華南師範大學

本科学生实验（实践）报告

院　　系：计 算 机 学 院

实验课程：编译原理项目

实验项目：词法分析程序、语法分析程序

指导老师：黄煜廉

开课时间：2023 ～ 2024 年度第 2 学期

专　　业：计算机科学与技术（师范）

班　　级：21 级 师范 1　班

学生姓名：秦紫茉

华南师范大学教务处

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

一、项目内容

必做内容：

项目任务一任务要求：

（1）以文本文件的方式输入某一高级程序设计语言的所有单词对应的正则表达式，系统需要提供一个操作界面，让用户打开某一语言的所有单词对应正则表达式文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。

（2）需要提供窗口以便用户可以查看转换得到的 NFA（可用状态转换表呈现）

（3）需要提供窗口以便用户可以查看转换得到的 DFA（可用状态转换表呈现）

（4）需要提供窗口以便用户可以查看转换得到的最小化 DFA（可用状态转换表呈现）

（5）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 语言描述）[只能使用讲稿中的方法一或方法二来生成词法分析程序]

（6）对要求(5)得到的源程序进行编译生成一个可执行程序，并以该高级程序设计语言的一个源程序进行测试，输出该该源程序的单词编码。需要提供窗口以便用户可以查看该单词编码。

（7）对系统进行测试: （A）先以 TINY 语言的所有单词的正则表达式作为文本来测试，生成一个 TINY 语言的词法分析源程序；（B）接着对这个词法分析源程序利用 C/C++编译器进行编译，并生成可执行程序；（C）以 sample.tny 来测试，输出该 TINY 语言源程序的单词编码文件 sample.lex。

（8）要求应用程序为 Windows 界面

（9）书写完善的软件文档

项目任务二任务要求

（1）以文本文件的方式输入某一高级程序设计语言的所有语法对应的 BNF 文法，因此系统需要提供一个操作界面，让用户打开某一语言的所有语法对应的 BNF 文法的文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。

（2）求出文法的每个非终结符号的 First 集合和 Follow 集合，并需要提供窗口以便用户可以查看该结果（可用两张表格的形式分别进行呈现）

(3)需要提供窗口以便用户可以查看文法对应的 LR(0)DFA 图。（可以用画图的方式呈现，也可用表格方式呈现该图点与边数据）

（4）构造出 SLR(1)分析表，并需要提供窗口以便用户可以查看该结果（可用表格形式进行呈现）

（5）采用 SLR(1)语法分析方法进行语法分析并生成相应的语法树，每个语句的语法树结构可根据实际的需要进行定义。（语法树需要采用树状形式进行呈现）

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

（6）以 TINY 语言的所有语法以及第一项任务的测试结果 sample.lex 作为测试，并生成对应的语法树并呈现出来。
（7）要求应用程序为 Windows 界面
（8）书写完善的软件文档
（9）选做内容：可以生成某种中间代码【具体的中间代码形式可以自定】

项目任务三**任务要求**
mini-c 语言作为测试
（1）以 mini-c 的词法进行测试，并以至少一个 mini-c 源程序进行词法分析的测试（该 mini-c 源程序需要自己根据 mini-c 词法和语法编写出来，类似于 sample.tny）。
（2）以 mini-c 的语法进行测试，并以测试步骤（1）的源程序所生成的单词编码文件进行语法分析，生成对应的语法树。

选做内容：

1.采用 SLR(1)语法分析方法进行语法分析并生成相应的中间代码，中间代码可以自选（可以是四元组、三元组、伪代码等中间代码），中间代码生产结果可以在屏幕上显示或以文件的形式保存

二、项目目的

1.掌握词法分析程序的原理和算法，并能够编写一个能够正确识别源程序中各种词法单元的词法分析程序。通过实验，加深对编译原理中词法分析知识的理解和应用能力。
2.深入学习语法分析程序的原理和算法，理解上下文无关文法、语法分析树、LL(1)分析、LR(1)分析等相关概念和算法。

三、项目文档

（一）项目任务一

1.总体设计

1.1 需求概述

（1）以文本文件的方式输入某一高级程序设计语言的所有单词对应的正则表达式，系统需要提供一个操作界面，让用户打开某一语言的所有单词对应正则表达式文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。
（2）需要提供窗口以便用户可以查看转换得到的 NFA（可用状态转换表呈现）

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

（3）需要提供窗口以便用户可以查看转换得到的 DFA（可用状态转换表呈现）

（4）需要提供窗口以便用户可以查看转换得到的最小化 DFA（可用状态转换表呈现）

（5）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 语言描述）[只能使用讲稿中的方法一或方法二来生成词法分析程序]

（6）对要求(5)得到的源程序进行编译生成一个可执行程序，并以该高级程序设计语言的一个源程序进行测试，输出该该源程序的单词编码。需要提供窗口以便用户可以查看该单词编码。

