天津大学


# 编译原理
# 词法分析器/语法分析器
# 测试报告

学　　院 智能与计算/求是学部
专　　业 计算机科学与技术
组　　长 杨亦凡 **3019234258**
组　　员 李自安 **3019207257**
组　　员 石昊　**3019208051**
组　　员 华溢　**3019244091**

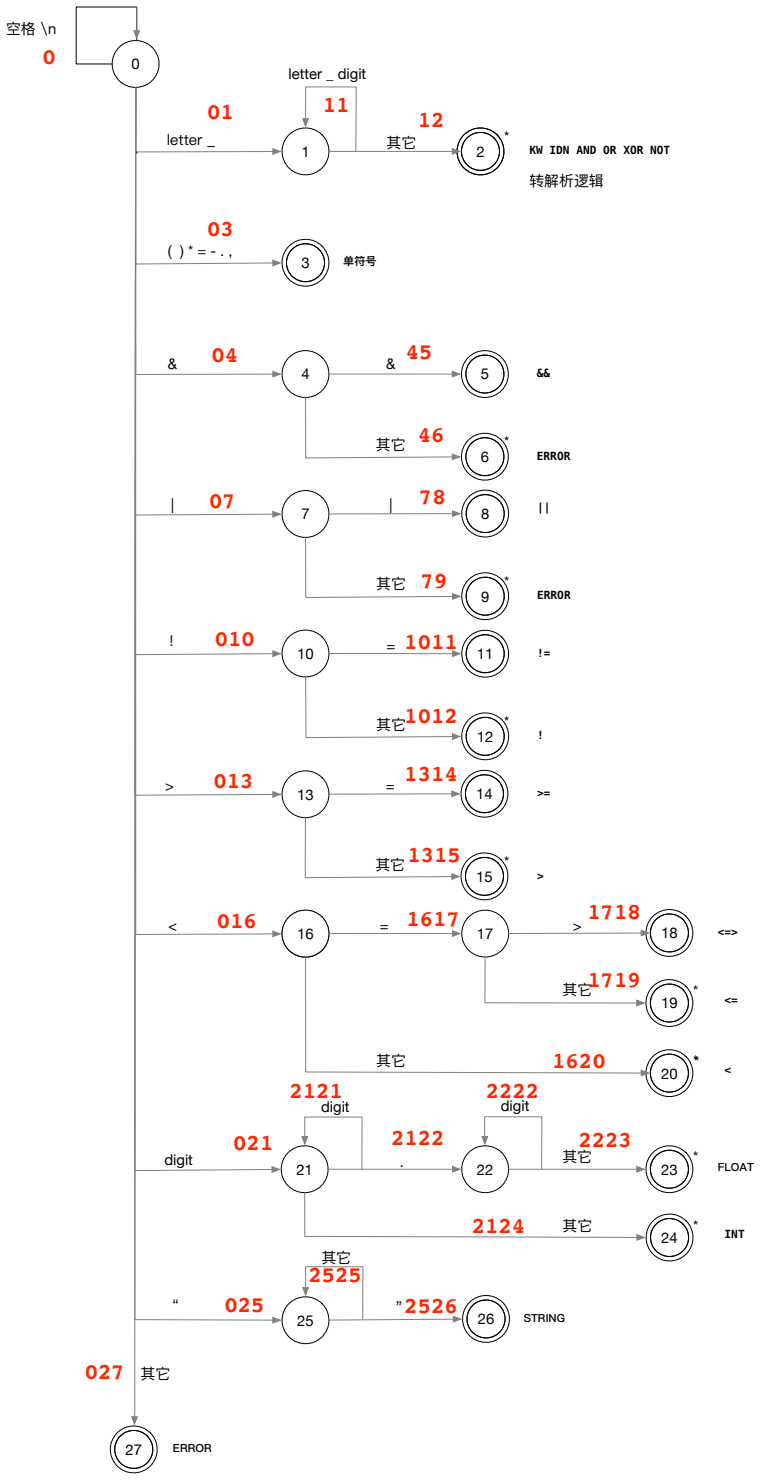
# 目录

# 1 词法分析测试

## 1.1 测试方法

对于词法分析器，我们采用**白盒测试法**，覆盖标准选择为**判定/条件覆盖**，针对词法分析器的有限状态机逻辑设计测试用例，验证分析运行结果与预期结果。

## 1.2 测试用例设计



针对词法分析器的有限状态机逻辑1.2设计测试用例，按照判定/条件覆盖标准，应满足以下覆盖情况：

条件（每条弧以相邻两个状态序号标识）

0:空格
0:\n
0:非空格
0:非\n
01:letter
01:_
01:非letter
01:非_
11:letter
11:_
11:digit
11:非letter
11:非_
11:digit
12:任意字符
03:(
03:)
03:*
03:=
03:-
03:.
03:,
03:非(
03:非)
03:非*
03:非=
03:非-
03:非.
03:非,
04:&
04:非&
45:&
45:非&
46:任意字符
07:|

07:非|
78:|
78:非|
79:任意字符
010:!
010:非!
1011:=
1011:非=
1012:任意字符
013:>
013:非>
1314:=
1314:非=
1315:任意字符
016:<
016:非<
1617:=
1617:非=
1718:>
1718:非>
1719:任意字符
1620:任意字符
021:任意数字
021:任意非数字
2121:任意数字
2121:任意非数字
2122:.
2122:非.
2222:任意数字
2222:任意非数字
2223:任意字符
2124:任意字符
025:"
025:非"
2525:任意字符
2526:"
2526:非"
027:@
027:$

027:%
027:^
027:&
027:+
027:\
027:`
027:~
027:[
027:{
027:]
027:}
027:;
027::
027:''
027:/
027:?

应执行路径:

0->0->27
0->1->2
0->1->1->2
0->3
0->4->5
0->4->6
0->7->8
0->7->9
0->10->11
0->10->12
0->13->14
0->13->15
0->16->17->18
0->16->17->19
0->16->20
0->21->22->23
0->21->24
0->25->26
0->27

