项目一测试报告

一、测试项目名称

项目一：词法分析和单词编码测试

二、测试内容

对 TINY 和 mini-c 语言的正则表达式进行词法分析，生成词法分析程序，对于两个语言的源程序，分析词法元素并输出编码。

三、测试用例

1）mini-c 正则.txt

2）Sample.minic

3）TINY 正则.txt

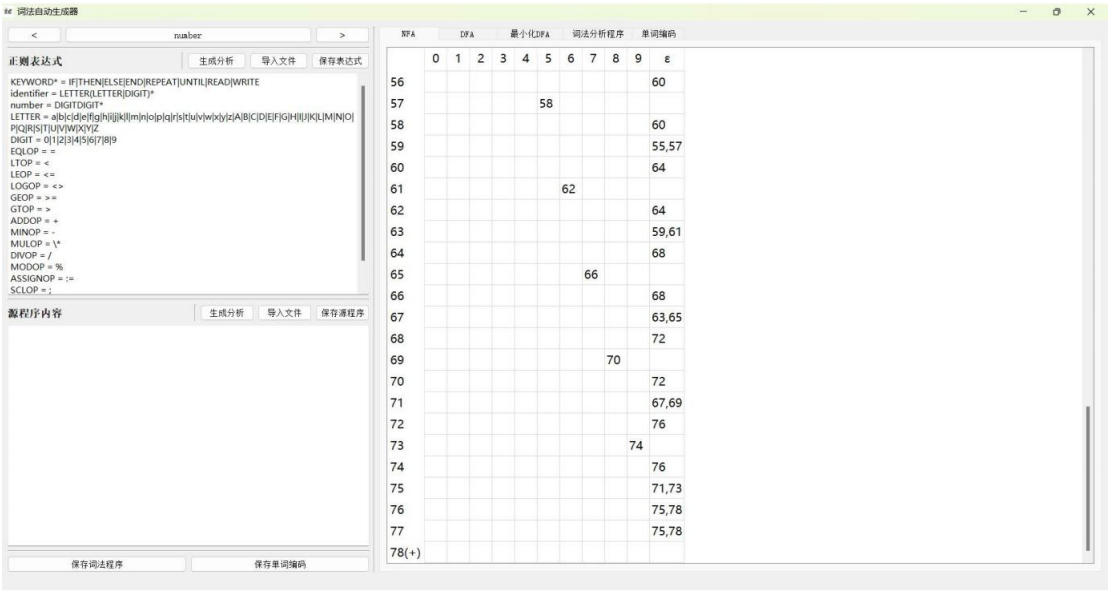
4）Sample.tny

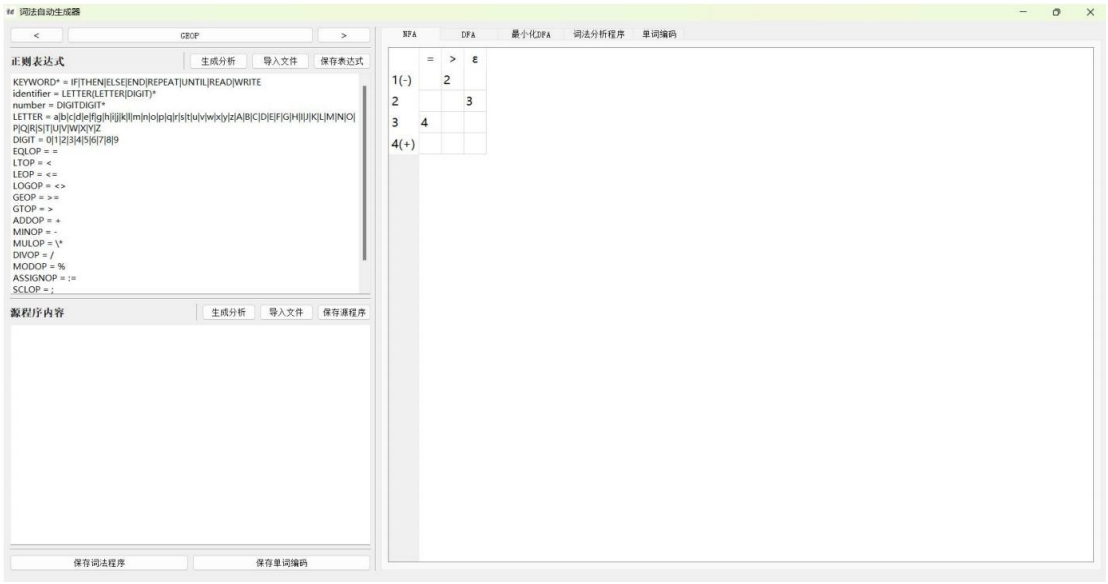
四、测试结果

由于有几条正则表达式状态过多，部分板块无法展示完全，因此只展示部分，若要查看所有信息，可通过可执行程序进行测试。

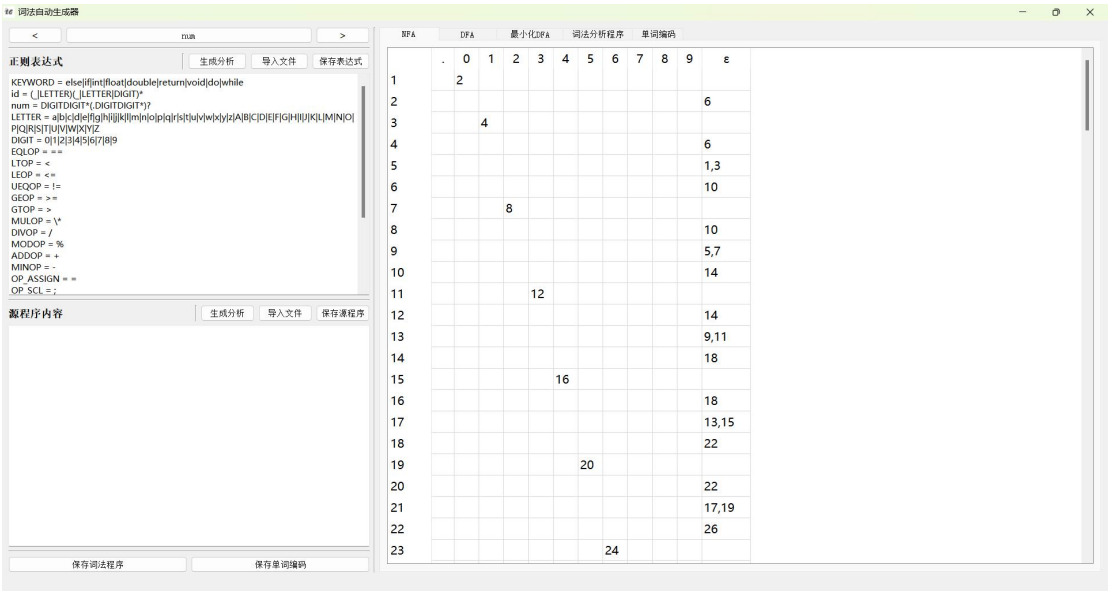
4.1 NFA

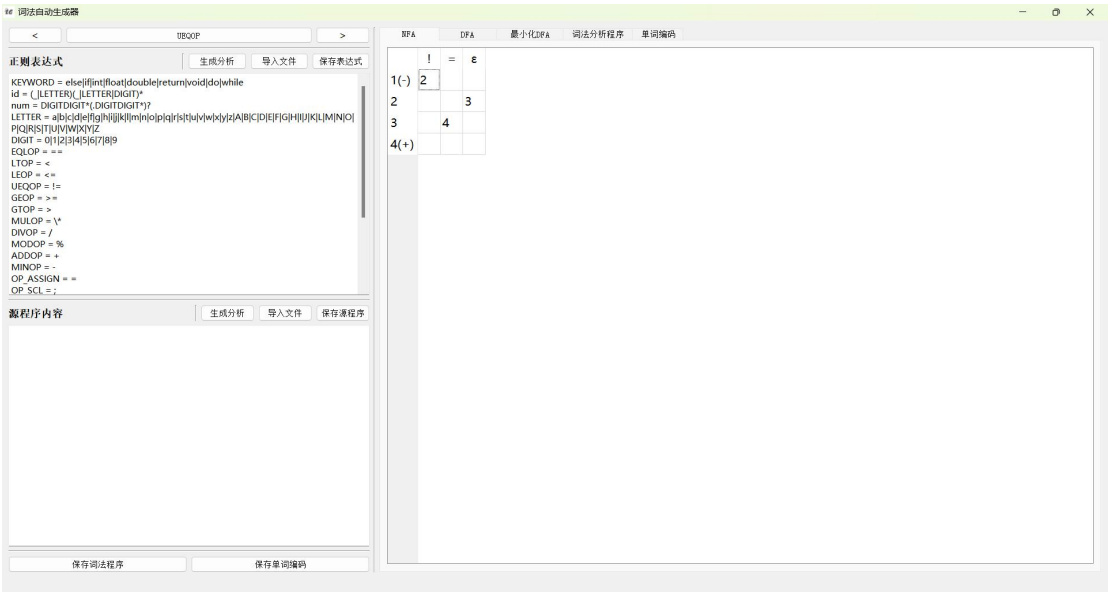
TINY


词法自动生成器
number
正则表达式
生成分析
导入文件
保存表达式
KEYWORD* = IF|THEN|ELSE|END|REPEAT|UNTIL|READ|WRITE
identifier = LETTER(LETTER|DIGIT)*
number = DIGITDIGIT*
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = =
LTOP = <
LEOP = <=
LOGOP = <>
GEOP = >=
GTOP = >
ADDOP = +
MINOP = -
MULOP = *
DIVOP = /
MODOP = %
ASSIGNOP = :=
SCLOP = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
0 1 2 3 4 5 6 7 8 9 ε
56 60
57 58 60
58 60
59 55,57
60 64
61 62 64
62 64
63 59,61
64 68
65 66 68
66 68
67 63,65
68 72
69 70 72
70 72
71 67,69
72 76
73 74 76
74 76
75 71,73
76 75,78
77 75,78
78(+)


词法自动生成器
GEOP
正则表达式
生成分析
导入文件
保存表达式
KEYWORD* = IF|THEN|ELSE|END|REPEAT|UNTIL|READ|WRITE
identifier = LETTER(LETTER|DIGIT)*
number = DIGITDIGIT*
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = =
LTOP = <
LEOP = <=
LOGOP = <>
GEOP = >=
GTOP = >
ADDOP = +
MINOP = -
MULOP = *
DIVOP = /
MODOP = %
ASSIGNOP = :=
SCLOP = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
= > ε
1(-) 2
2 3
3 4
4(+)