1.2 处理流程

（1）处理正则表达式：将其由中缀转为后缀，添加“-”符号，表示正则表达式的连接运算，对正则表达式中可能出现的数字、字母以及运算符号进行相应的处理，方便后续处理

（2）正则表达式转 NFA：在 NFA 图中，首先构建初态和终态，然后根据第一步中优化后的正则表达式，在其中不断添加结点和按照它们之间的关系如闭包、选择、连接得到 NFA 图，将状态编号，并将编号与状态间的关系写入状态转换表

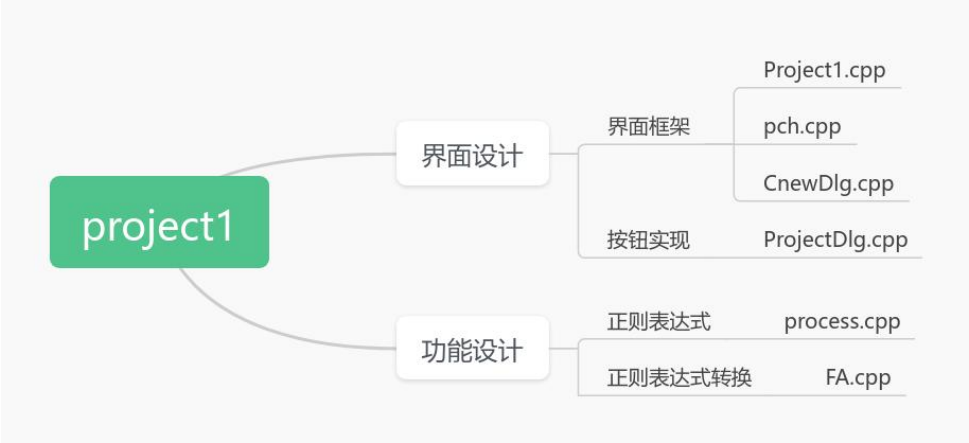
（3）NFA 转 DFA：主要是去除多重转化和 ε，使用子集法，也就是构建从初态开始的闭包，不断重复，直到状态表中数据不再发生变化，重新定义状态，并写入状态转换表

（4）最小化 DFA：将状态划分为两个等价的集合，然后不断迭代地将等价集合分为更小的等价集合，直到不能再分割为止。这样就得到了最小化的 DFA。

（5）生成语法分析程序：根据最小化 DFA 的状态转移表中的非终态和终态集合生成由 C 语言编写的词法分析程序，并保存 cpp 文件

（6）搭建程序界面并与对应函数连接：设计界面，添加按钮以实现实验要求的功能，并按照要求输入正则表达式输出 NFA,DFA,最小化 DFA，词法分析程序，并对词法分析程序进行编译根据输入的例子生成相应的单词编码

1.3 总体结构与模块外部设计



华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

2.数据结构设计

2.1 逻辑结构设计

2.1.1 正则表达式处理

（1）处理正则表达式

添加连接符号：因为正则表达式中的连接运算是隐含的，没有明确的符号表示，例如 ab 表示 a 和 b 的连接，但是在中缀转后缀的过程中，需要有一个明确的符号来表示连接运算的优先级，否则会导致歧义或错误。例如，如果没有添加连接符号，那么 a|b*的后缀表达式可能是 ab*|或者 a|b*，前者表示 a 或者 b 的闭包，后者表示 a 或者 b 的连接，这两者的含义是不同的。因此，为了避免这种情况，需要在正则表达式中添加连接符号，我采用的是“-”，表示连接运算的优先级高于或运算和闭包运算，比如 a(a|b)* ，加 ‘-’后的表达式：a-(a|b)*，表示 a 和(a|b)*是拼接起来的。

处理[0-9][a-Z]：将其展开成所有可能，即[0-3]转化为（0|1|2|3）的形式

（2）正则表达式中缀转后缀

中缀表达式即日常生活中的算术表达式的形式，其中运算符位于操作数的中间，其特点是容易阅读和理解，但在计算机程序中操作起来比较复杂。因此为了方便后续处理，将中缀表达转换为后缀表达，后缀表达式（也称为逆波兰表达式）是一种将运算符置于操作数之后的数学表达式表示方法。在后缀表达式中，运算符跟在操作数之后，因此不需要括号来表示运算优先级。

将中缀转为后缀，首先创建两个栈，运算符栈 s1 和数值栈 s2。从左到右扫描中缀表达式。遇到数的话直接假如到数栈 s2 中，遇到运算符的话，分为以下几种情况：

1） 如果 s1 为空，或者 s1 的栈顶为左括号“（”就直接将运算符入栈

2）如果 s1 不为空，扫描到的运算符的优先级高于栈顶运算符的优先级的话，也直接入栈

3）如果 s1 不为空，扫描到的运算符的优先级小于或等于栈顶运算符的优先级的话,就将 s1 中的运算符弹出并压入到 s2 栈中,然后再转到与 s1 中新的栈顶的运算符进行比较。

遇到括号的情况:

1）当遇到左括号时，直接入栈

2）当遇到右括号时，将 s1 栈中的符号依次弹出并压入到 s2 中，直到遇见左括号为止。然后将这一对括号丢弃

重复上述的步骤,直到扫描到表达式的最右边(即表达式最后),将 s1 中剩余的运算符依次弹出，并压入到 s2 中，此时的结果的逆序就是我们要的后缀表达式

华 南 师 范 大 学 实 验 报 告

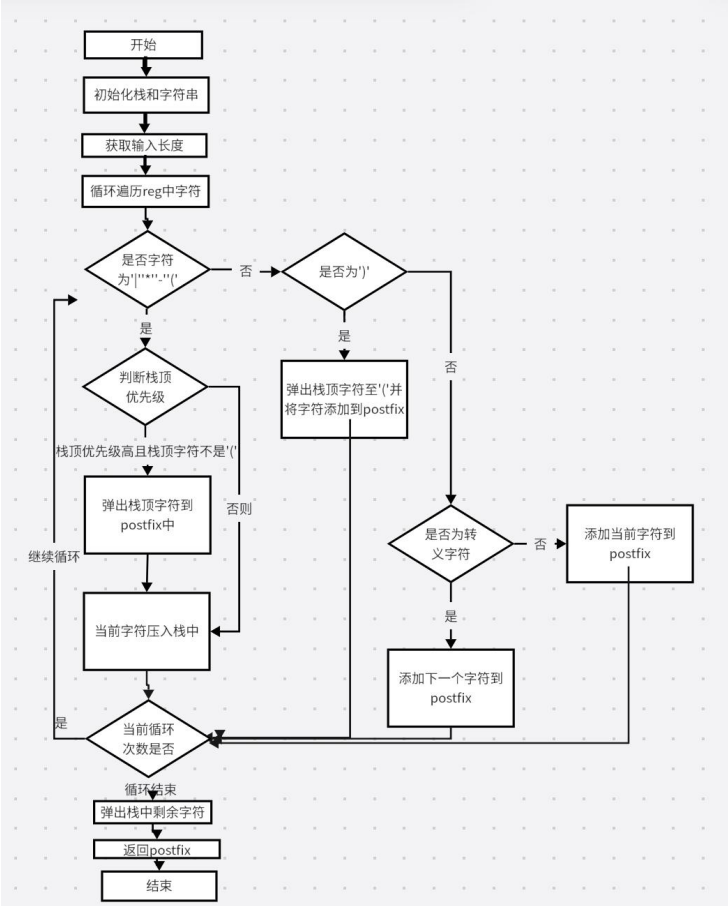
学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

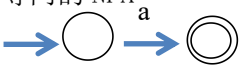


2.1.2 正则表达式转 NFA

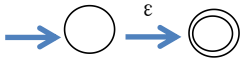
一个非确定有限自动机（NFA）M 是一个五元式即 M=（S，Σ，T，SO，A），其中，S 有穷状态集，Σ字母表，T 状态转换函数，SO 初态，A 终态

（1）基本正则表达式

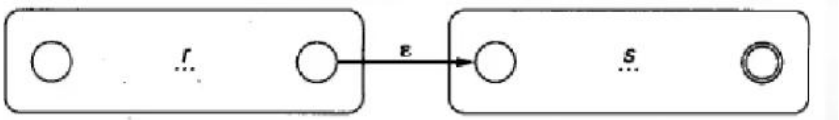
与正则表达式 a 等同的 NFA



与 ε 等同的符号



（2）并置　与 rs 对应的 NFA



（3）选择　与 r|s 对应的 NFA

华 南 师 范 大 学 实 验 报 告

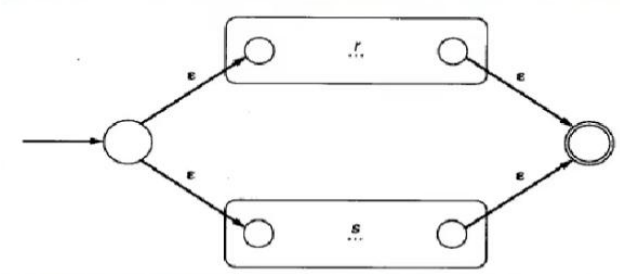
学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

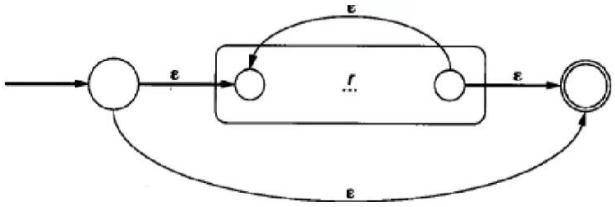
课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分



（4）闭包　与 r*对应的 NFA



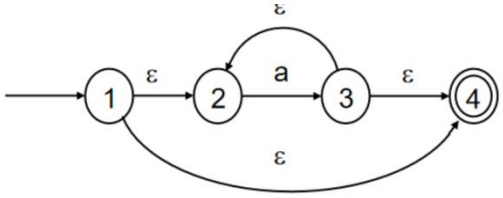
将正则表达式转化为 NFA，首先构建初态与终态，以及各结点状态，然后将结点间的关系按照上述对应关系进行转换，得到最后的状态图，将各状态标号后得到状态转移表