选择用例

[(a),(a  <IDN,a>)]

```
[(b),(b   <IDN,b>)]
[(c),(c   <IDN,c>)]
[(d),(d   <IDN,d>)]
[(e),(e   <IDN,e>)]
[(f),(f   <IDN,f>)]
[(g),(g   <IDN,g>)]
[(h),(h   <IDN,h>)]
[(i),(i   <IDN,i>)]
[(j),(j   <IDN,j>)]
[(k),(k   <IDN,k>)]
[(l),(l   <IDN,l>)]
[(m),(m   <IDN,m>)]
[(n),(n   <IDN,n>)]
[(o),(o   <IDN,o>)]
[(p),(p   <IDN,p>)]
[(q),(q   <IDN,q>)]
[(r),(r   <IDN,r>)]
[(s),(s   <IDN,s>)]
[(t),(t   <IDN,t>)]
[(u),(u   <IDN,u>)]
[(v),(v   <IDN,v>)]
[(w),(w   <IDN,w>)]
[(x),(x   <IDN,x>)]
[(y),(y   <IDN,y>)]
[(z),(z   <IDN,z>)]
[(A),(A   <IDN,A>)]
[(B),(B   <IDN,B>)]
[(C),(C   <IDN,C>)]
[(D),(D   <IDN,D>)]
[(E),(E   <IDN,E>)]
[(F),(F   <IDN,F>)]
[(G),(G   <IDN,G>)]
[(H),(H   <IDN,H>)]
[(I),(I   <IDN,I>)]
[(J),(J   <IDN,J>)]
[(K),(K   <IDN,K>)]
[(L),(L   <IDN,L>)]
[(M),(M   <IDN,M>)]
[(N),(N   <IDN,N>)]
```

```
[(O),(O   <IDN,O>)]
[(P),(P   <IDN,P>)]
[(Q),(Q   <IDN,Q>)]
[(R),(R   <IDN,R>)]
[(S),(S   <IDN,S>)]
[(T),(T   <IDN,T>)]
[(U),(U   <IDN,U>)]
[(V),(V   <IDN,V>)]
[(W),(W   <IDN,W>)]
[(X),(X   <IDN,X>)]
[(Y),(Y   <IDN,Y>)]
[(Z),(Z   <IDN,Z>)]
[(_),(_   <IDN,_>)]
[(_,t,j,u,_,1,8,9,5,_,_),(_tju_1895__   <IDN,_tju_1895__>)]
[((),((   <SE,1>)]
[()),()   <SE,2>)]
[(*),(*   <KW,5>)]
[(=),(=   <OP,1>)]
[(-),(-   <OP,15>)]
[(.),(.   <OP,16>)]
[(,),(,   <SE,3>)]
[(&,&),(&&   <OP,9>)]
[(&,a),(Error &)]
[(&),(Error &)]
[(|,|),(||   <OP,11>)]
[(|,a),(Error |)]
[(|),(Error |)]
[(!,=),(!=   <OP,6>)]
[(!),(!   <OP,14>)]
[(>,=),(>=   <OP,4>)]
[(>),(>   <OP,2>)]
[(<,=,>),(<=>   <OP,7>)]
[(<,=),(<=   <OP,5>)]
[(<),(<   <OP,3>)]
[(0,.),(0.   <FLOAT,0.>)]
[(0,.,1),(0.1   <FLOAT,0.1>)]
[(1,.,1,2,3),(1.123   <FLOAT,1.123>)]
[(1,.,2,3,4,0),(1.2340   <FLOAT,1.2340>)]
[(0),(0   <INT,0>)]
```

```
[(1),(1   <INT,1>)]
[(2),(2   <INT,2>)]
[(3),(3   <INT,3>)]
[(4),(4   <INT,4>)]
[(5),(5   <INT,5>)]
[(6),(6   <INT,6>)]
[(7),(7   <INT,7>)]
[(8),(8   <INT,8>)]
[(9),(9   <INT,9>)]
[(0,1,2,3),(0123   <INT,0123>)]
[(1,2,3,4),(1234   <INT,1234>)]
[(1,2,3,0),(1230   <INT,1230>)]
[(",t,j,u,t,j,u,"),("tjutju"   <STRING,tjutju>)]
[(","),(""   <STRING,>)]
[(@),(Error else @)]
[($),(Error else $)]
[(%),(Error else %)]
[(^),(Error else ^)]