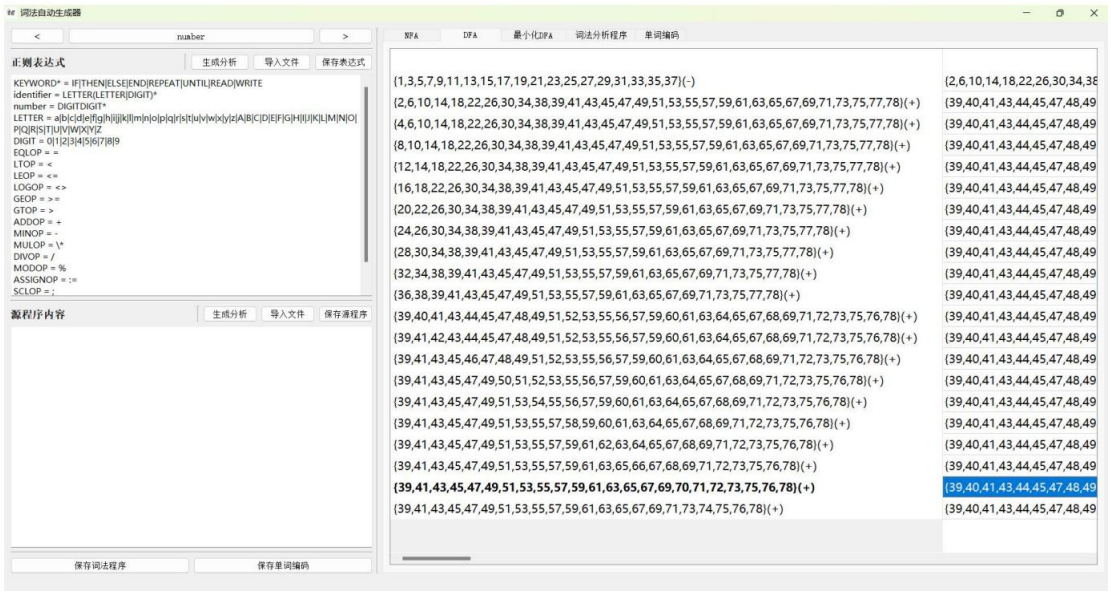
mini-c

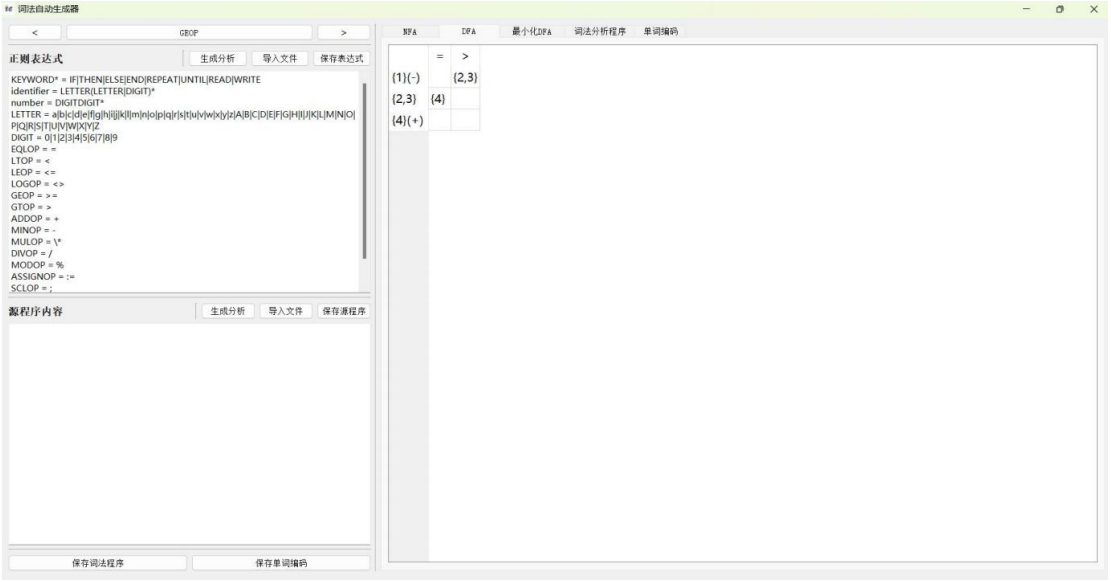

词法自动生成器
num
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
. 0 1 2 3 4 5 6 7 8 9 ε
1 2
2 6
3 4
4 6
5 1,3
6 10
7 8
8 10
9 5,7
10 14
11 12
12 14
13 9,11
14 18
15 16
16 18
17 13,15
18 22
19 20
20 22
21 17,19
22 26
23 24


词法自动生成器
UEQOP
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
! = ε
1(-) 2
2 3
3 4
4(+)

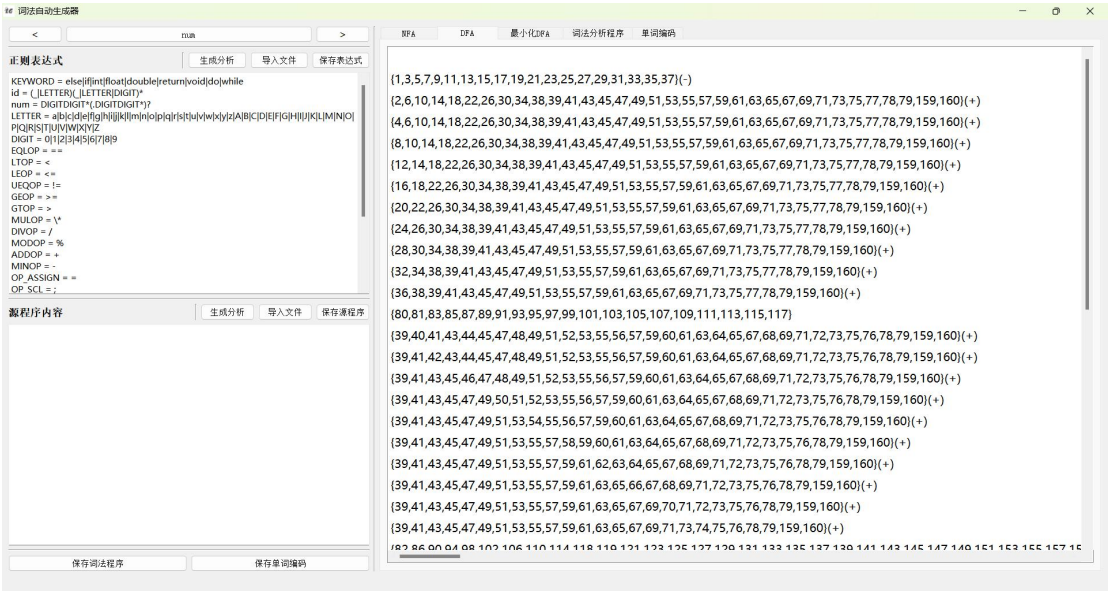
4.2 DFA

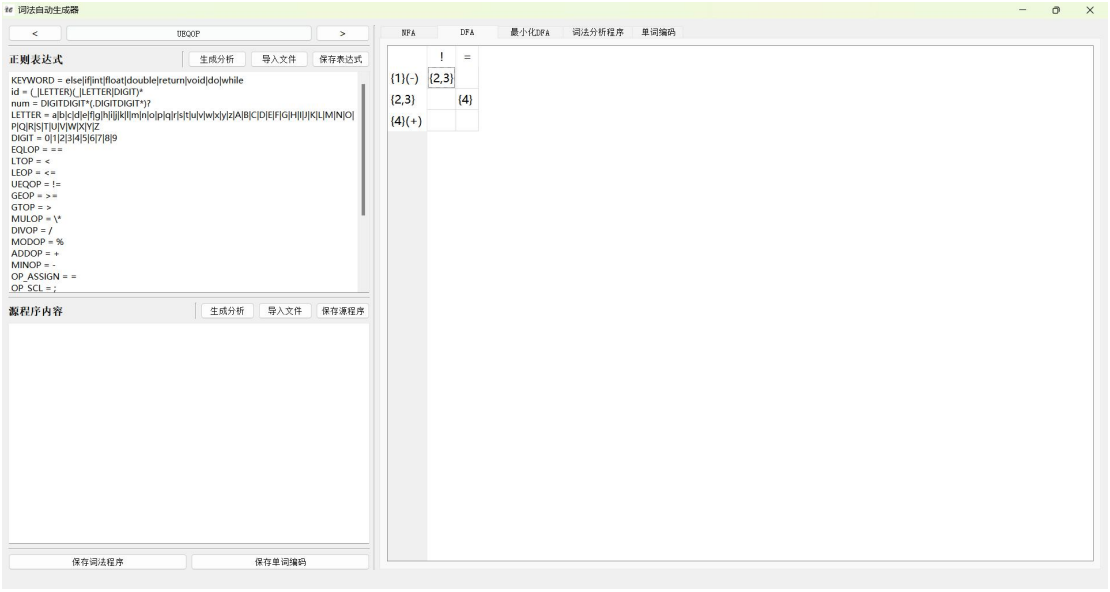
TINY


词法自动生成器
number
正则表达式
生成分析
导入文件
保存表达式
KEYWORD* = IF|THEN|ELSE|END|REPEAT|UNTIL|READ|WRITE
identifier = LETTER(LETTER|DIGIT)*
number = DIGITDIGIT*
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = =
LTOP = <
LEOP = <=
LOGOP = <>
GEOP = >=
GTOP = >
ADDOP = +
MINOP = -
MULOP = *
DIVOP = /
MODOP = %
ASSIGNOP = :=
SCLOP = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
{1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37}(-)
{2,6,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{4,6,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{8,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{12,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{16,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{20,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{24,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{28,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{32,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{36,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78}(+)
{39,40,41,43,44,45,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,42,43,44,45,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,46,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,50,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,54,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,55,57,58,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,62,63,64,65,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,66,67,68,69,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,70,71,72,73,75,76,78}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,74,75,76,78}(+)
{2,6,10,14,18,22,26,30,34,38
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49
{39,40,41,43,44,45,47,48,49