2.1.3 NFA 转 DFA

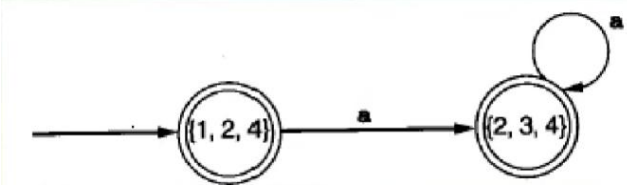
确定有限自动机（DFA）M 是一个五元式 M=（S，Σ，T，SO，A），其中，S 有穷状态集，Σ 字母表，T 状态转换函数，SO 初态，A 终态，NFA 与 DFA 不同的是 NFA 包含 ε 状态转换边和多重转化，因此想要将 NFA 转化为 DFA 就需要去除 NFA 中的多重转化以及 ε 边

采用的方法为子集法，即从初态结点 SO 开始，求状态集的 ε -闭包，并对状态集再次求 ε -闭包，重复此过程直到状态集不再发生改变，对每一个状态集重新定义一个状态，包含原终态的均为 DFA 的终态，最后得到 DFA

例如：NFA 图如下，将其转化为 DFA 图，首先求初态结点 1 的 ε -闭包并记录在表格内，对得到的状态集再次进行如上操作，直到不再发生改变，得到的最终表格以及 DFA 图如下：



	a
{1，2，4}	{2，3，4}
{2，3，4}	{2，3，4}



2.1.4 最小化 DFA

最小化 DFA 即得到一个状态数最少的 DFA，即将多余的、等价的状态进行合并

根据上面步骤中的状态转换表写出 DFA 表，确定 DFA 的初始状态和接受状态，将其拷贝到新的最小化 DFA 中。将 DFA 的状态按照它们是否可达性进行分类。即从初始状态出发，将所有可达的状态分为一组，所有不可达的状态分为另一组。针对每一组，对其中的状态进行等价性判断。两个状态等价，意味着它们在同样的输入下会转移到等价的状态。对于不同的输入，可能转移到不同的状态，持续合并等价的状态，直到无法再合并为止。这样得到的新 DFA 就是最小化的 DFA。更新最小化 DFA 的状态转移矩阵和接受状态集合。最后得到的最小化 DFA 具有最少可能的状态数，但保留了原始 DFA 的所有信息和行为。

2.1.5 生成词法分析程序

编写程序来实现对输入文本的扫描和识别。程序包括识别不同词法单元的函数，以及一个主程序来协调各个函数之间的调用。

2.2 物理结构设计

使用 map 存储 NFA、DFA 节点集合，两个容器 dfaNodes_m 和 dfaNodes_v 互相配合，在 NFA 转 DFA 过程中将 NFA 的节点集合编号，以构造 DFA

```
map<vector<int>, int> dfaNodes_m; // NFA 节点集合 映射 DFA 结点编号
vector<int> dfaNodes_v[5000];     // DFA 结点编号 映射 NFA 节点集合
map<string, vector<int>> hash_m;  // hash 值 映射 DFA 结点编号组
int belongTo[5000];               // 存储 DFA 最小化后需要进行的节点编号变换，如{1，2，3}确定为等价节点，由 belongTo[1]=belongTo[2]=belongTo[3]=1
```

使用 map 建立 NFA 图中终态与对应正则表达式名称的映射

```
map<int, string> acceptNodes;
```

使用 map 建立 DFA 图中接收态与对应正则表达式名称的联系

```
map<int, vector<string>> acceptDFANodes;
```

采用结构体形式建立状态转换边

```
struct edge
{
    char c;   // 边上对应的转换字符
    int dest; // 该边指向的节点
}
```

采用类的形式搭建 NFA 和 DFA 图

```
class NFA_DFA
{public:
    int getBeginNode();                          // 获取 NFA/DFA 图的初态节点编号
    int getEndNode();                            // 获取 NFA/DFA 图的终态节点编号
    void setBeginNode(int i);                    // 设置 NFA/DFA 图的初态节点
    void setEndNode(int i);                      // 设置 NFA/DFA 图的终态节点
    void addEdge(int begin, int end, char c); // 在 NFA/DFA 图中增加一条 c 的状态转换边
    void addNode(int i);                         // 在 NFA/DFA 图中增加一个节点
    NFA_DFA(int begin, int end);
    NFA_DFA() {};
    map<int, vector<edge>> edges;
    vector<int> getEclosure(const vector<int>& nodes);
    vector<int> c_transform(const vector<int>& nodes, char c);
```

