**项目测试报告**

**任务一**

**1、准备工作**

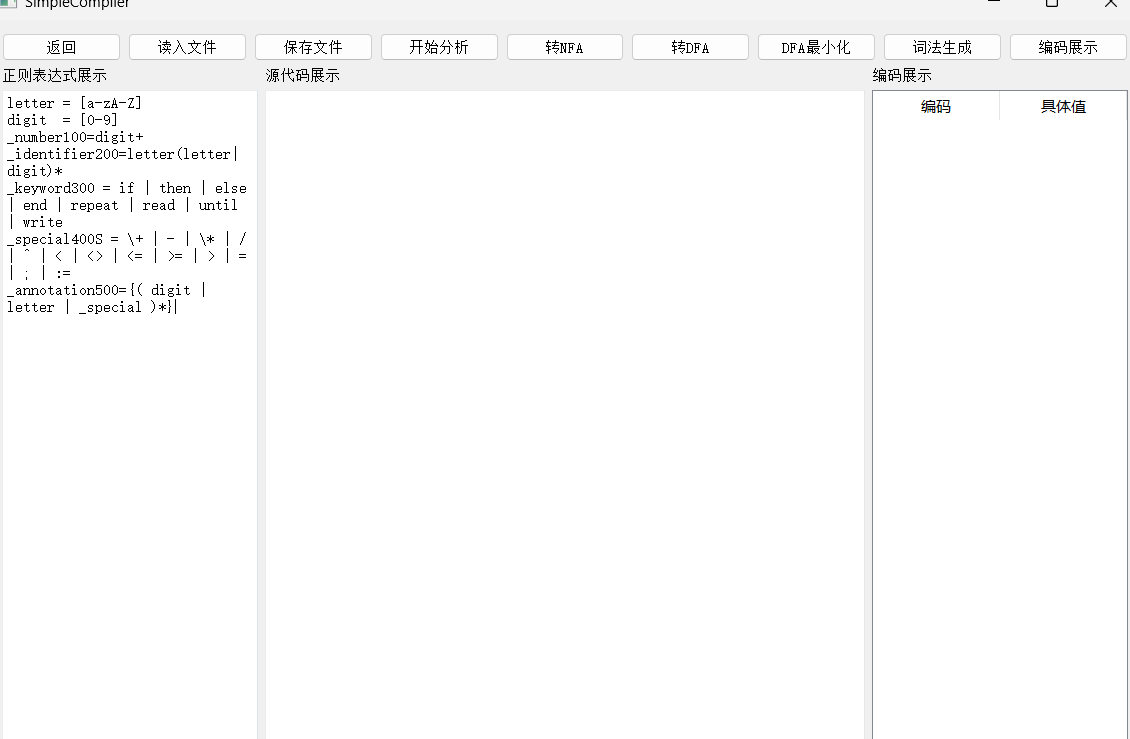
**tiny的正则表达式：**

|  |
| --- |
| letter = [a-zA-Z]  digit = [0-9]  \_number100=digit+  \_identifier200=letter(letter|digit)\*  \_keyword300 = if | then | else | end | repeat | read | until | write  \_special400S = \+ | - | \\* | / | ^ | < | <> | <= | >= | > | = | ; | :=  \_annotation500={( digit | letter | \_special )\*} |

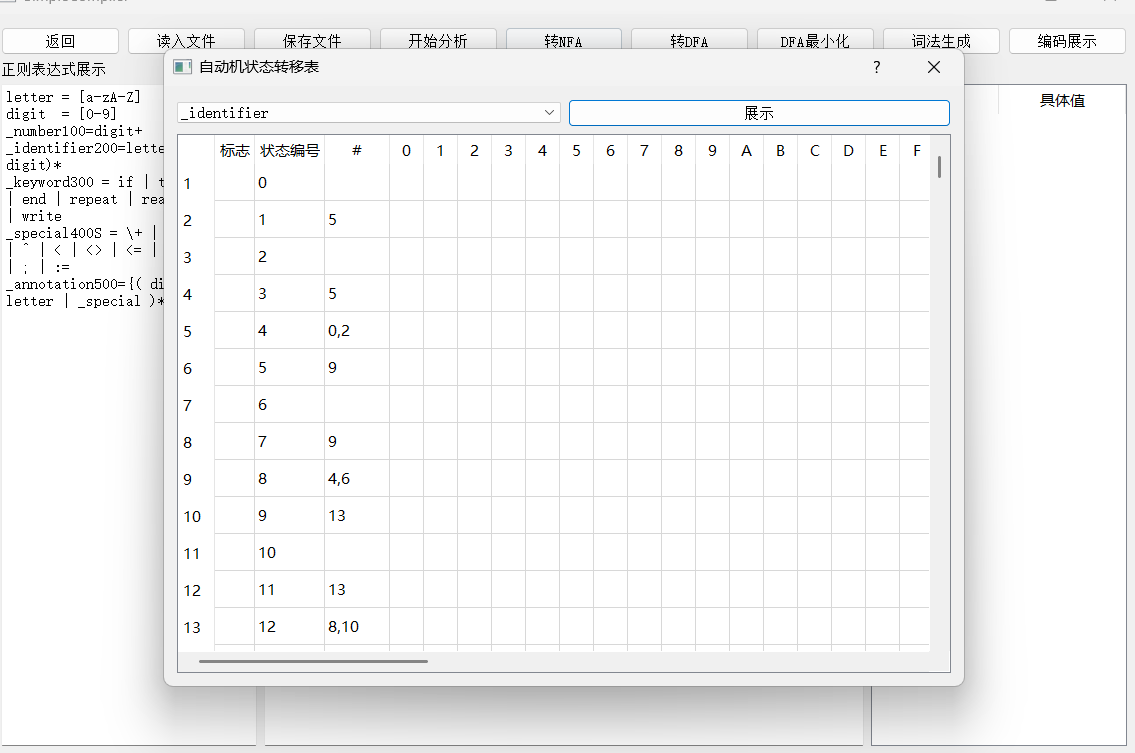
**sample.tny文件：**

|  |
| --- |
| read x ;  {  多行注释  }  if x < 10 then  y := x \* 3 / 2 ;  write y  end  else  repeat  x := x + 1 - 2  until x = 0  end ;  {符号部分}  if x <= 99 then  x := 1  end  else  x := 1  end ;  if x <> 100 then  x := 1  end ;  if x > 99 then  x := 1  end ;  if x >= 100 then  x := x ^ 2  end  else  x := 0  end |