词法自动生成器
GEOP
正则表达式
生成分析
导入文件
保存表达式
KEYWORD* = IF|THEN|ELSE|END|REPEAT|UNTIL|READ|WRITE
identifier = LETTER(LETTER|DIGIT)*
number = DIGITDIGIT*
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = =
LTOP = <
LEOP = <=
LOGOP = <>
GEOP = >=
GTOP = >
ADDOP = +
MINOP = -
MULOP = *
DIVOP = /
MODOP = %
ASSIGNOP = :=
SCLOP = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
= >
{1}(-)
{2,3}
{2,3} {4}
{4}(+)

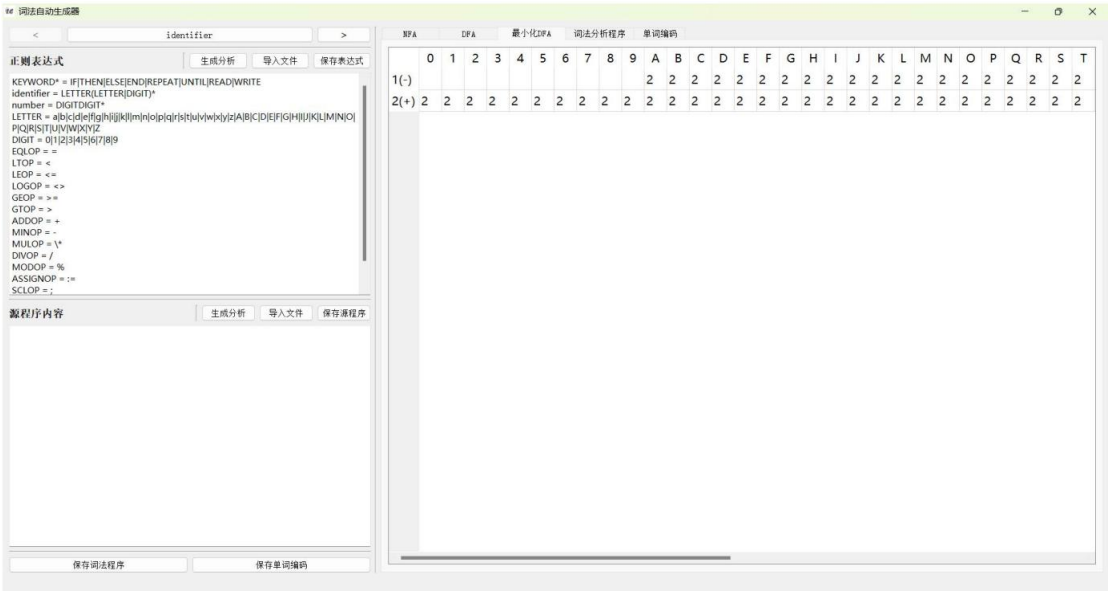
mini-c

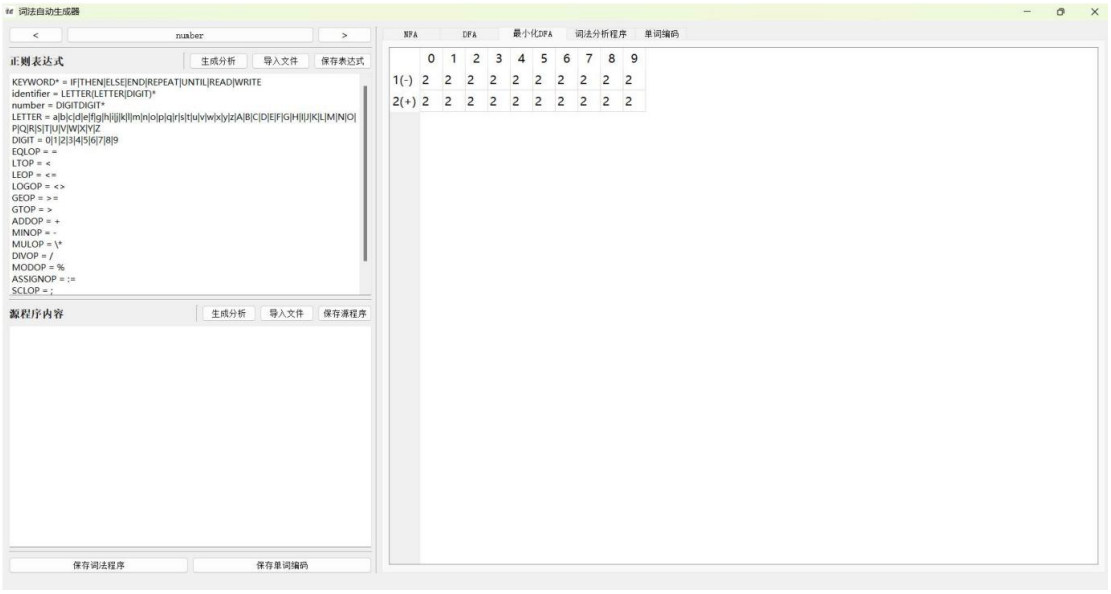

词法自动生成器
num
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
{1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37}(-)
{2,6,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{4,6,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{8,10,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{12,14,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{16,18,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{20,22,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{24,26,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{28,30,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{32,34,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{36,38,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,78,79,159,160}(+)
{80,81,83,85,87,89,91,93,95,97,99,101,103,105,107,109,111,113,115,117}
{39,40,41,43,44,45,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,42,43,44,45,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,46,47,48,49,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,50,51,52,53,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,54,55,56,57,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,55,57,58,59,60,61,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,62,63,64,65,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,66,67,68,69,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,70,71,72,73,75,76,78,79,159,160}(+)
{39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,74,75,76,78,79,159,160}(+)


词法自动生成器
UEQOP
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
!
=
{1}(-)
{2,3}
{2,3}
{4}
{4}(+)