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

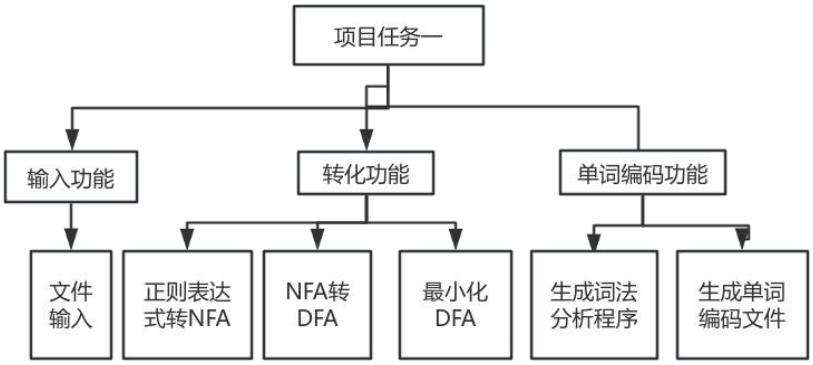
实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
    void print();
private:
    int begin; // 初态
    int end;   // 终态
};
```

3. 程序描述：

3.1 功能结构图



3.2 输入功能

3.2.1 性能

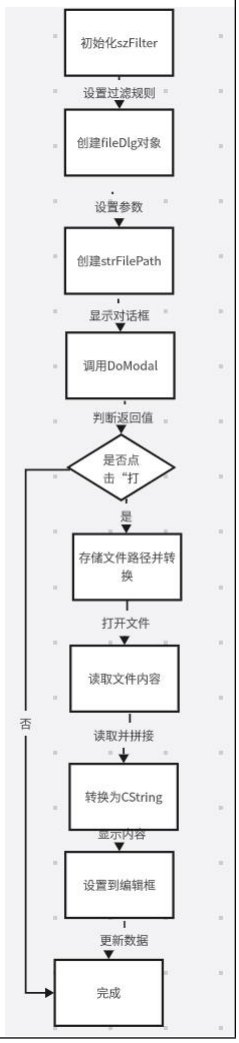
以文本文件的方式输入某一高级程序设计语言的所有单词对应的正则表达式，系统需要提供一个操作界面，让用户打开某一语言的所有单词对应正则表达式文本文件。

3.2.2 输入项目

用户可以在程序界面中点击“open file”按钮，选择文件路径，打开某一高级语言所有单词对应的正则表达式的文本文件，程序将会自动将文本文件中内容输入进程序并输出在相应对话框中。

3.2.3 算法

```
void CLexGeneratorDlg::OnBnClickedButton1()
{
	// TODO: 在此添加控件通知处理程序代码
	// 设置过滤器
	TCHAR szFilter[] = _T("文本文件(*.txt)|*.txt||");
	// 构造打开文件对话框
	CFileDialog fileDlg(TRUE, _T("txt"), NULL, 0, szFilter, this);
	CString strFilePath;
	// 显示打开文件对话框
	if (IDOK == fileDlg.DoModal())
	{
		// 如果点击了文件对话框上的“打开”按钮，则将选择的文件路径显示到编辑框里
		strFilePath = fileDlg.GetPathName();
		file_path = CT2A(strFilePath.GetString());
		ifstream ifs(file_path);
		string program = "";
		string line;
		while (!ifs.eof()) {
			getline(ifs, line);
			line += "\r\n";
			program += line;
```



华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
            }
            CString cstr = CString(program.c_str());
            //edit1.SetWindowTextW(cstr);
            GetDlgItem(IDC_EDIT1)->SetWindowText(cstr);
            UpdateData(FALSE);//将数据更新到控件上
        }
}
```

3.2.4 程序逻辑

设置文件过滤器，只能选择文本文件（*.txt）。构造打开文件对话框对象。显示打开文件对话框，并等待用户选择文件。如果用户点击了文件对话框上的“打开”按钮，则将选择的文件路径保存到 strFilePath 变量中。将选择的文件路径转换为标准字符串格式并保存到 file_path 变量中。使用 ifstream 打开文件，将文件内容逐行读取并保存到 program 变量中。将 program 的内容转换为 CString 类型，并设置到编辑框中。更新控件数据以将新内容显示到编辑框中。

3.3 转换功能

3.3.1 性能

正则表达式转 NFA

需要提供窗口以便用户可以查看转换得到的 NFA（可用状态转换表呈现）

NFA 转 DFA

需要提供窗口以便用户可以查看转换得到的 DFA（可用状态转换表呈现）

最小化 DFA

需要提供窗口以便用户可以查看转换得到的最小化 DFA（可用状态转换表呈现）

3.3.2 输入项目

上一个功能中经过处理的正则表达式

3.3.3 算法

正则表达式转 NFA

函数将文件中的若干行正则表达式转换为 NFA 图。首先打开文件并逐行读取正则表达式，然后对每行正则表达式进行处理，将其转换为后缀表达式。接着利用后缀表达式构建对应的 NFA 图，具体步骤如下：

