



华南师范大学

## 本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：编译原理实验

实验项目： XLEX-词法自动生成器

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：21 级

学 生： 钟文宏

学 号：20212121022

华南师范大学教务处

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分

## 一、实验内容

1. 设计一个应用软件,以实现将正则表达式-->NFA--->DFA-->DFA 最小化-->词法分析程序。
2. 正则表达式应该支持单个字符, 运算符号有: 连接、选择 (|)、闭包 (\*)、括号 ()、可选 (?)
3. 要提供一个源程序编辑界面, 让用户输入一行 (一个) 或多行 (多个) 正则表达式 (可保存、打开正则表达式文件)
4. 需要提供窗口以便用户可以查看转换得到的 NFA (用状态转换表呈现即可)
5. 需要提供窗口以便用户可以查看转换得到的 DFA (用状态转换表呈现即可)
6. 需要提供窗口以便用户可以查看转换得到的最小化 DFA (用状态转换表呈现即可)
7. 需要提供窗口以便用户可以查看转换得到的词法分析程序(该分析程序需要用 C/C++语言描述)
8. 用户界面应该是 windows 界面
9. 应该书写完善的软件文档

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分

## 二、实验目的

1. 探索 and 了解词法分析自动生成器的原理和应用。
2. 让学生掌握词法分析的基本概念和技术。
3. 实验还希望培养学生对编译原理的理解和应用能力，提高他们在软件开发和编程领域的实践能力。
4. 提高对书本知识的理解和掌握。

## 三、实验文档：

### 1. 实验项目分析

- (1) 明确项目需要 window 界面，采用 QtCreator 来创建项目，实现带有用户界面的项目。
- (2) 明确项目的实现要求。要实现正则表达式分析，自动生成识别代码。要求的正则表达式运算符有 `?`、`*`、`连结`、`|`、`(、)`。明确它们先后顺序，括号内先运算，`*`、`?` 是一元运算符，运算优先级最高，和前面一个字符运算，然后是连接、最后是 `|`。
- (3) 明确项目的实现过程，正则表达式 `-> NFA -> DFA -> DFA 最小化 -> 词法分析程序`。

### 2. 实现项目设计

- (1) 存储结构：对于 NFA、DFA 图的存储采用边集数组。结构如下

```
struct Edge
{
    int nxt;
```

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分 \_\_\_\_\_

```
char c;  
Edge(char cc='#',int n=-1)  
{  
    nxt=n;  
    c=cc;  
}  
};  
struct Vertex{  
    int state;  
    QVector<Edge> edges;  
    Vertex(int vv=0):state(vv){  
    }  
};  
struct Vertex_DFA{  
    QSet<int> v;  
    QVector<Edge> edges;  
    void set_V(QSet<int>&s)  
    {  
        v=s;  
        return;  
    }  
};
```

边的结构为下一顶点、以及该边的字符组成。

有两种类型的顶点，Vertex\_DFA 是给 DFA 顶点用的，因为状态是一个集合。Vertex 是给 NFA 和最小化 DFA 用的。它们结构里都有一个 `QVector<Edge> edges`;代表边集数组。

同时还要存储各图的起始状态、接受状态。

```
int state_first,state_end;//NFA 的起始和结束状态  
int state_MDFA_first;// DFA 的起始  
int state_DFA_first;// 最小化 DFA 的起始  
QVector<int> V_DFA_end; //DFA 结束状态  
QVector<int> V_MDFA_end; //最小化 DFA 的结束状态
```

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分 \_\_\_\_\_

(2) 重要算法。其一，能够正确进行正则表达式的运算，构建 NFA 图。其二，能够将 NFA 转换成 DFA。其三，将 DFA 进行最小化。其四，根据 DFA 状态图生成词法分析器。

`void MainWindow::on_btn_analysis_clicked()`; 构建 NFA 图

`void generate_program()`; 根据 DFA 状态图生成词法分析器。

`void converttoDFA(QSet<int>begin)`; 将 NFA 转换成 DFA

`void converttoMDFA()`; 将 DFA 进行最小化

## 3. 项目实施

主要实现四个函数就行，然后保存好展示的数据。

`void MainWindow::on_btn_analysis_clicked()`; 实现思路，根据书本对|、连接、\*、? 的生产机器，进行状态连接，构建 NFA。由于运算符的运算顺序，采用了栈，来调整运算。栈里面存储的任意一个小部分 NFA 图的起点和终点，栈储存的类型。

```
struct ListNode{
    int first;
    int end;
    ListNode(int f=-1,int e=-1)
    {
        first=f;
        end=e;
    }
};
```