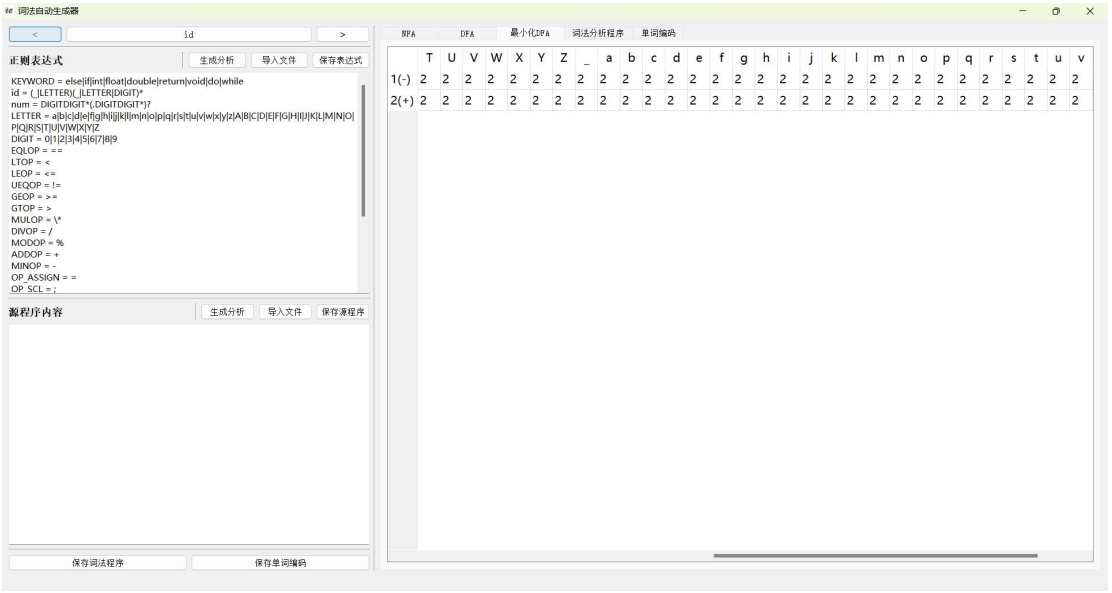
4.3 最小化 DFA

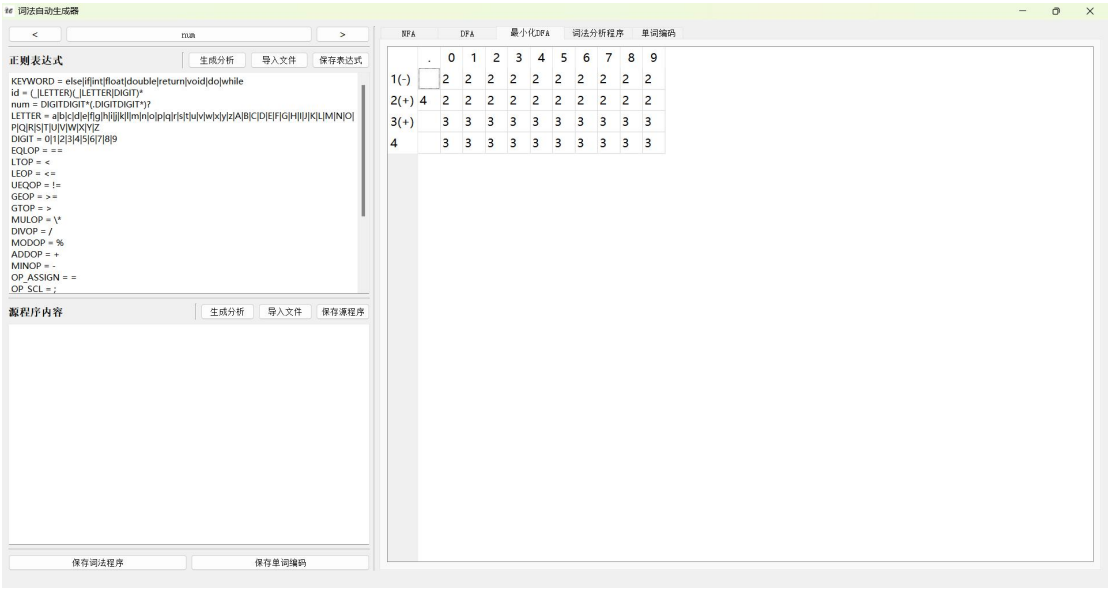
TINY





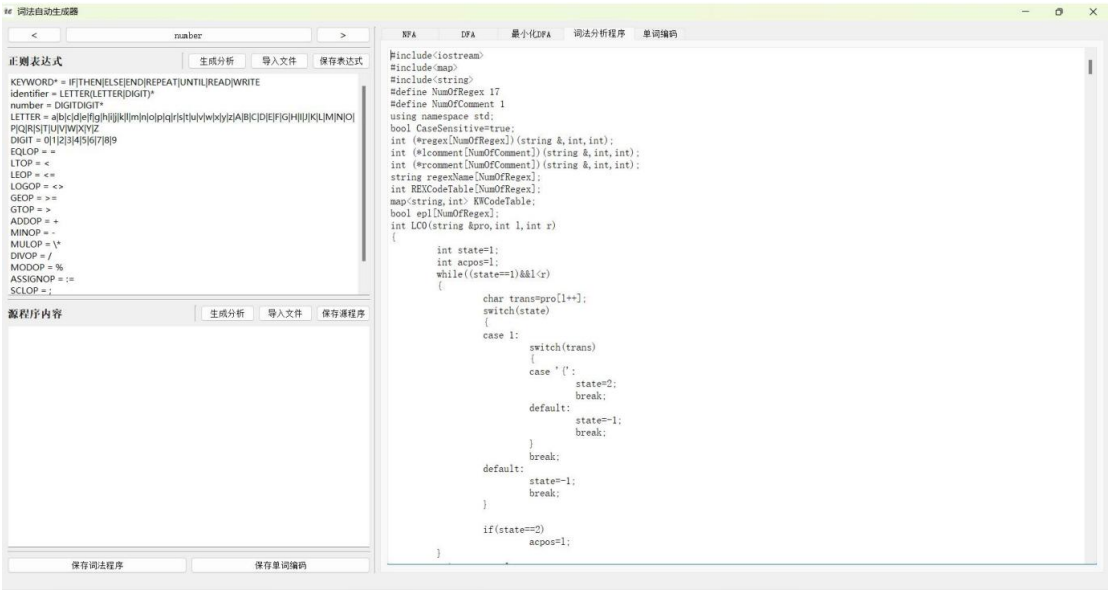
mini-c

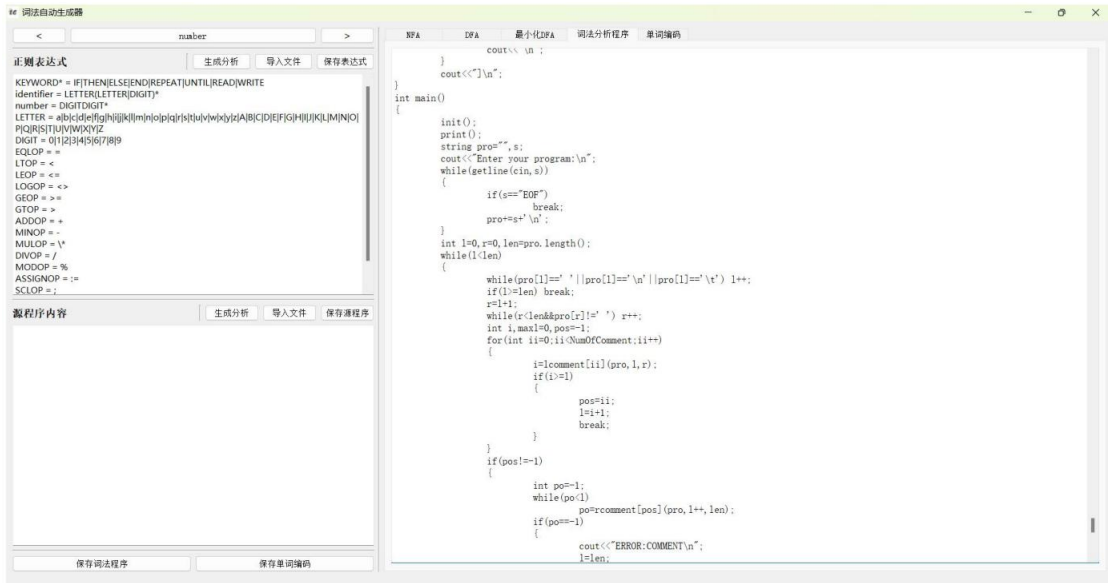

词法自动生成器
id
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEQP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
T U V W X Y Z _ a b c d e f g h i j k l m n o p q r s t u v
1(-) 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2(+) 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2


