <INT,1>
alpha   <IDN,alpha>
==  <OP,38>
5   <INT,5>
)   <SE,51>
```

图表 23 refuse2.tok

```
1   void#void   move
2   funcType#Ident  reduction
3   Ident#main  move
4   (#( move
5   )#) move
6   {#{ move
7   int#int move
8   error: unexpected '1' at state 11
```

图表 24 refuse2.spe

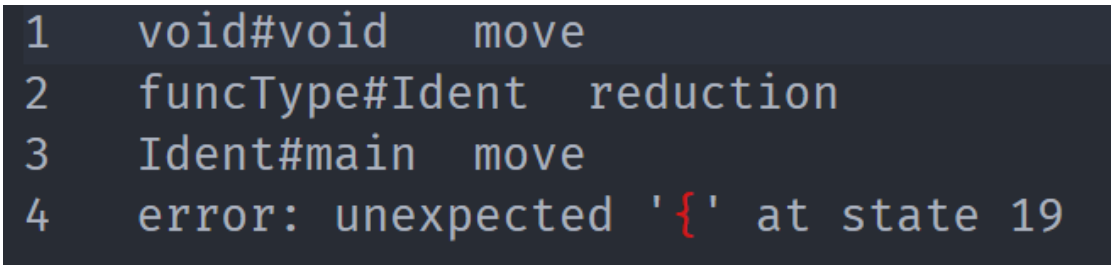
9. refuse3，非法样例，主要检测未合拢的‘{’和‘(’。

```
void main {
    int a = 10

    if (a == 10 {
        return a;
    }

    else
        int b = 5;
```

图表 25 refuse3.sy

```
void    <KW,2>
main    <IDN,main>
{   <SE,52>
int <KW,1>
a   <IDN,a>
=   <OP,35>
10  <INT,10>
if  <KW,7>
(   <SE,50>
a   <IDN,a>
==  <OP,38>
10  <INT,10>
{   <SE,52>
return  <KW,3>
a   <IDN,a>
;   <SE,54>
}   <SE,53>
else    <KW,8>
int <KW,1>
b   <IDN,b>
=   <OP,35>
5   <INT,5>
;   <SE,54>
```

图表 26 refuse3.tok



图表 27 refuse3.spe

10. refuse4，非法样例，主要检测非法运算符。

```
int main() {
    int val = 10;
    int result;

    result = val * ;

    if (val $ 10) {
        return 0;
    }

    result = 5 + return 0;

    return 1;
}
```

图表 28 refuse4.sy

```
int <KW,1>
main    <IDN,main>
(   <SE,50>
)   <SE,51>
{   <SE,52>
int <KW,1>
val <IDN,val>
=   <OP,35>
10  <INT,10>
;   <SE,54>
int <KW,1>
result  <IDN,result>
;   <SE,54>
result  <IDN,result>
=   <OP,35>
val <IDN,val>
*   <OP,32>
;   <SE,54>
if  <KW,7>
(   <SE,50>
val <IDN,val>
$   <ERROR,201>
10  <INT,10>
)   <SE,51>
{   <SE,52>
return  <KW,3>
0   <INT,0>
;   <SE,54>
```

图表 29 refuse4.tok

```
int#int move
bType#Ident reduction
Ident#main  move
(#( move
)#) move
{#{ move
int#int move
bType#Ident reduction
Ident#val   move
=#= move
IntConst#10 move
number#;    reduction
primaryExp#;    reduction
unaryExp#;  reduction
mulExp#;    reduction
addExp#;    reduction
relExp#;    reduction
eqExp#; reduction
lAndExp#;   reduction
lOrExp#;    reduction
exp#;   reduction
initVal#;   reduction
varDef#;    reduction
;#; move
varDecl#int reduction
decl#int    reduction
blockItem#int   reduction
int#int move
bType#Ident reduction
Ident#result    move
```

图表 30 refuse4.spe