1.创建一个空的 NFA 图，并设定其起始节点。

2.遍历后缀表达式中的每个字符，根据不同的操作符进行相应的操作：

如果是闭包运算符'*'，则弹出一个 NFA 图，调用闭包函数构造闭包操作后的 NFA 图，并将其压入栈中。

如果是连接运算符'-'，则依次弹出两个 NFA 图，调用连接函数构造连接操作后的 NFA 图，并将其压入栈中。

如果是选择运算符'|'，则依次弹出两个 NFA 图，调用选择函数构造选择操作后的 NFA 图，并将其压入栈中。

如果是转义符'/'，则读取下一个字符，新建一个 NFA 图表示该字符，并将其压入栈中。

如果是其他字符，新建一个 NFA 图表示该字符，并将其压入栈中。

3.处理完后缀表达式后，将得到的 NFA 图压入栈中，加入 NFA 图到最终的 NFA 图中，并建立

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

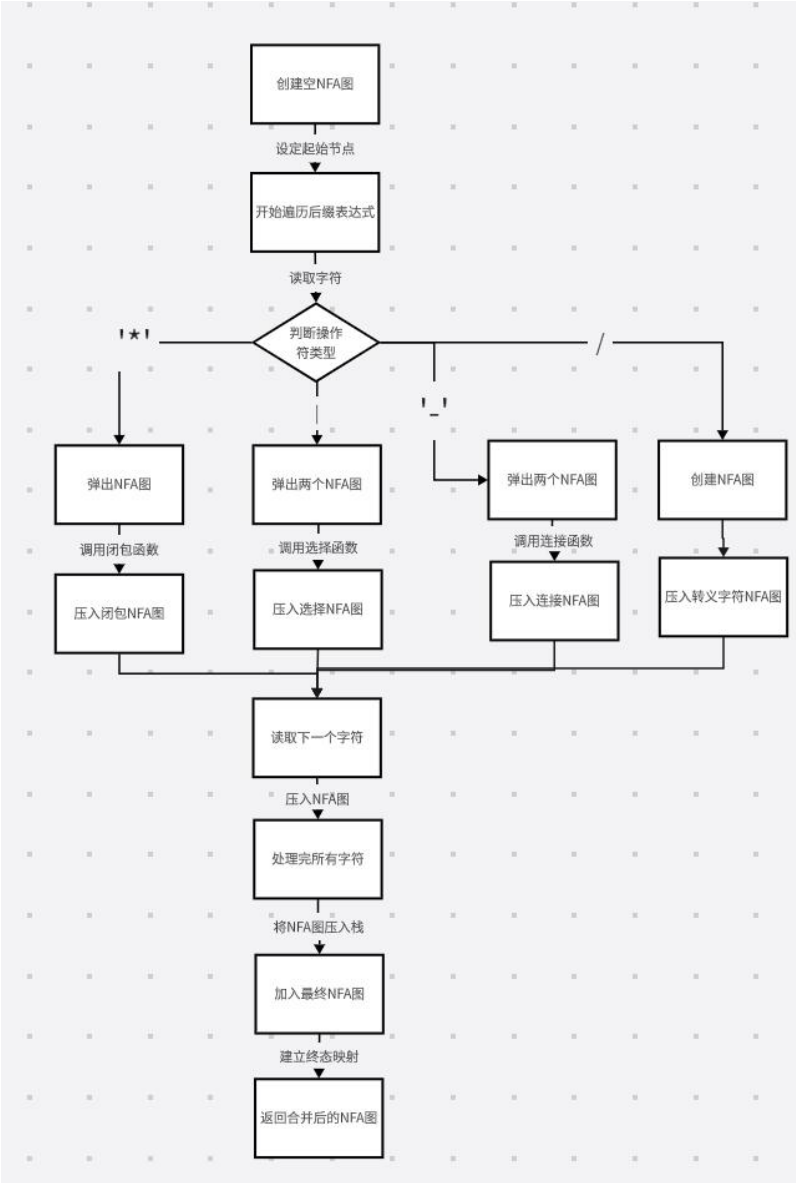
课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