词法自动生成器
num
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEQP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
. 0 1 2 3 4 5 6 7 8 9
1(-) 2 2 2 2 2 2 2 2 2 2
2(+) 4 2 2 2 2 2 2 2 2 2
3(+) 3 3 3 3 3 3 3 3 3
4 3 3 3 3 3 3 3 3 3

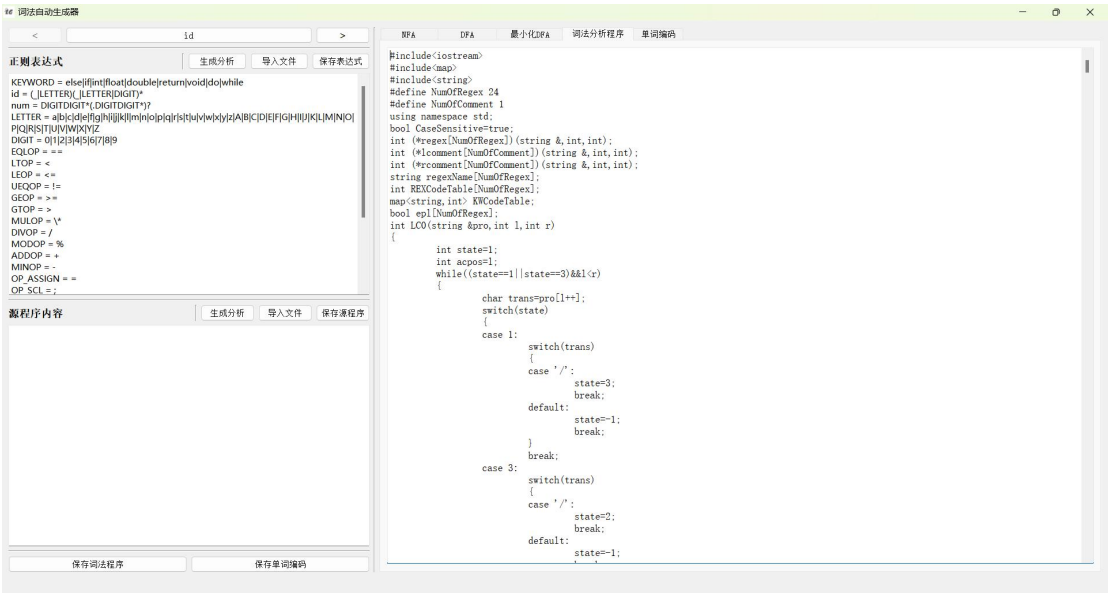
4.4 词法分析程序（具体见文件“TINY 词法分析程序.txt”和“mini-c 词法分析程序.txt”）