[(&),(Error &)]
[(+),(Error else +)]
[(\),(Error else \)]
[(`),(Error else `)]
[(~),(Error else ~)]
[([),(Error else [)]
[({),(Error else {)]
[(]),(Error else ])]
[(}),(Error else })]
[(;),(Error else ;)]
[(:),(Error else :)]
[(','),(Error else ')]
[(/),(Error else /)]
[(?),(Error else ?)]
[(G,R,O,U,P),(GROUP   <IDN,GROUP>)]
[(O,R,D,E,R),(ORDER   <IDN,ORDER>)]
[(G,R,O,U,P, ,B,Y, ,a),(GROUP BY   <KW,24>,a   <IDN,a>)]
[(O,R,D,E,R, ,B,Y, ,a),(ORDER BY   <KW,27>,a   <IDN,a>)]
[(G,R,O,U,P, ,B,Y),(GROUP   <IDN,GROUP>,BY   <IDN,BY>)]
[(O,R,D,E,R, ,B,Y),(ORDER   <IDN,ORDER>,BY   <IDN,BY>)]
```

## 1.3 测试结果

```
a   <IDN,a>     _   <IDN,_>
b   <IDN,b>     _tju_1895__   <IDN,_tju_1895__>
c   <IDN,c>     (   <SE,1>
d   <IDN,d>     )   <SE,2>
e   <IDN,e>     *   <KW,5>
f   <IDN,f>     =   <OP,1>
g   <IDN,g>     -   <OP,15>
h   <IDN,h>     .   <OP,16>
i   <IDN,i>     ,   <SE,3>
j   <IDN,j>     &&   <OP,9>
k   <IDN,k>     ||   <OP,11>
l   <IDN,l>     !=   <OP,6>
m   <IDN,m>     !   <OP,14>
n   <IDN,n>     >=   <OP,4>
o   <IDN,o>     >   <OP,2>
p   <IDN,p>     <=>   <OP,7>
q   <IDN,q>     <=   <OP,5>
r   <IDN,r>     <   <OP,3>
s   <IDN,s>     0.   <FLOAT,0.>
t   <IDN,t>     0.1   <FLOAT,0.1>
u   <IDN,u>     1.123   <FLOAT,1.123>
v   <IDN,v>     1.2340   <FLOAT,1.2340>
w   <IDN,w>     0   <INT,0>
x   <IDN,x>     1   <INT,1>
y   <IDN,y>     2   <INT,2>
z   <IDN,z>     3   <INT,3>
A   <IDN,A>     4   <INT,4>
B   <IDN,B>     5   <INT,5>
C   <IDN,C>     6   <INT,6>
D   <IDN,D>     7   <INT,7>
E   <IDN,E>     8   <INT,8>
F   <IDN,F>     9   <INT,9>
G   <IDN,G>     0123   <INT,0123>
H   <IDN,H>     1234   <INT,1234>
I   <IDN,I>     1230   <INT,1230>
J   <IDN,J>     "tjutju"   <STRING,tjutju>
K   <IDN,K>     ""   <STRING,>
L   <IDN,L>     GROUP   <IDN,GROUP>
M   <IDN,M>     ORDER   <IDN,ORDER>
N   <IDN,N>     GROUP BY   <KW,24>
O   <IDN,O>     a   <IDN,a>
P   <IDN,P>     ORDER BY   <KW,27>
Q   <IDN,Q>     a   <IDN,a>
R   <IDN,R>     GROUP   <IDN,GROUP>
S   <IDN,S>     BY   <IDN,BY>
T   <IDN,T>     ORDER   <IDN,ORDER>
U   <IDN,U>     BY   <IDN,BY>
V   <IDN,V>
W   <IDN,W>
X   <IDN,X>
Y   <IDN,Y>
Z   <IDN,Z>
```

