

**院 系：计 算 机 学 院**

**实验课程：编译原理**

**实验项目：XLEX-词法自动生成器（测试报告）**

**指导老师：黄煜廉**

**开课时间：2023 ～ 2024年度第 1学期**

**专 业：计算机科学与技术**

**班 级：计科1班**

**学 生：李达良**

**学 号：20203231004**

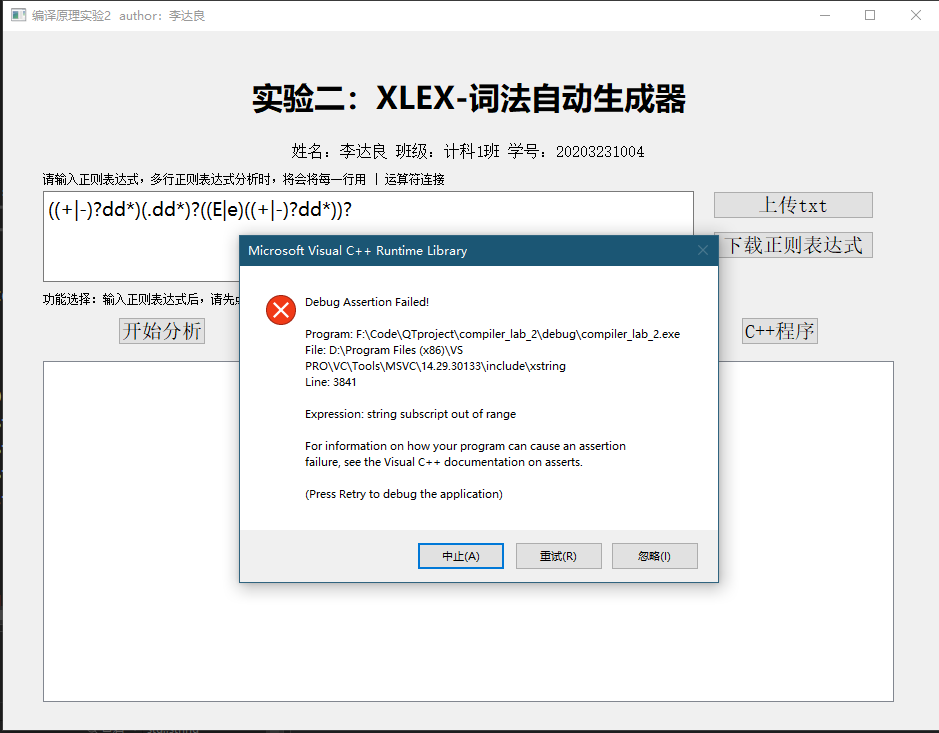
**华南师范大学教务处**

1. **测试样例**

((+|-)?dd\*)(.dd\*)?((E|e)((+|-)?dd\*))?