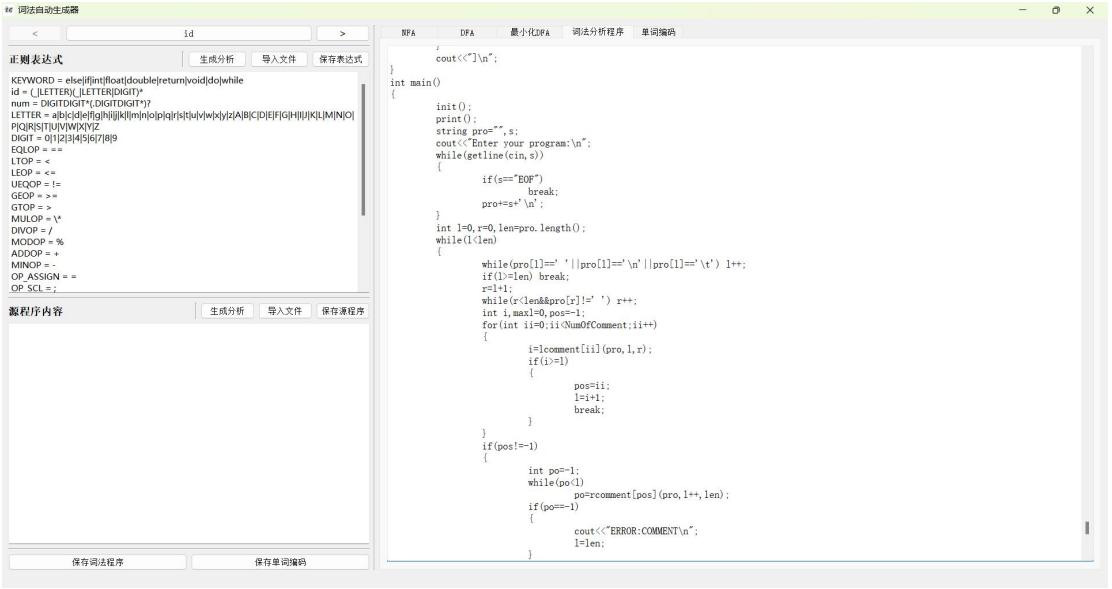
TINY





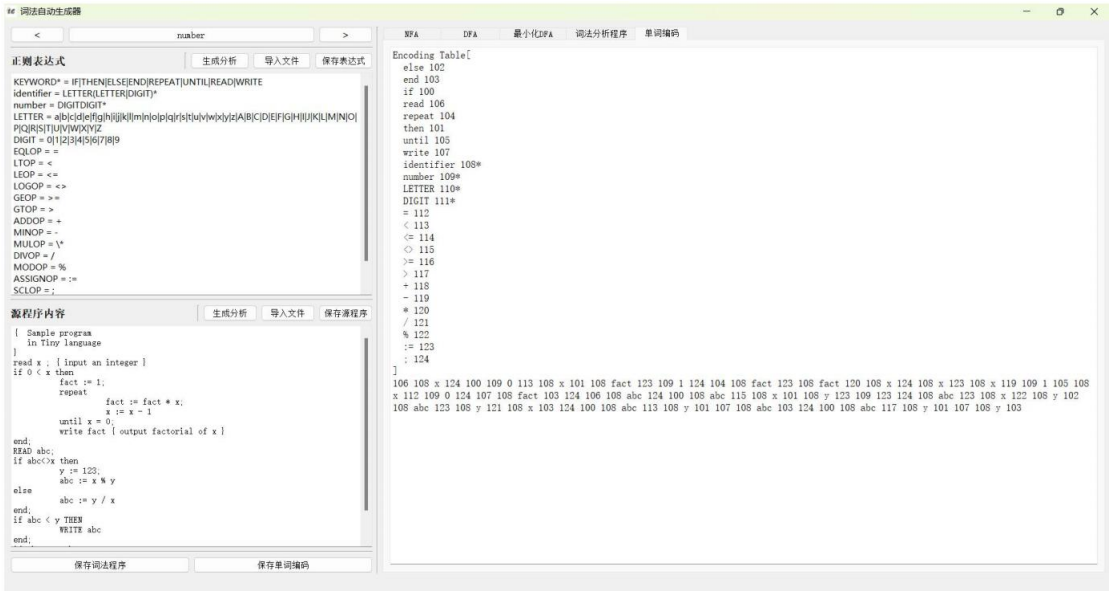
mini-c


词法自动生成器
id
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
#include<iostream>
#include<map>
#include<string>
#define NumOfRegex 24
#define NumOfComment 1
using namespace std;
bool CaseSensitive=true;
int (*regex[NumOfRegex])(string &, int, int);
int (*lcomment[NumOfComment])(string &, int, int);
int (*rcomment[NumOfComment])(string &, int, int);
string regexName[NumOfRegex];
int REXCodeTable[NumOfRegex];
map<string, int> KWCodeTable;
bool epl[NumOfRegex];
int LCO(string &pro, int l, int r)
{
int state=1;
int acpos=1;
while((state==1||state==3)&&l<r)
{
char trans=pro[l++];
switch(state)
{
case 1:
switch(trans)
{
case '/':
state=3;
break;
default:
state=-1;
break;
}
break;
case 3:
switch(trans)
{
case '/':
state=2;
break;
default:
state=-1;

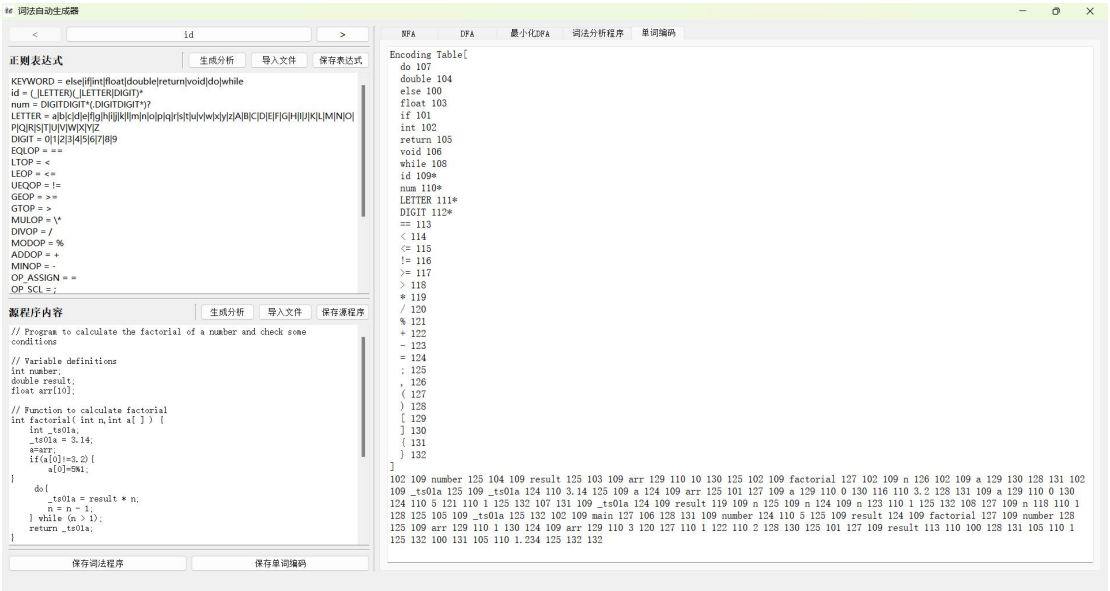

词法自动生成器
id
正则表达式
生成分析
导入文件
保存表达式
KEYWORD = else|if|int|float|double|return|void|do|while
id = (_|LETTER)(_|LETTER|DIGIT)*
num = DIGITDIGIT*(.DIGITDIGIT*)?
LETTER = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
DIGIT = 0|1|2|3|4|5|6|7|8|9
EQLOP = ==
LTOP = <
LEOP = <=
UEQOP = !=
GEOP = >=
GTOP = >
MULOP = *
DIVOP = /
MODOP = %
ADDOP = +
MINOP = -
OP_ASSIGN = =
OP_SCL = ;
源程序内容
生成分析
导入文件
保存源程序
保存词法程序
保存单词编码
NFA
DFA
最小化DFA
词法分析程序
单词编码
cout<<"]\n";
}
int main()
{
init();
print();
string pro="",s;
cout<<"Enter your program:\n";
while(getline(cin,s))
{
if(s=="EOF")
break;
pro+=s+'\n';
}
int l=0,r=0,len=pro.length();
while(l<len)
{
while(pro[l]==' '||pro[l]=='\n'||pro[l]=='\t') l++;
if(l>=len) break;
r=l+1;
while(r<len&&pro[r]!=' ') r++;
int i,maxl=0,pos=-1;
for(int ii=0;ii<NumOfComment;ii++)
{
i=lcomment[ii](pro,l,r);
if(i>=l)
{
pos=ii;
l=i+1;
break;
}
}
if(pos!=-1)
{
int po=-1;
while(po<l)
po=rcomment[pos](pro,l++,len);
if(po==-1)
{
cout<<"ERROR:COMMENT\n";
l=len;
}