测试错误输出：

```
Error &
Error &
Error |
Error |
Error else
@
Error else
$
Error else
%
Error else
^
Error &
Error else
+
Error else
\
Error else
`
Error else
~
Error else
[
Error else
{
Error else
]
Error else
}
Error else
;
Error else
:
Error else
'
Error else
/
Error else
?
```

经比对，所有测试样例均符合预期输出。

# 2 语法分析测试

## 2.1 测试方法

对于语法分析器，我们采用**黑盒测试法**，具体方法选择为**等价分类法**，针对输入条件划分为有效等价类和无效等价类，从等价类中选出具有代表性的用例进行测试，验证分析运行结果与预期结果。

值得注意的是，为了方便验证，测试样例的输出我们采用二值表示（T 代表规约成功/ F 代表规约失败）。

## 2.2 测试用例设计

根据 SQL- - 所支持的语法规则，我们设计了百余个测试样例，测试范围覆盖全部关键词和全部的语法规则。

为了方便展示，本报告仅以 GROUP BY 为例展示测试流程与结果.

等价类设计：

| 输入条件 | 合理的等价类 | 不合理的等价类 |
| --- | --- | --- |
| 有无WHERE | 有(1)、无(2) | WHERE语法错误(3) |
| 有无HAVING | 有(4)、无(5) | HAVING语法错误(6) |
| 有无ORDER BY | 有(7)、无(8) | ORDER BY语法错误(9) |
| 有无SUM | 有(10)、无(11) | SUM语法错误(12) |
| 有无AVG | 有(13)、无(14) | AVG语法错误(15) |

针对等价类设计测试用例，测试样例按照测试「语法类型-等价类序号-预期结果」命名：

**testcase-GROUPBY-(2)(5)(8)(11)(14)-T.sql:**

```
SELECT id, name, MAX(daily_typing_pages)
FROM employee_tbl
GROUP BY name
```

**testcase-GROUPBY-(1)(5)(8)(11)(13)-T.sql:**

```
SELECT name,AVG(movie_1.length)
FROM movie_1,exec_1
WHERE  movie_1._2_= exec_1.we_10
GROUP BY name
```

**testcase-GROUPBY-(2)(4)(8)(10)(14)-T.sql:**

```
SELECT region, SUM(population), SUM(area)
FROM bbc
GROUP BY region
HAVING SUM(area)>1000000
```

**testcase-GROUPBY-(1)(4)(7)(10)(13)-T.sql:**

```
SELECT class, SUM(female), AVG(a3)
FROM students
WHERE class = "2"
```

```
GROUP BY class
HAVING a = "GROUP"
ORDER BY b
```

**testcase-GROUPBY-(1)(4)(8)(10)(14)-T.sql:**

```
SELECT from_._1_,SUM(from_._2_)
FROM from_ JOIN _1A ON from_._1_=_1A.cr7
WHERE from_._2_>1 AND from_._3_<3.1415926 OR 1.25 IS NOT NULL
GROUP BY from_._2_
HAVING from_._3_="ORDER BY #><=="
```