`void converttoDFA(QSet<int>begin)`; 主要根据上一步生成的 NFA，消除空字符，形成 DFA，空字符的闭包具体通过深度优先搜索实现。

`void converttoMDFA()`; 将 DFA 进行最小化。采取逆向思维，先假设只有 2 个状态，接受状态和非接受状态。然后通过字母表是否可以区别接受状态的任意 2 个 DFA 状态，若是则将其分裂，否则形成一组。非接受状态也是如此。

`void generate_program()`; 根据 DFA 状态生成代码，这一步比较简单，只需控制遍历各个 DFA 状态即可。

展现 DFA、NFA 也是比较简单，因为有了边集数组，直接遍历形成状态表，然后呈现就可以了。

具体实现看源程序。

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分

## 4. 测试内容

### 4.1 l(l|d)\*

DNF 最小化结果:



状态/字母表	d	l
-	1	2
+	2	2

生成的源程序:

```
#include <iostream>
#define ERROR -1
using namespace std;
int main()
{
    string str;
    cin>>str;
    int k = 0;
    int state = 1;
    while(state >= 1 && state <=2)
    {
        char c=str[k++];
        switch(state)
        {
            case 1:
                if(c == 'l')
                    state = 2 ;
```

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022

专 业 计算机科学与技术 年级、班级 计科1班

课程名称 编译原理实验 实验项目 XLEX-词法自动生成器

实验时间 2023 年 10 月 25 日

实验指导老师 黄煜廉 实验评分

```
else state = ERROR;
break;
case 2:
if(c == 'd')
state = 2 ;
else if(c == 'l')
state = 2 ;
else state = 3 ;
break;
}
}
if(state == 3)
cout<<"accept";
else cout<<"error";
}
```

4.2 a?b\*

DFA 最小化

状态/字母表		a	b
- +	1	2	2
+	2		2

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分

源程序：

```
#include <iostream>
#define ERROR -1
using namespace std;
int main()
{
    string str;
    cin>>str;
    int k = 0;
    int state = 1;
    while(state >= 1 && state <=2)
    {
        char c=str[k++];
        switch(state)
        {
            case 1:
                if(c == 'a')
                    state = 2 ;
                else if(c == 'b')
                    state = 2 ;
                else state = 3 ;
                break;
            case 2:
                if(c == 'b')
                    state = 2 ;
                else state = 3 ;
                break;
        }
    }
    if(state == 3)
        cout<<"accept";
    else cout<<"error";
}
```



# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分

4.3

$(a|b)^*((aa|bb)(a|b)^*)^*(a|b)^*((a|b)(a|b)(a|b)(a|b)(a|b))^?(a|b)^*((ab|ba)(ab|ba)^*)^*$

状态/字母表	a	b
0	0	1
1	0	1

以上测试成功。

## 四、实验总结（心得体会）

1. 理解了词法分析的重要性：词法分析是编译器的第一个阶段，它负责将源代码转化为一个个词法单元，为后续的语法分析提供基础。在本次实验中，我深入理解了词法分析的作用和原理，并通过实践加深了对其重要性的认识。
2. 明白了将一个复杂问题转换为几个简单问题的重要性。我们一开始很难根据正则表达式，直接画出最简的 DFA 图，但是我们在它们中间引入 NFA，

# 华南师范大学实验报告

学生姓名 钟文宏 学 号 20212121022  
专 业 计算机科学与技术 年级、班级 计科1班  
课程名称 编译原理实验 实验项目 XLEX-词法自动生成器  
实验时间 2023 年 10 月 25 日  
实验指导老师 黄煜廉 实验评分 \_\_\_\_\_

那么再得到最简的 DFA 图，只需将 NFA 化简为 DFA，然后 DFA 最小化。  
3. 对于 DFA、NFA 图的存储很考验数据结构的知识，选取不同的存储结构会有不同的思路 and 过程。

## 五、参考文献：

《编译原理及实践》