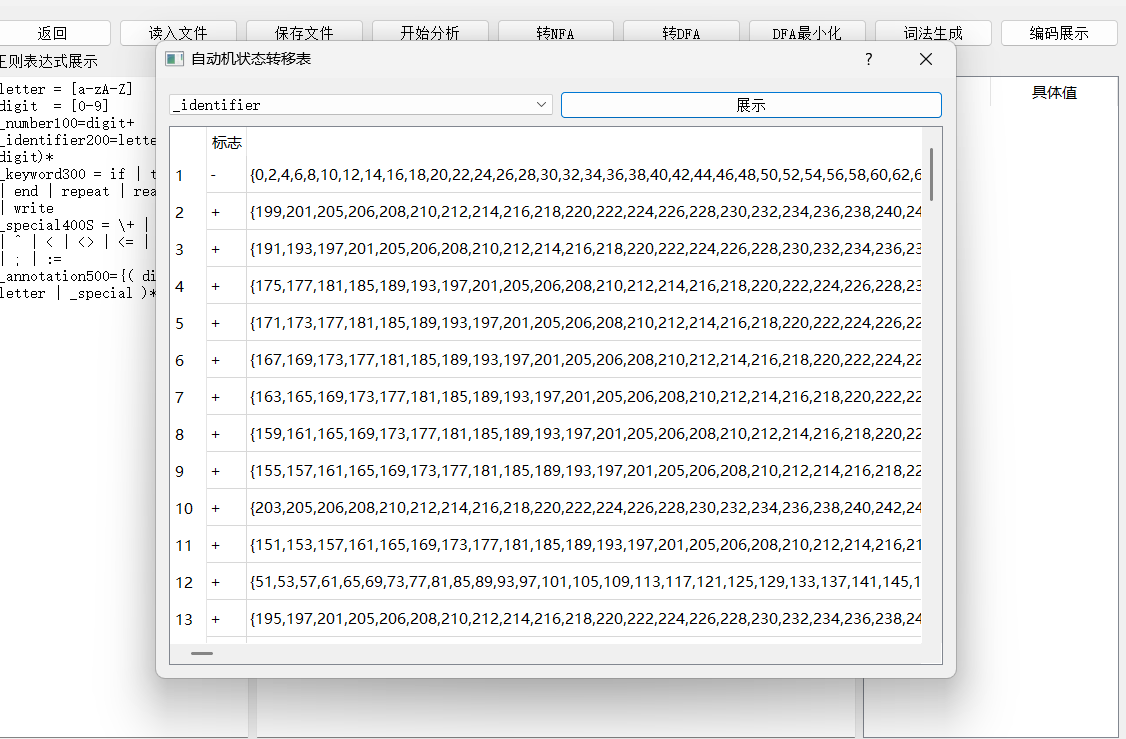
**输入正则表达式：**



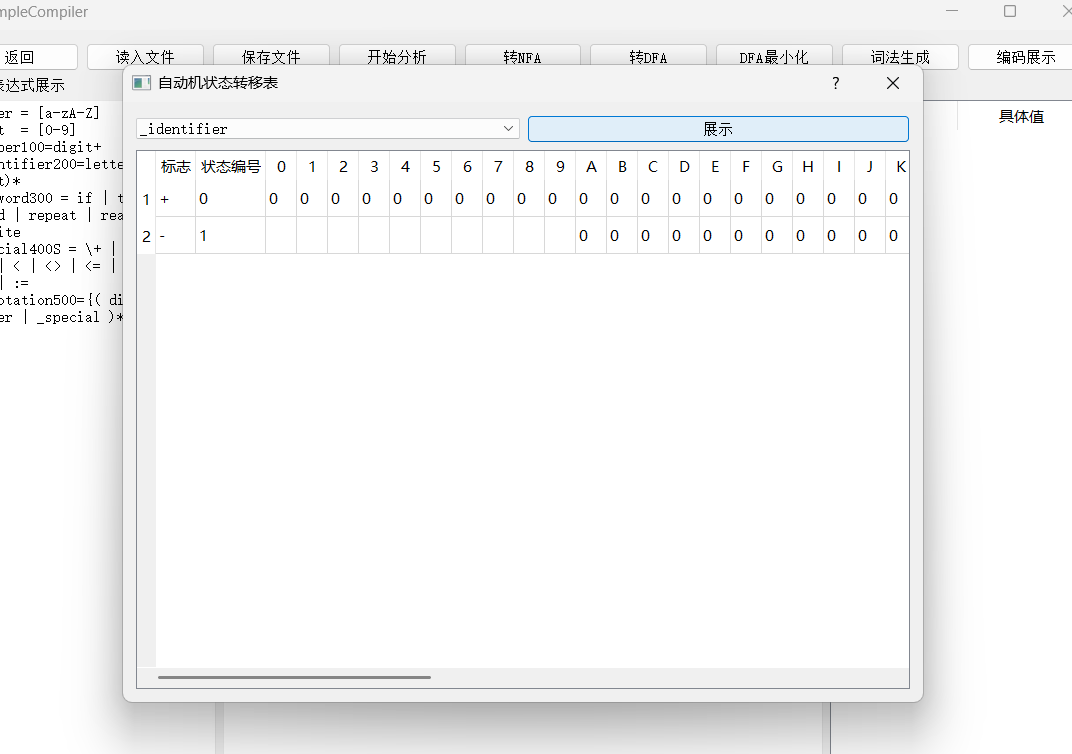
**2、NFA**



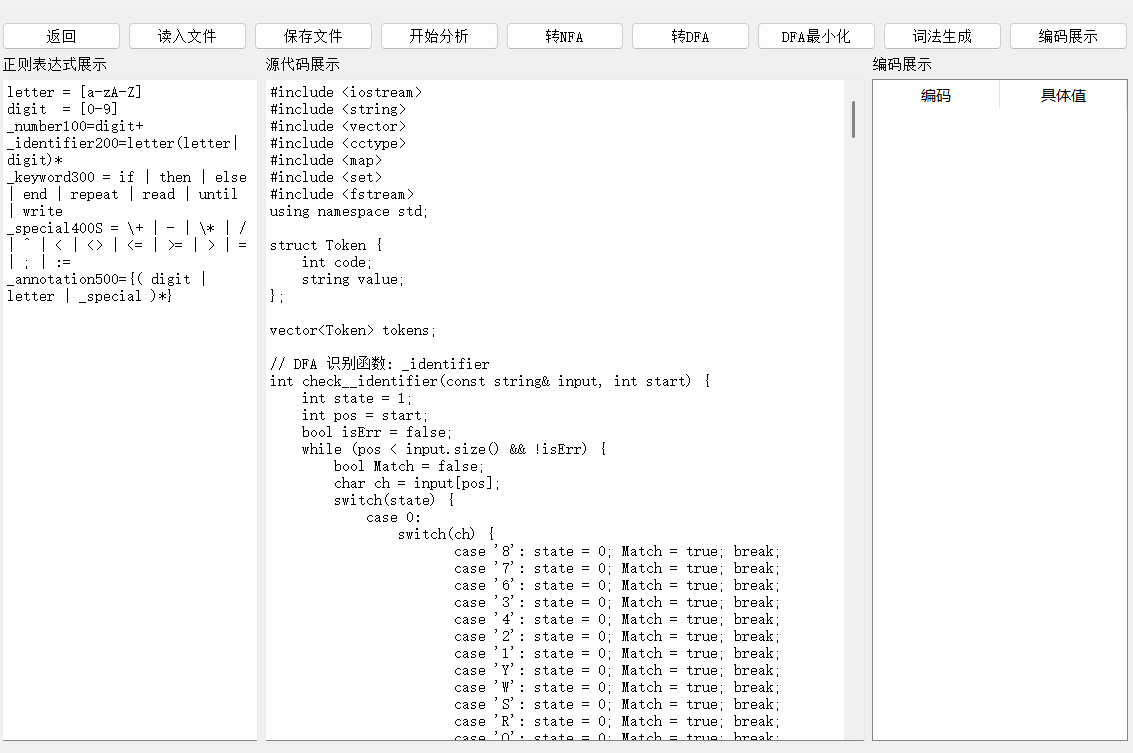
**3、DFA图**



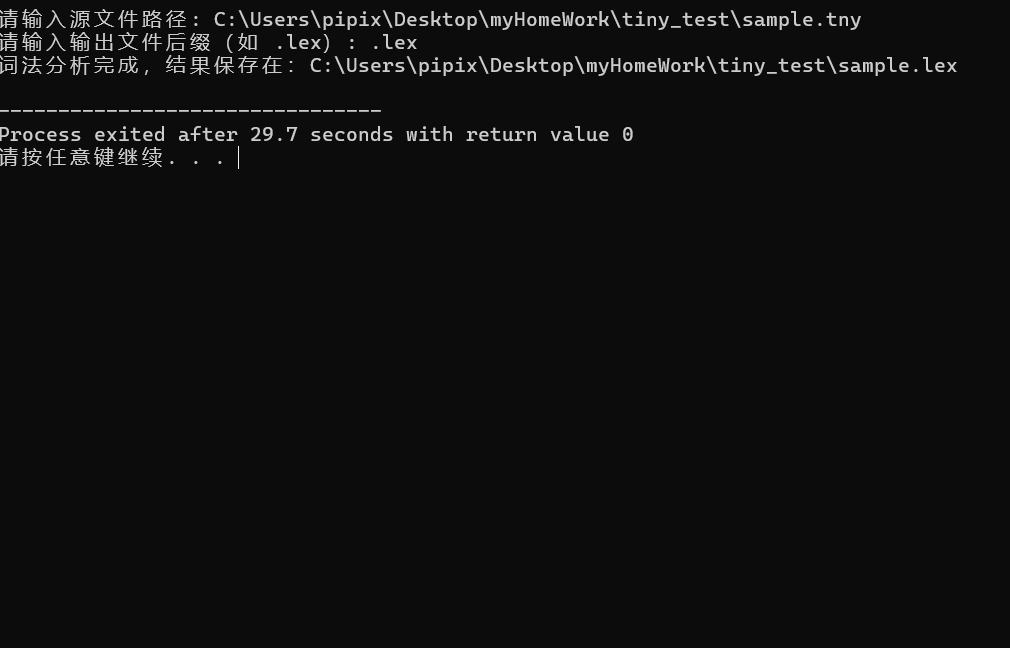
**4、DFA图最小化**



**5、生成词法程序**



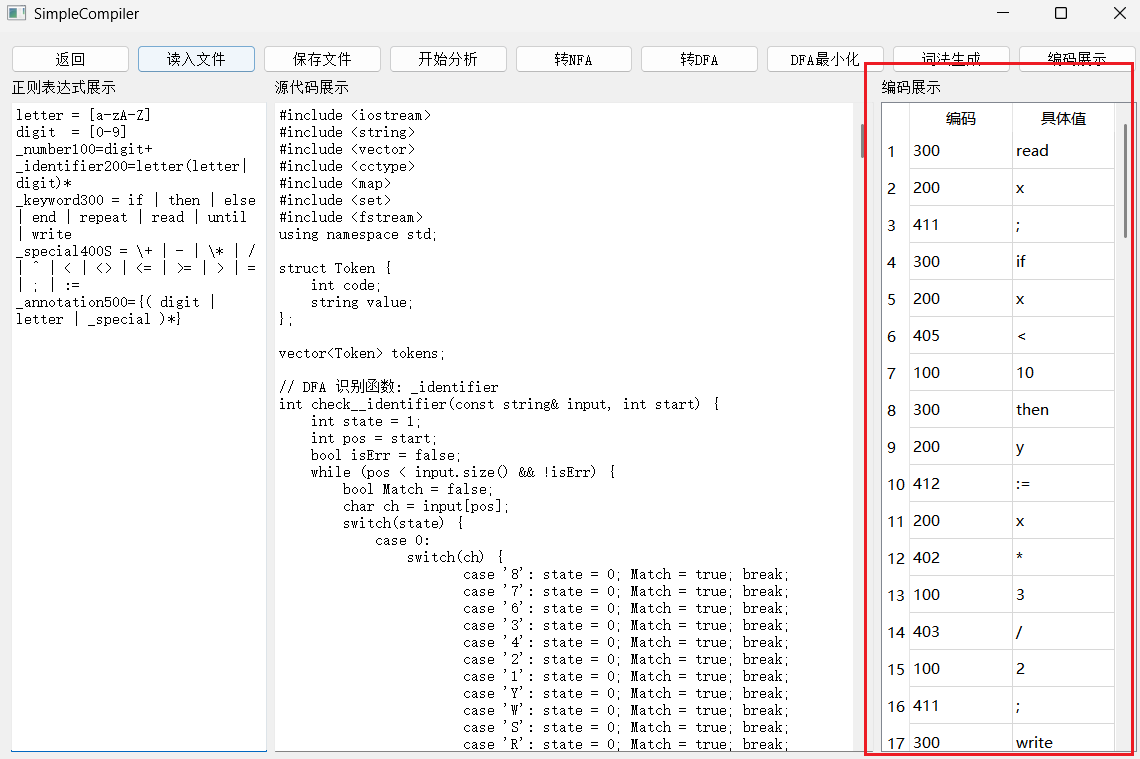
**6、编译cpp文件并运行**



上图可知：编译成功

并成功生成lex文件：

**7、查看lex文件**



具体lex如下：

|  |
| --- |
| 300 read 200 x 411 ; 300 if 200 x 405 < 100 10 300 then 200 y 412 := 200 x 402 \* 100 3 403 / 100 2 411 ; 300 write 200 y 300 end 300 else 300 repeat 200 x 412 := 200 x 400 + 100 1 401 - 100 2 