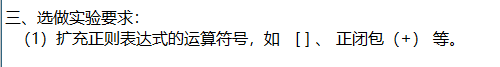
1. **测试内容**

输入正则表达式，检测是否能够生成结果

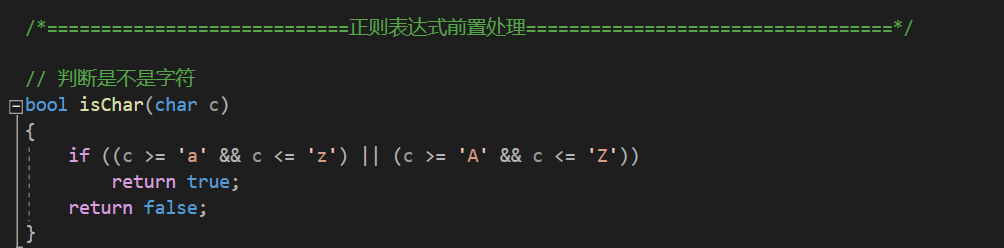


错误原因：

1. 在题目中，+号为正闭包，不应该出现在正则表达式当作字符，应该用字母代替



1. 在程序中有isChar检测，如果不是字母，不作为普通字符：



因此，需要将正则表达式中的非字母字符换成字母字符，方可进行分析：

+替换成a

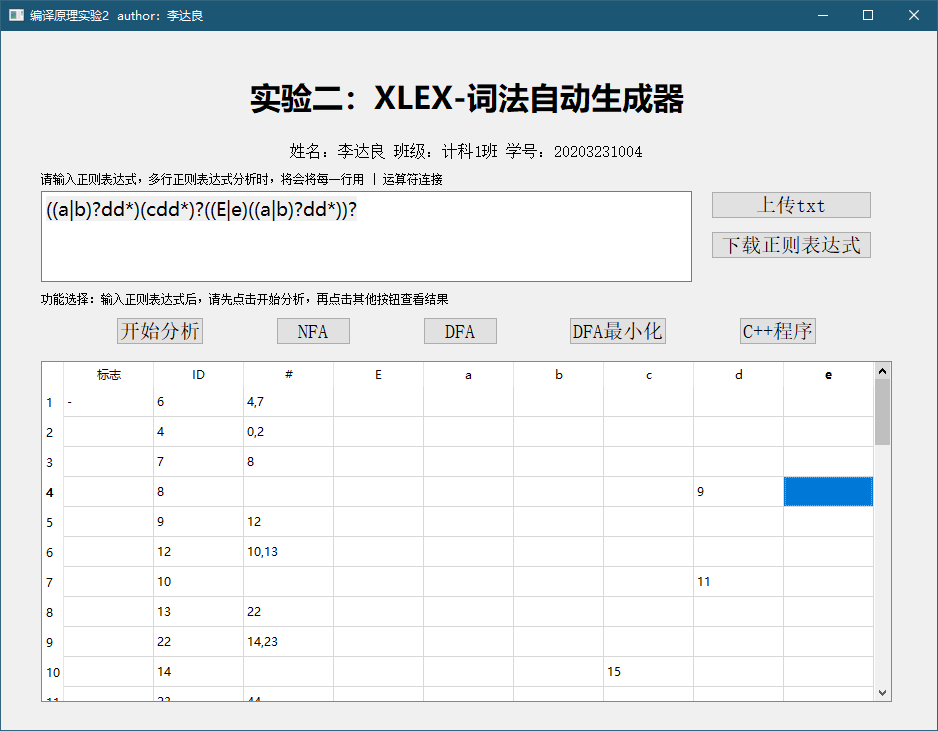
-替换成b

.替换成c

得到正则表达式为：((a|b)?dd\*)(cdd\*)?((E|e)((a|b)?dd\*))?