终态与对应正则表达式名称的映射关系。

4.最后返回合并后的 NFA 图。



NFA 转 DFA

函数实现了将 NFA（非确定有限状态自动机）转换为 DFA（确定有限状态自动机）的功能。其主要逻辑如下：

1.首先求出初始状态的 ε -closure 闭包集合，并将其作为 DFA 的初始状态。

2.将初始状态的闭包集合与 DFA 结点编号建立映射关系，并入栈。

3.循环处理栈中的结点，对每个结点集合通过正则表达式上所有可能的字符进行转换计算。

4.对每个转换得到的集合求闭包集合，如果该闭包集合已经存在于 DFA 中，则只需添加转换

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

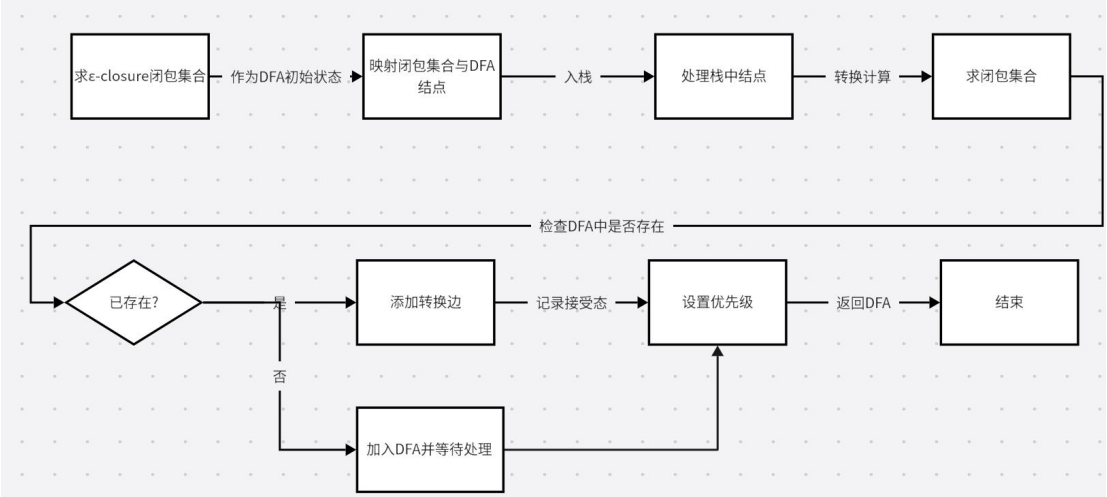
课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

边；如果不存在，则需要将该闭包集合和转换边加入 DFA 中，并将该闭包集合入队等待处理。在处理过程中记录 DFA 的接受态，并为设置关键字或其他正则表达式的接收态的优先级大于 id、letter、digit。

5. 最后，将转换得到的 DFA 返回。



最小化 DFA

函数实现将 DFA 图最小化的逻辑如下：

1 找出 DFA 图中所有的等价结点，并在等价结点中选择其一作为代表结点

1.1 遍历 DFA 图中所有的结点，然后计算每个结点的 hash 值

1.2 将所有 hash 一样的结点存储起来，记作等价结点集合

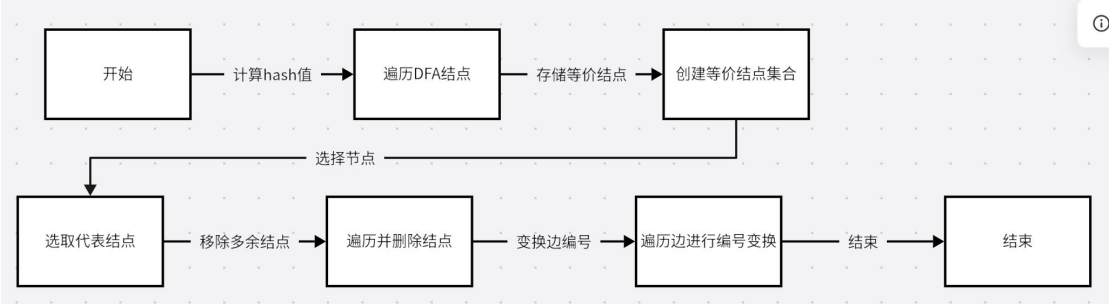
1.3 对所有的等价结点集合中均选取一个结点作为该等价结点结合的代表结点

2 移除所有的多余结点，即在所有的等价结点中，只保留代表结点

遍历 DFA 图中所有的结点，如果结点编号不为代表结点，则进行删除操作

3、在 DFA 图中的边存储结构中进行编号变换，将多余结点的编号用代表结点编号替换

遍历边上所有的结点编号，如果不为代表结点，则使用代表结点进行替换



3.4 单词编码功能

3.4.1 性能

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 语言描述）

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

[只能使用讲稿中的方法一或方法二来生成词法分析程序],并且能够利用生成的程序分析代码，生成单词编码

3.4.2 输入项目

上面步骤中所产生的最小 DFA 以及需要分析的程序

3.4.3 程序逻辑

template.cpp 中存储了固定的模板，如头文件引入，main 函数框架等，我们只需要在 template.cpp 补足 switch 语句，以实现词法分析。遍历 DFA 中的每一个结点，对于每一个结点均需要生成一个 case 语句，将结点编号作为 case 的编号，对于该结点的每条边均需要生成一个选择分支语句，作状态转换，遍历完该结点的所有边之后，需要判断三种情况：

1 该结点能够匹配任意字符：则收到其余字符时，生成一个 else 语句，将读入的字符保存，并且进行状态转移

2 该结点为接受态：收到其余字符时，需要把当前解析好的词语（lexeme）进行输出，然后状态转移至初始态 0，清空缓存，并且将多读入的字符做回退处理

3 该结点非接受态：收到其余字符时，需要输出报错信息：无法解析该词语，如果该结点至少存在一条边，而且不是这种情况（只有一条边，且该边匹配任意字符），则要加上 else，其余部分直接输出即可。