300 until 200 x 410 = 100 0 300 end 411 ; 300 if 200 x 407 <= 100 99 300 then 200 x 412 := 100 1 300 end 300 else 200 x 412 := 100 1 300 end 411 ; 300 if 200 x 406 <> 100 100 300 then 200 x 412 := 100 1 300 end 411 ; 300 if 200 x 409 > 100 99 300 then 200 x 412 := 100 1 300 end 411 ; 300 if 200 x 408 >= 100 100 300 then 200 x 412 := 200 x 404 ^ 100 2 300 end 300 else 200 x 412 := 100 0 300 end |

**任务一测试完全通过**

**任务二**

**1、准备工作**

**tiny的文法**

|  |
| --- |
| program -> stmt-sequence  stmt-sequence -> stmt-sequence ; statement | statement  statement -> if-stmt | repeat-stmt | assign-stmt | read-stmt | write-stmt  if-stmt -> if exp then stmt-sequence end | if exp then stmt-sequence end else stmt-sequence end  repeat-stmt -> repeat stmt-sequence until exp  assign-stmt -> identifier := exp  read-stmt -> read identifier  write-stmt -> write exp  exp -> simple-exp comparison-op simple-exp | simple-exp  comparison-op -> < | > | = | <= | <> | >=  simple-exp -> simple-exp addop term | term  addop -> + | -  term -> term mulop factor | factor  mulop -> \* | / | % | ^  factor -> ( exp ) | number | identifier |