**testcase-GROUPBY-(3)-F.sql:**

```
SELECT t.GROUP BY_HAHA,MIN(t.order) AS ORDERs
FROM t,_t_
WHERE
GROUP BY ORDERs
HAVING SUM(t.order)<-12.340
```

**testcase-GROUPBY-(6)-F.sql:**

```
SELECT t.GROUP BY_HAHA,MIN(t.order) AS ORDERs
FROM t,_t_
WHERE NOT BY_HAHA IS NULL
GROUP BY ORDERs
HAVING
```

**testcase-GROUPBY-(9)-F.sql:**

```
SELECT class, SUM(female), AVG(a3)
FROM students
WHERE class = "2"
GROUP BY class
HAVING a = "GROUP"
ORDER BY
```

**testcase-GROUPBY-(12)-F.sql:**

```
SELECT t.GROUP BY_HAHA,MIN(t.order) AS ORDERs
FROM t,_t_
WHERE NOT BY_HAHA IS NULL
GROUP BY ORDERs
HAVING SUM()<-12.340
```

**testcase-GROUPBY-(15)-F.sql:**

```
SELECT class, SUM(female), AVG()
FROM students
WHERE class = "2"
GROUP BY class
HAVING a = "GROUP"
ORDER BY b
```

## 2.3 测试结果

“parsed successfully” 表示该样例成功通过了词法与语法分析器的测试，并且返回了相应的输出结果到./output 文件夹中。

“parsed failed” 表示该样例成功被词法分析器解析，但语法分析器规约失败，

测试结果输出：

```
testing testcase-GROUPBY-(2)(5)(8)(11)(14)-T.sql.....
test parsed successfully
testing testcase-GROUPBY-(1)(5)(8)(11)(13)-T.sql.....
test parsed successfully
testing testcase-GROUPBY-(2)(4)(8)(10)(14)-T.sql.....
test parsed successfully
testing testcase-GROUPBY-(1)(4)(7)(10)(13)-T.sql.....
test parsed successfully
testing testcase-GROUPBY-(1)(4)(8)(10)(14)-T.sql.....
test parsed successfully
testing testcase-GROUPBY-(3)-F.sql.....
test parsed failed
testing testcase-GROUPBY-(6)-F.sql.....
test parsed failed
testing testcase-GROUPBY-(9)-F.sql.....
test parsed failed
testing testcase-GROUPBY-(10)-F.sql.....
test parsed failed
testing testcase-GROUPBY-(11)-F.sql.....
test parsed failed
```

token 序列输出（仅给出 testcase-GROUPBY-(1)(4)(7)(10)(13)-T.sql）：

```
SELECT   <KW,1>
class   <IDN,class>
,   <SE,3>
SUM   <KW,21>
(   <SE,1>
female   <IDN,female>
)   <SE,2>
,   <SE,3>
AVG   <KW,20>
(   <SE,1>
a3   <IDN,a3>
)   <SE,2>
FROM   <KW,2>
students   <IDN,students>
WHERE   <KW,3>
class   <IDN,class>
=   <OP,1>
"2"   <STRING,2>
GROUP BY   <KW,24>
class   <IDN,class>
HAVING   <KW,25>
a   <IDN,a>
=   <OP,1>
"GROUP"   <STRING,GROUP>
ORDER BY   <KW,27>
b   <IDN,b>
```