4.5 单词编码

TINY

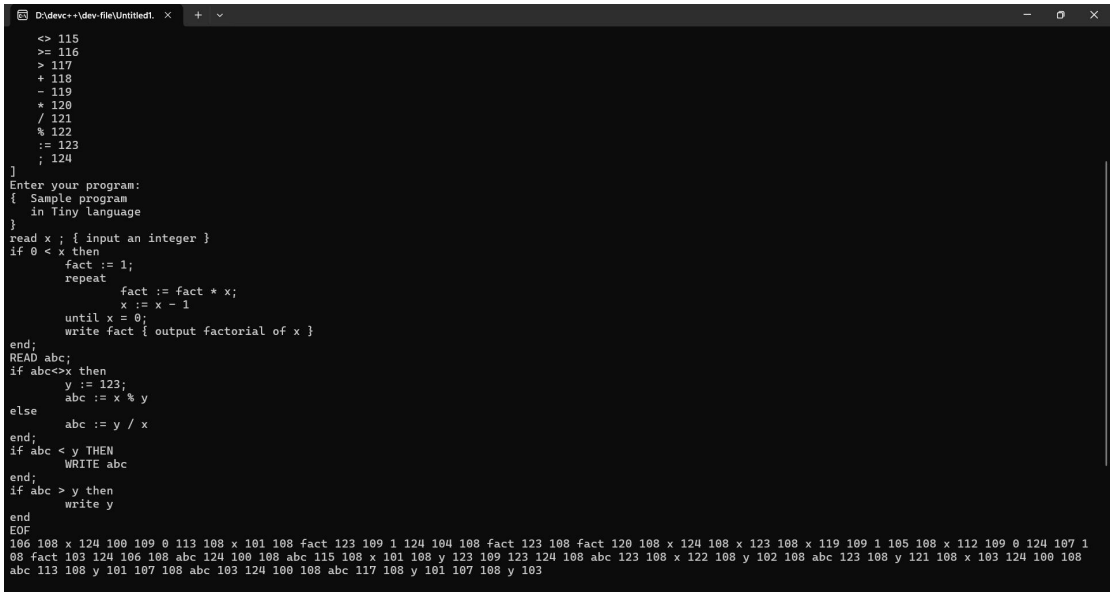


mini-c



将生成的词法分析程序在编译器中单独运行，输入 Sample.txt 和 Sample.minic，结果如下图所示：

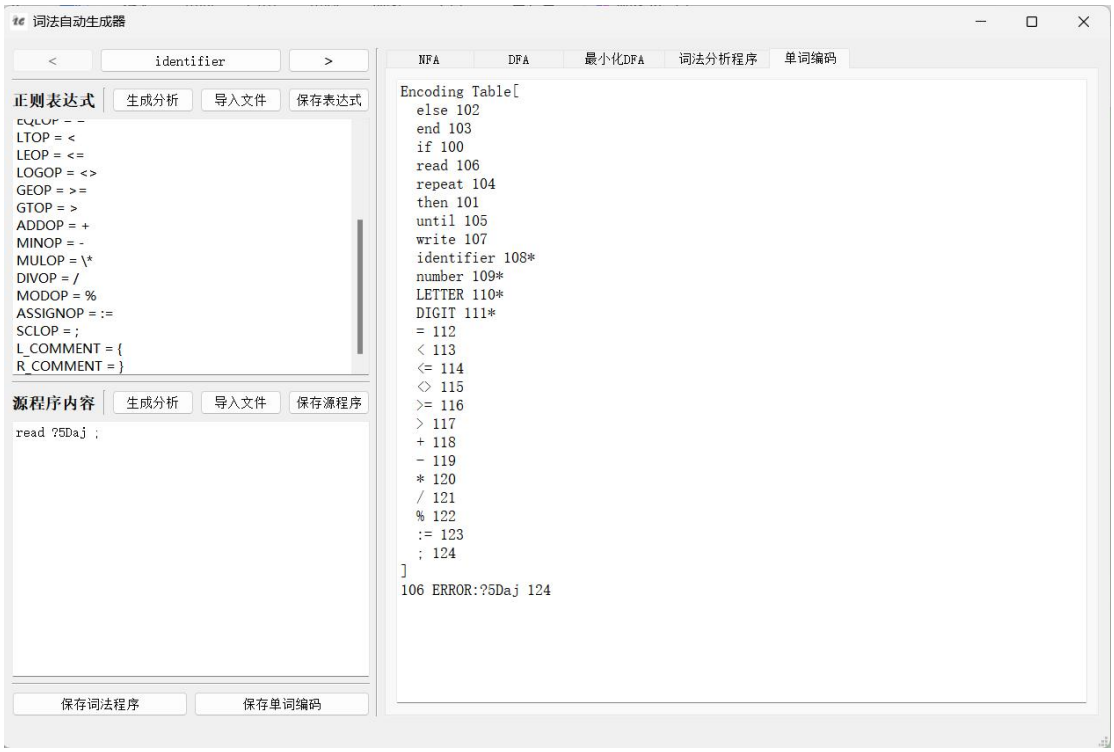
TINY


D:\devc++\dev-file\Untitled1.
<> 115
>= 116
> 117
+ 118
- 119
* 120
/ 121
% 122
:= 123
; 124
]
Enter your program:
{ Sample program
   in Tiny language
}
read x ; { input an integer }
if 0 < x then
fact := 1;
repeat
fact := fact * x;
x := x - 1
until x = 0;
write fact { output factorial of x }
end;
READ abc;
if abc<>x then
y := 123;
abc := x % y
else
abc := y / x
end;
if abc < y THEN
WRITE abc
end;
if abc > y then
write y
end
EOF
106 108 x 124 100 109 0 113 108 x 101 108 fact 123 109 1 124 104 108 fact 123 108 fact 120 108 x 124 108 x 123 108 x 119 109 1 105 108 x 112 109 0 124 107 1
08 fact 103 124 106 108 abc 124 100 108 abc 115 108 x 101 108 y 123 109 123 124 108 abc 123 108 x 122 108 y 102 108 abc 123 108 y 121 108 x 103 124 100 108
abc 113 108 y 101 107 108 abc 103 124 100 108 abc 117 108 y 101 107 108 y 103

mini-c


D:\devc++\dev-file\Untitled1.
// Program to calculate the factorial of a number and check some conditions
// Variable definitions
int number;
double result;
float arr[10];
// Function to calculate factorial
int factorial(int n,int a[ ]) {
int _ts01a;
_ts01a = 3.14;
a=arr;
if(a[0]!=3.2){
a[0]=5%1;
}
do{
_ts01a = result * n;
n = n - 1;
} while (n > 1);
return _ts01a;
}
// Main function
int main(void) {
number = 5;
result = factorial(number);
arr[1]=arr[3/(1+2)];
// Print factorial result
if (result == 100) {
return 1;
} else {
return 1.234;
}
}
EOF
102 109 number 125 104 109 result 125 103 109 arr 129 110 10 130 125 102 109 factorial 127 102 109 n 126 102 109 a 129 130 128 131 102 109 _ts01a 125 109 _t
s01a 124 110 3.14 125 109 a 124 109 arr 125 101 127 109 a 129 110 0 130 116 110 3.2 128 131 109 a 129 110 0 130 124 110 5 121 110 1 125 132 107 131 109 _ts0
1a 124 109 result 119 109 n 125 109 n 124 109 n 123 110 1 125 132 108 127 109 n 118 110 1 128 125 105 109 _ts01a 125 132 102 109 main 127 106 128 131 109 nu
mber 124 110 5 125 109 result 124 109 factorial 127 109 number 128 125 109 arr 129 110 1 130 124 109 arr 129 110 3 120 127 110 1 122 110 2 128 130 125 101 1
27 109 result 113 110 100 128 131 105 110 1 125 132 100 131 105 110 1.234 125 132 132

当输入词法有误的源程序时，程序在输出单词编码时指出产生错误的 token。



五、评价

5.1 软件能力

该软件能够对多种编程语言的正则表达式进行词法分析，并生成相应的分析程序；能够根据正则表达式，将编程语言的源程序转换为编码。

5.2 缺陷与限制

该软件的缺陷之一在于对输入的规范与否的检查并不严格，若输入的正则表达式不符合规范，软件可能不会提示错误信息，而是继续执行分析过程，这可能导致软件中断运行。

5.3 建议

加强在输入时对输入内容格式的检查，并在输入有误时产生提示并中断分析。

5.4 测试结论

软件可以正确生成正则表达式的 NFA、DFA、最小 DFA 和单词编码，单词编码中能够成功实现对变量、数字的额外解释。且生成的词法分析程序分析同个程序产生的单词编码与软件输出的单词编码相同。在某些单词不符合词法时，软件能够产生错误提示并在相应位置指出。