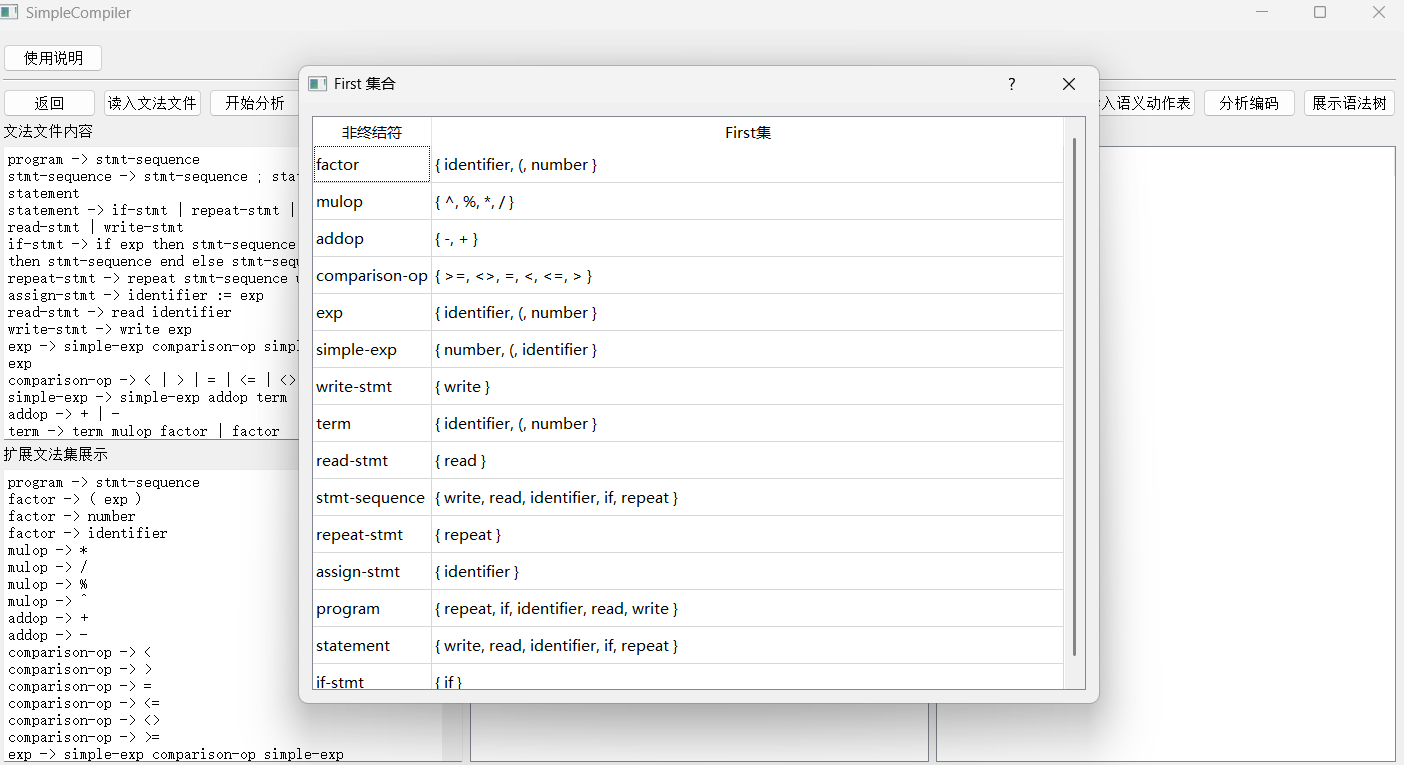
**tiny的sample.lex编码文本（见上任务一最底部的编码）**

**tiny的语义动作表**

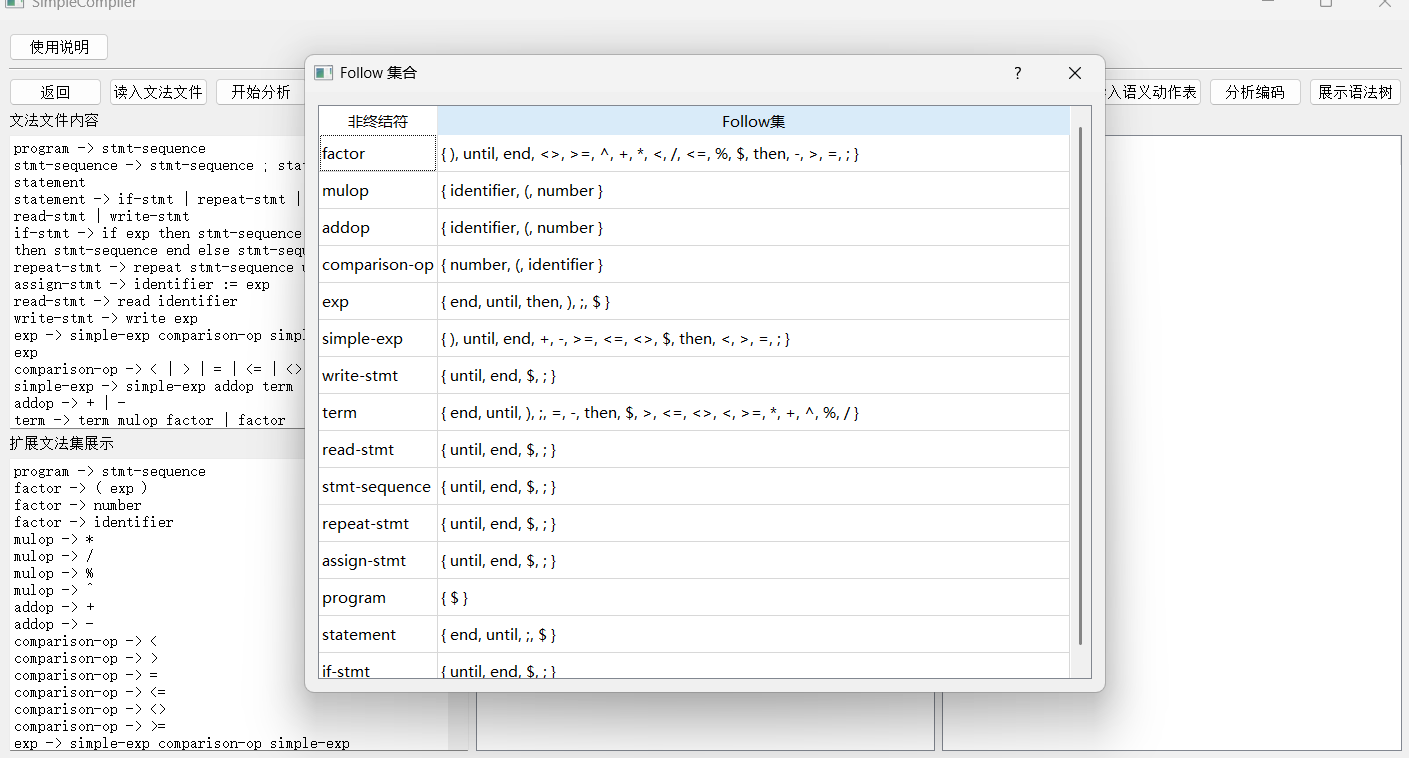
|  |
| --- |
| program -> stmt-sequence  1  stmt-sequence -> stmt-sequence ; statement  1 0 3  stmt-sequence -> statement  1  statement -> if-stmt  1  statement -> repeat-stmt  1  statement -> assign-stmt  1  statement -> read-stmt  1  statement -> write-stmt  1  if-stmt -> if exp then stmt-sequence end  1 2 0 2 0  if-stmt -> if exp then stmt-sequence end else stmt-sequence end  1 2 0 2 0 0 2 0  repeat-stmt -> repeat stmt-sequence until exp  1 2 0 2  assign-stmt -> identifier := exp  2 1 2  read-stmt -> read identifier  1 2  write-stmt -> write exp  1 2  exp -> simple-exp comparison-op simple-exp  2 1 2  exp -> simple-exp  1  comparison-op -> <  1  comparison-op -> >  1  comparison-op -> =  1  comparison-op -> <=  1  comparison-op -> <>  1  comparison-op -> >=  1  simple-exp -> simple-exp addop term  2 1 2  simple-exp -> term  1  addop -> +  1  addop -> -  1  term -> term mulop factor  2 1 2  term -> factor  1  mulop -> \*  1  mulop -> /  1  mulop -> %  1  mulop -> ^  1  factor -> ( exp )  0 1 0  factor -> number  1  factor -> identifier  1 |