规约序列输出（仅给出 testcase-GROUPBY-(1)(4)(7)(10)(13)-T.sql）:

```
1       /       ##SELECT        move
2       14      SELECT#IDN      reduction
3       /       unionType#IDN   move
4       48      IDN#,   reduction
5       51      uid#,   reduction
6       49      dottedId#,      reduction
7       35      fullColumnName#,        reduction
8       31      elementNameAlias#,      reduction
9       28      selectElement#, reduction
10      /       selectElementHead#,     move
11      /       ,#SUM   move
12      105     SUM#(   reduction
13      /       function#(      move
14      14      (#IDN   reduction
15      /       unionType#IDN   move
16      48      IDN#)   reduction
17      51      uid#)   reduction
18      49      dottedId#)      reduction
19      /       fullColumnName#)        move
20      101     )#,     reduction
21      100     aggregateWindowedFunction#,     reduction
22      35      functionCall#,  reduction
23      32      elementNameAlias#,      reduction
24      /       selectElement#, move
25      /       ,#AVG   move
26      102     AVG#(   reduction
27      /       function#(      move
28      14      (#IDN   reduction
29      /       unionType#IDN   move
30      48      IDN#)   reduction
31      51      uid#)   reduction
32      49      dottedId#)      reduction
33      /       fullColumnName#)        move
34      101     )#FROM  reduction
35      100     aggregateWindowedFunction#FROM  reduction
36      35      functionCall#FROM       reduction
37      32      elementNameAlias#FROM   reduction
38      30      selectElement#FROM      reduction
39      29      selectElementListRec#FROM       reduction
40      29      selectElementListRec#FROM       reduction
41      26      selectElementListRec#FROM       reduction
42      /       selectElements#FROM     move
43      /       FROM#IDN        move
44      48      IDN#WHERE       reduction
45      44      uid#WHERE       reduction
46      35      tableName#WHERE reduction
47      42      elementNameAlias#WHERE  reduction
48      41      tableSourceItem#WHERE   reduction
49      39      joinParts#WHERE reduction
50      38      tableSource#WHERE       reduction
51      36      tableSourceListRec#WHERE        reduction
52      /       tableSources#WHERE      move
53      /       WHERE#IDN       move
54      48      IDN#=   reduction
55      51      uid#=   reduction
56      49      dottedId#=      reduction
57      76      fullColumnName#=        reduction
58      /       expressionAtom#=        move
59      85      =#STRING        reduction
60      /       comparisonOperator#STRING       move
61      97      STRING#GROUP_BY reduction
62      79      stringLiteral#GROUP_BY  reduction
63      75      constant#GROUP_BY       reduction
64      74      expressionAtom#GROUP_BY reduction
65      72      predicateRight#GROUP_BY reduction
66      73      predicate#GROUP_BY      reduction
67      72      predicateRight#GROUP_BY reduction
68      61      predicate#GROUP_BY      reduction
69      58      expressionRight#GROUP_BY        reduction
70      132     expression#GROUP_BY     reduction
71      18      whereExpression#GROUP_BY        reduction
72      /       fromClause#GROUP_BY     move
73      /       GROUP_BY#IDN    move
74      48      IDN#HAVING      reduction
75      51      uid#HAVING      reduction
76      49      dottedId#HAVING reduction
77      76      fullColumnName#HAVING   reduction
78      74      expressionAtom#HAVING   reduction
79      72      predicateRight#HAVING   reduction
80      61      predicate#HAVING        reduction
81      58      expressionRight#HAVING  reduction
82      56      expression#HAVING       reduction
83      54      expressionRec#HAVING    reduction
84      20      expressions#HAVING      reduction
85      /       groupByClause#HAVING    move
86      /       HAVING#IDN      move
87      48      IDN#=   reduction
88      51      uid#=   reduction
89      49      dottedId#=      reduction
90      76      fullColumnName#=        reduction
91      /       expressionAtom#=        move
92      85      =#STRING        reduction
93      /       comparisonOperator#STRING       move
94      97      STRING#ORDER_BY reduction
95      79      stringLiteral#ORDER_BY  reduction
96      75      constant#ORDER_BY       reduction
97      74      expressionAtom#ORDER_BY reduction
98      72      predicateRight#ORDER_BY reduction
99      73      predicate#ORDER_BY      reduction
100     72      predicateRight#ORDER_BY reduction
101     61      predicate#ORDER_BY      reduction
102     58      expressionRight#ORDER_BY        reduction
103     22      expression#ORDER_BY     reduction
104     /       havingClause#ORDER_BY   move
105     /       ORDER_BY#IDN    move
106     48      IDN##   reduction
107     51      uid##   reduction
108     49      dottedId##      reduction
109     76      fullColumnName##        reduction
110     74      expressionAtom##        reduction
111     72      predicateRight##        reduction
112     61      predicate##     reduction
113     58      expressionRight##       reduction
114     56      expression##    reduction
115     54      expressionRec## reduction
116     24      expressions##   reduction
117     17      orderByClause## reduction
118     15      selectClause##  reduction
119     8       querySpecification##    reduction
120     6       unionStatements##       reduction
121     2       selectStatement##       reduction
122     1       dmlStatement##  reduction
123     /       root##  accept
```

经比对，所有测试样例均符合预期输出。