再次进行测试：

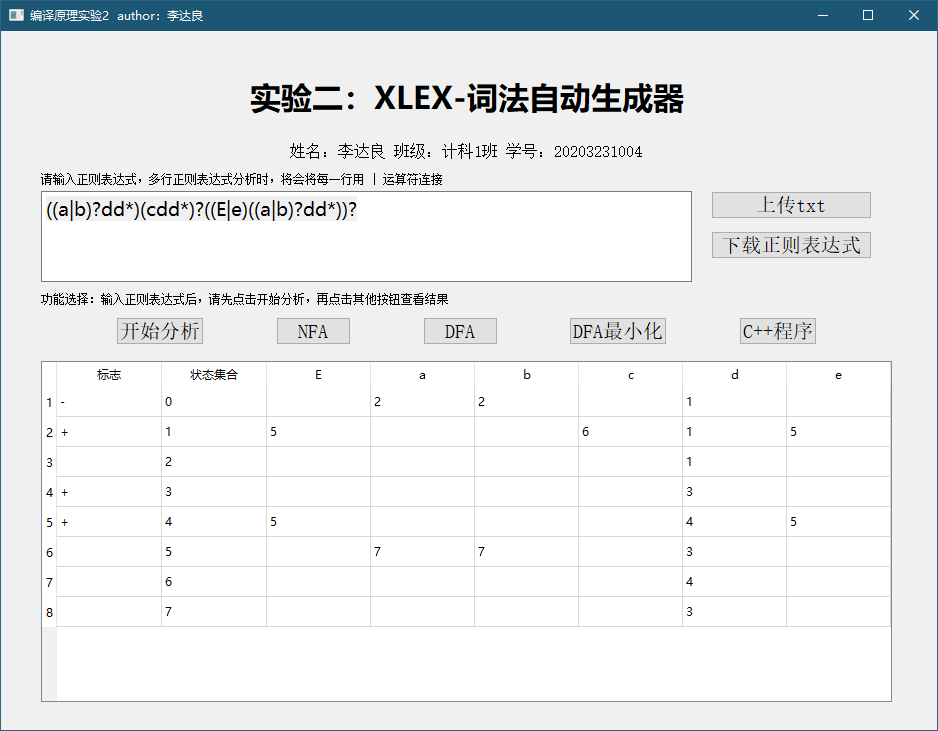
得到的NFA：



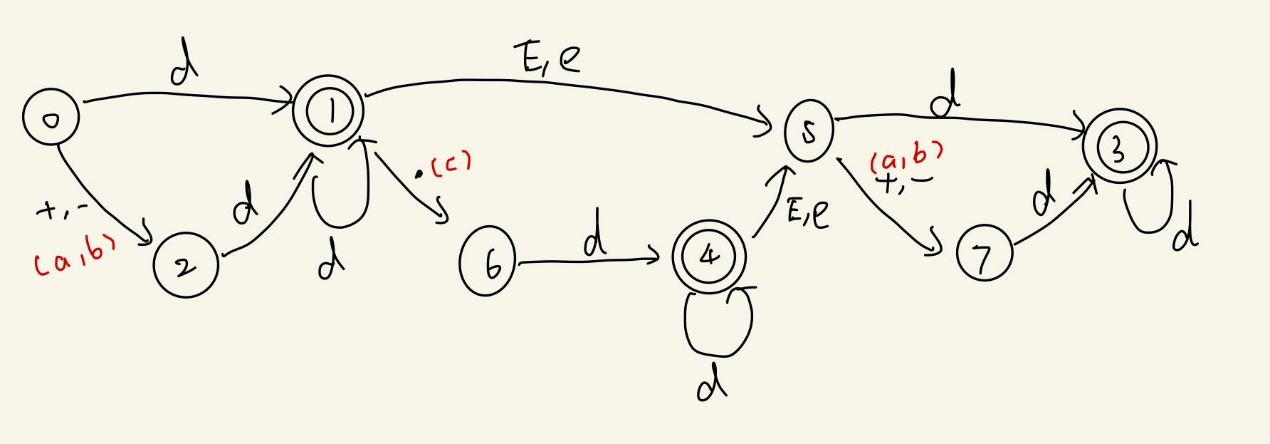
得到的DFA：



得到的DFA最小化：



通过手画作图，可以得到DFA最小化标准答案为：



对比程序生成的答案是**完全正确**的。

生成的代码（已将abc转换为对应符号）：

#include <iostream>

#include <string>

using namespace std;

int main() {

string input;

cout << "Enter input string: ";

cin >> input;

int currentState = 0;

int length = input.length();

for (int i = 0; i < length; i++) {

char c = input[i];

switch (currentState) {

case 0:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

currentState = 2;

break;

case '-':

currentState = 2;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 1;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 1:

switch (c) {

case 'E':

currentState = 5;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

currentState = 6;

break;

case 'd':

currentState = 1;

break;

case 'e':

currentState = 5;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 2:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 1;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 3:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 3;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 4:

switch (c) {

case 'E':

currentState = 5;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 4;

break;

case 'e':

currentState = 5;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 5:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

currentState = 7;

break;

case '-':

currentState = 7;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 3;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 6:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 4;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

case 7:

switch (c) {

case 'E':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '+':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '-':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case '.':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

case 'd':

currentState = 3;

break;

case 'e':

cout << "Error: Invalid input character '" << c << "'" << endl; return 1;

break;

default:

cout << "Error: Invalid input character '" << c << "'" << endl;

return 1;

}

break;

}

}

switch (currentState) {

case 1:

cout << "Accepted" << endl;

break;

case 3:

cout << "Accepted" << endl;

break;

case 4:

cout << "Accepted" << endl;

break;

default:

cout << "Not Accepted" << endl;

}

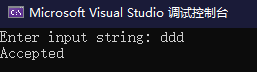
return 0;

}

对程序进行测试：

合法部分：

1. ddd



1. -ddd.ddd



1. d.ddde+dd



1. +de-d



1. dddEdd



1. d.dde-dd

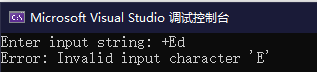


非法：

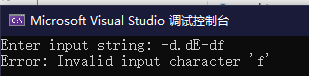
1. -.



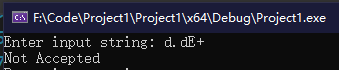
1. +Ed



1. -d.dE-df



1. d.dE+



1. **小结**

本次样例由于程序中带有正负号与正闭包检测冲突，同时程序仅支持字母作为单个字符输入，在测试时会直接报错，因此样例测试不通过**。**

但是，将正则表达式中的非字母字符换成字母字符后，再进行分析后，与手工做法得到的答案完全吻合，因此处理后的正则表达式样例是**通过**的。

最后生成的C++代码，通过10个测试样例证明生成的代码也是符合正则表达式规则的，因此本次实验二**测试通过**。