**2、开始测试**

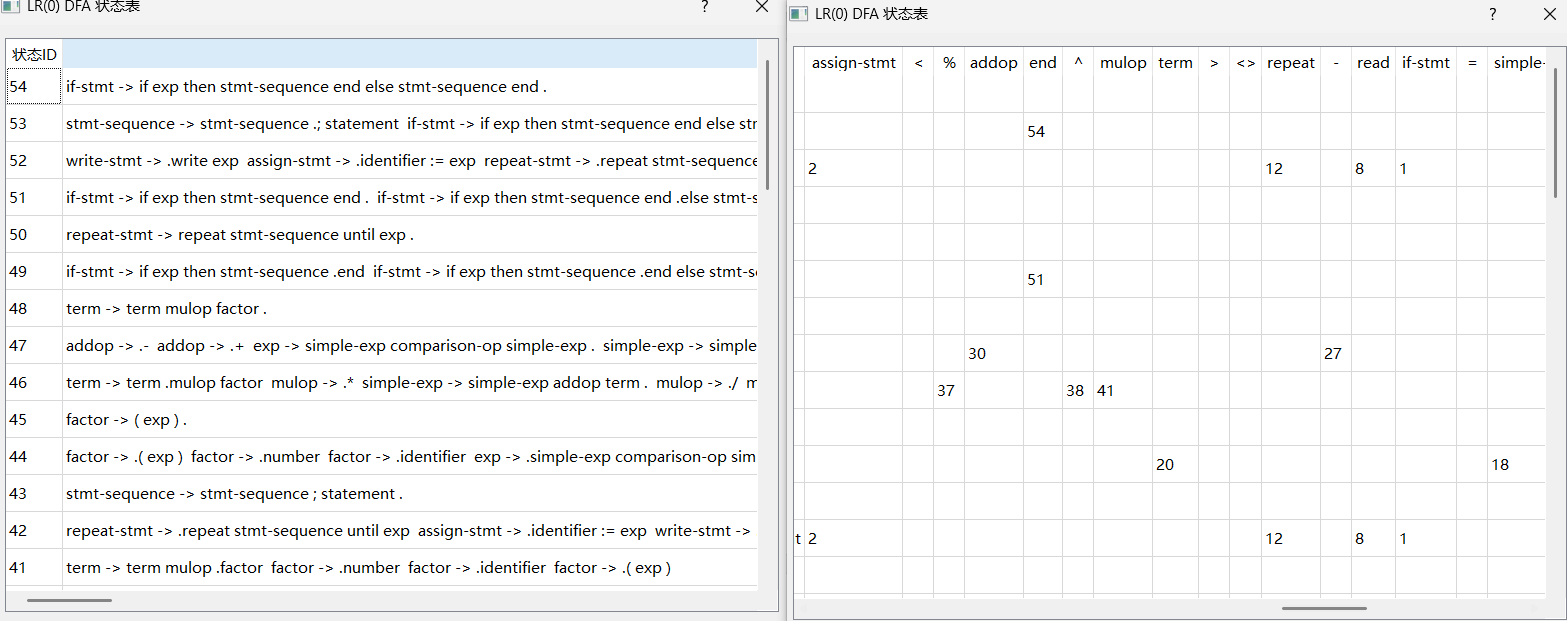
1. **First集合：**



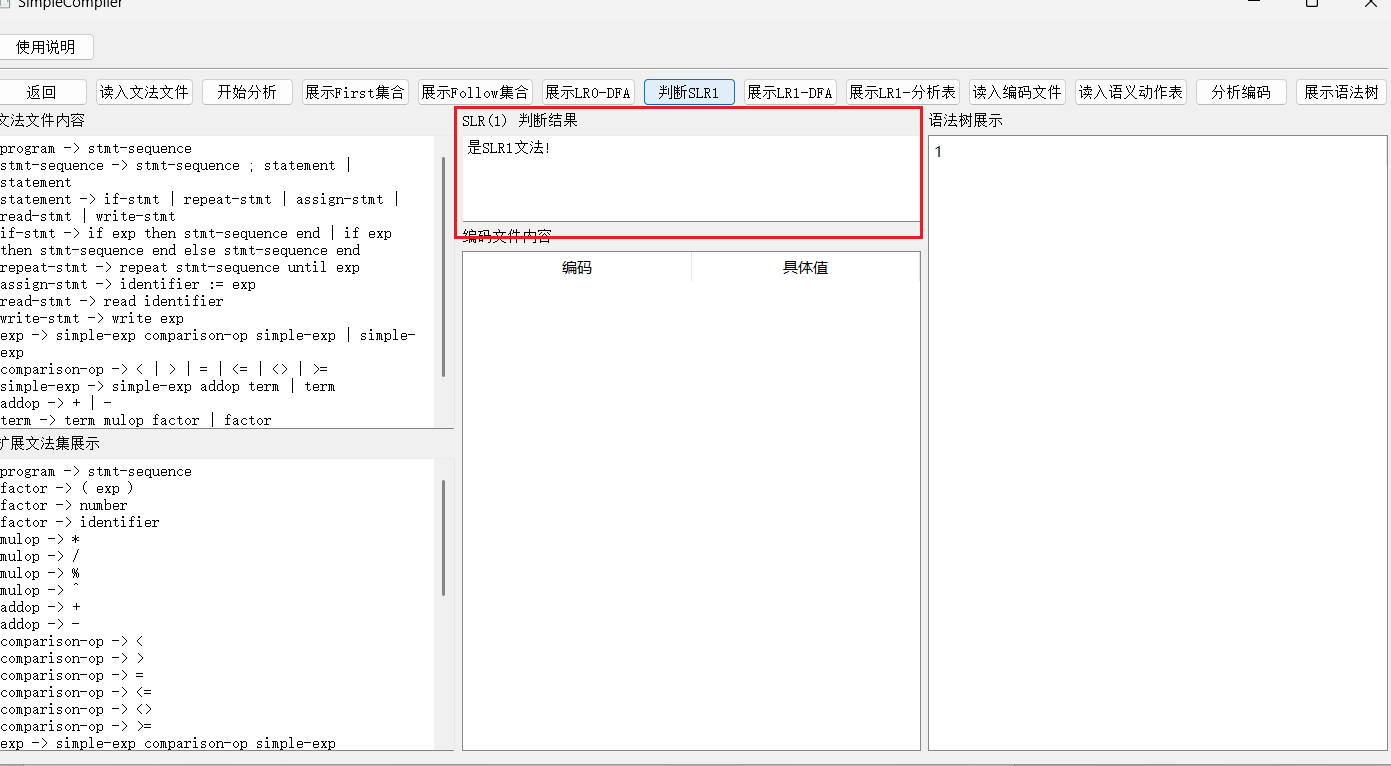
1. **Follow集合：**



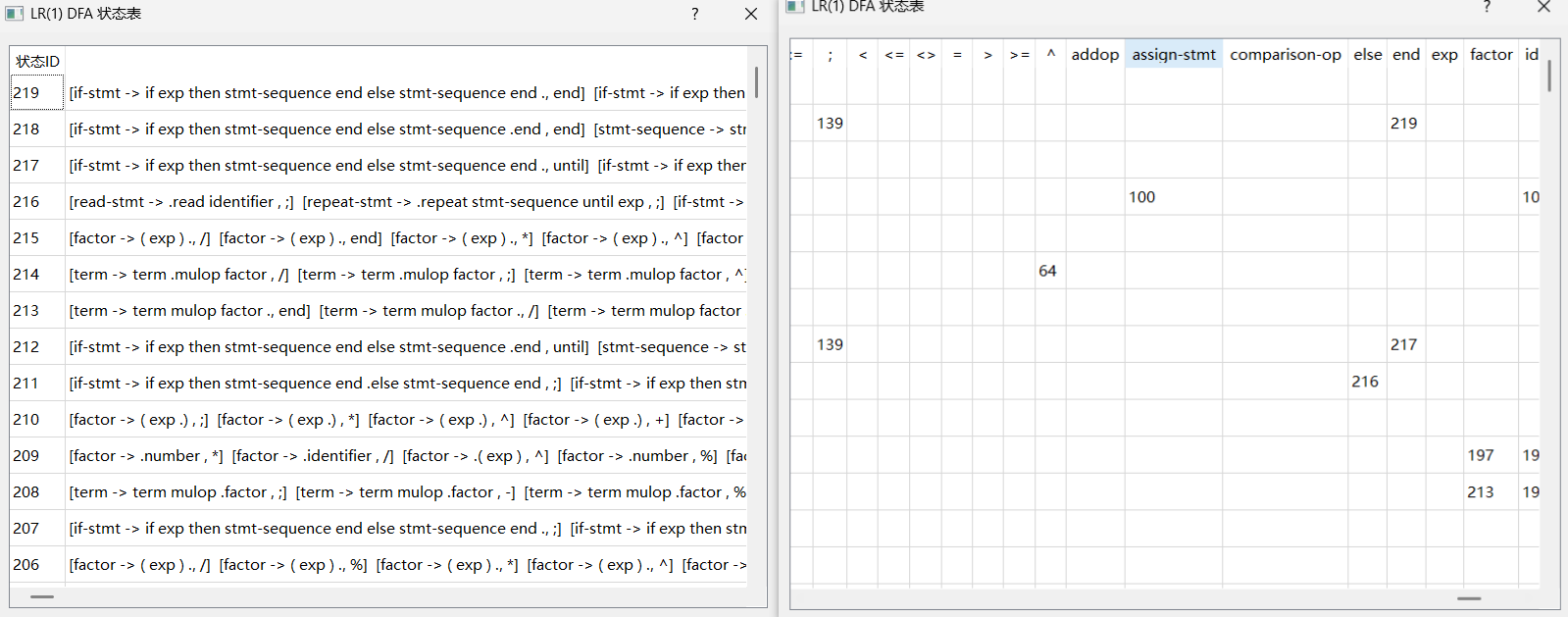
1. **LR0-DFA：**



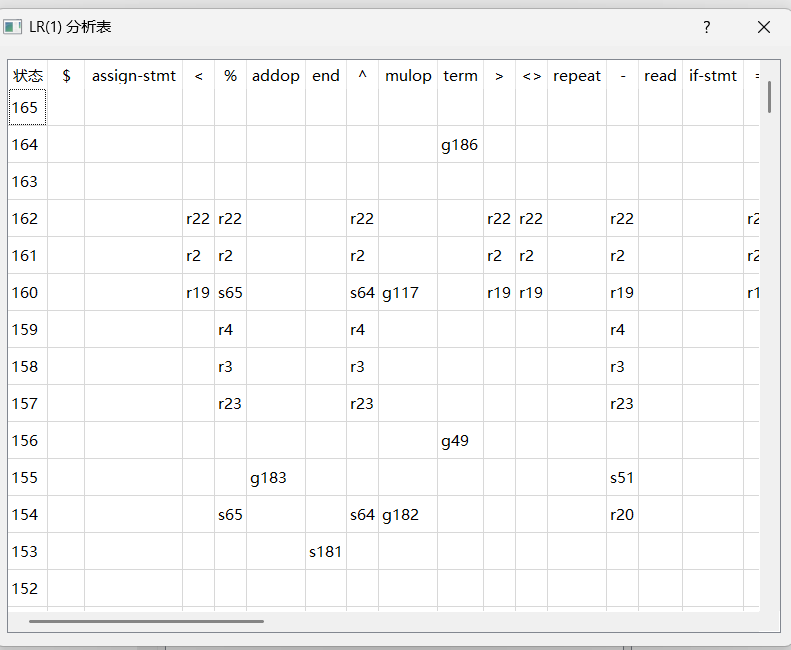
1. **判断是否为SLR1：**



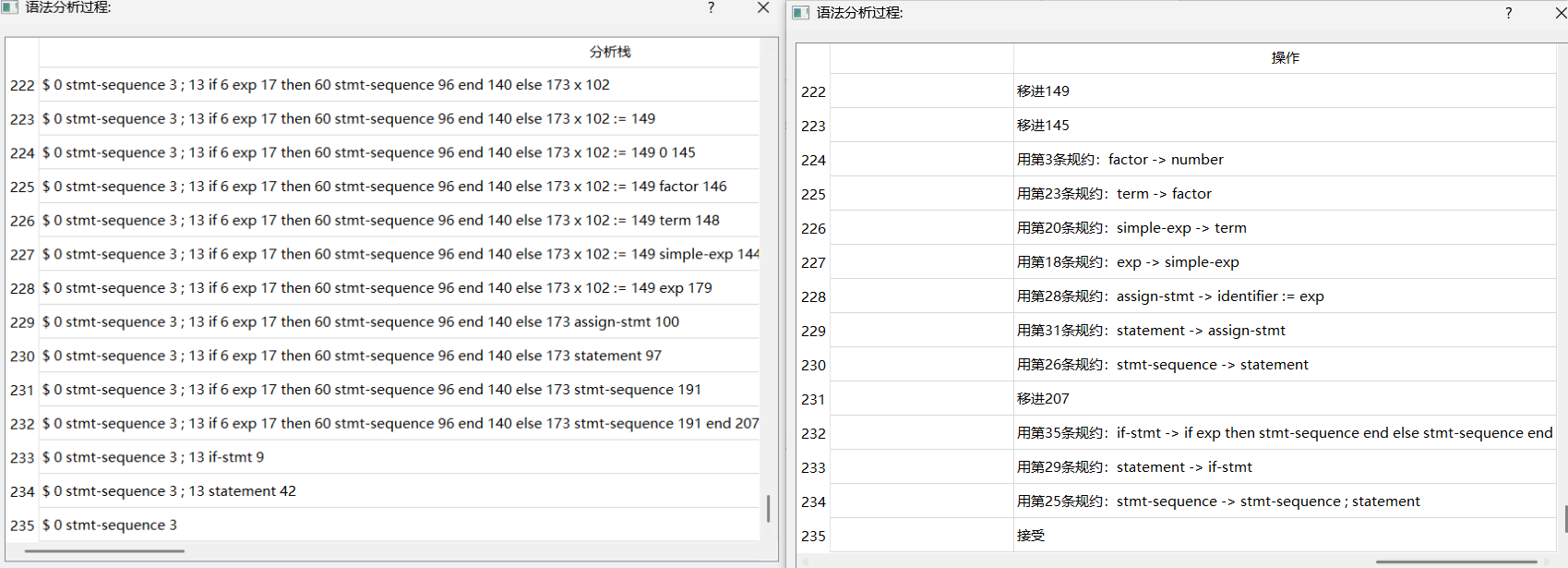
1. **LR1-DFA：**



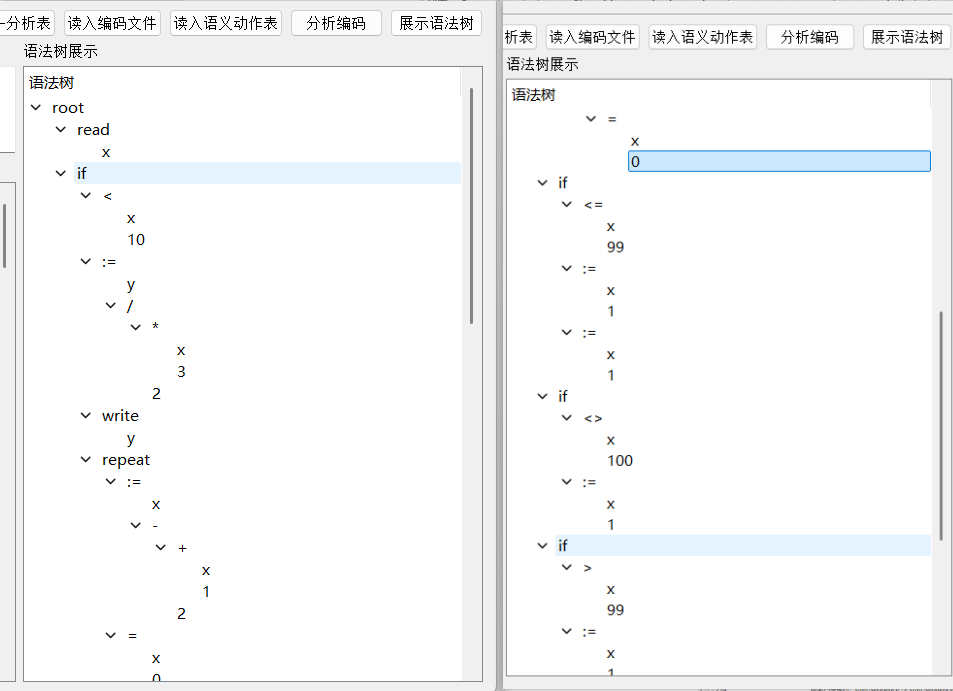
1. **LR1-分析表：**



1. **分析编码过程展示**



1. **语法树展示**



**任务二测试完全通过**

**总结：根据我们最后展示的语法分析过程的结果，以及语法树的展示结果来看，我们本次任务一和任务二的测试是完全通过的。**