4.测试用例

4.1 输入（数据和命令）

（A）先以 TINY 语言的所有单词的正则表达式作为文本来测试，生成一个 TINY 语言的词法分析源程序；

（B）接着对这个词法分析源程序利用 C/C++编译器进行编译，并生成可执行程序；

（C）以 sample.tny 来测试，输出该 TINY 语言源程序的单词编码文件 sample.lex。

TINY 语言所有单词的正则表达式如下：

```
identifier ([a-zA-Z])([a-zA-Z])*
number ([0-9])([0-9])*
reserve_word if|then|else|end|repeat|until|read|write
op +|\-|\*|/
relationOp <|=
assignOp :=
mark ;
note \{(.)*\}
```

对词法分析程序的测试用例为：

```
read x; { input an integer }
if 0 < x then { don't compute if x <= 0 }
```

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
  Fact := 1 + 2 * 3 - 8 / 4 ;
  repeat
    Fact := Fact * x;
    x := x - 1
  until x = 0;
  write Fact  { output factorial of x }
end
```

4.2 输出

正则表达式转 NFA：

	*	+	-	/	0	1	2	3
1(-)								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

	*	+	-	/	0	1	2	3
565								
567								
569								
571								
572								
573								
574		575						
575								
576			577					
577								
578								
579								
580	581							
581								
582								
583								
584				585				
585								
586								
587								
588								
589								

NFA 转 DFA：

	*	+	-	/	0	1	2	3
0(-)	1	2	3	4	5	6	7	8
1(+)								
2(+)								
3(+)								
4(+)								
5(+)					72	73	74	75
6(+)					72	73	74	75
7(+)					72	73	74	75
8(+)					72	73	74	75
9(+)					72	73	74	75
10(+)					72	73	74	75
11(+)					72	73	74	75
12(+)					72	73	74	75
13(+)					72	73	74	75
14(+)					72	73	74	75
15								
16(+)								
17(+)								
18(+)								
19(+)								
20(+)								
21(+)								

t	u	v	w	x	y	z	{	}	&
64	65	66	67	68	69	70	71		
128	129	130	131	132	133	134			
128	129	130	131	132	133	134			
128	129	130	131	132	133	134			

最小化 DFA：

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

		*	+	-	/	0	1	2	3	t	u	v	w	x	y	z	{	}	&
0(-)		1	1	1	1	5	5	5	5	64	65	19	67	19	19	19	71		
1(+)																			
5(+)						5	5	5	5										
15																			
16(+)																			
17(+)																			
19(+)										19	19	19	19	19	19	19			
49(+)										19	19	19	19	19	19	19			
53(+)										19	19	19	19	19	19	19			
62(+)										19	19	19	19	19	19	19			
64(+)										19	19	19	19	19	19	19			
65(+)										19	19	19	19	19	19	19			
67(+)										19	19	19	19	19	19	19			
71	71																	143	
82(+)																			
135(+)										19	19	19	19	19	19	19			
136(+)										19	19	19	19	19	19	19			
137(+)										19	19	19	19	19	19	19			
138(+)										19	19	19	19	19	19	19			
139(+)										19	19	19	19	19	19	19			
140(+)										149	19	19	19	19	19	19			
141(+)										19	19	19	19	19	19	19			

词法分析程序：

词法分析程序

```
#include <cstdio>
#include <iostream>
#include <string>
using namespace std;

int state = 0;
int main(){
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
    char cur;
    string lexeme;
    while((cur=cin.get())!=EOF){
        //ignore blank
        if((cur == '\n'|| cur == '\r'|| cur == '\t'|| cur==' ') && !state)
            continue;
        switch (state)
        {
        case 0:
            if(cur=='*'){
                state=1; lexeme+=cur;
            }
            else if(cur=='+'){
                state=1; lexeme+=cur;
            }
            else if(cur=='-'){
                state=1; lexeme+=cur;
            }
            else if(cur=='/'){
                state=1; lexeme+=cur;
            }
            else if(cur=='0'){
                state=5; lexeme+=cur;
            }
```

单词编码：

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
keyword: read
identifier: x
op: ;
keyword: if
number: 0
op: <
identifier: x
keyword: then
identifier: x
op: :=
identifier: x
op: +
number: 2
keyword: end
op: ;
keyword: repeat
identifier: fact
op: :=
identifier: fact
op: *
identifier: x
op: ;
identifier: x
op: :=
identifier: x
op: -
number: 1
keyword: until
identifier: x
```

4.3 步骤及操作

界面展示：

1. 点击“打开文件”按钮，打开文件路径，选择先要进行分析的高级语言所有单词的正则表达式文本文件，点击“打开”，将文件内容输入进程序，并展示在左边文本框
2. 依次点击“NFA”“DFA”“minDFA”按钮，将正则表达式转化为 NFA,DFA,最小化 DFA 的结果展示在右边文本框
3. 点击“词法分析程序”按钮，展示词法分析程序结果
4. 运行词法分析程序，程序会自动将同文件夹下的 input 文件中的源程序进行分析并将结果输出在 output 文件中
5. 点击“打开文件”按钮，展示存储在同路径下的单词编码文本文件 output.txt

5. 评价：

5.1 软件的能力

可以实现实验要求中必做内容的全部要求，界面简洁，简单易使用。

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

5.2 缺陷和限制

程序不能直接编译所生成的词法分析程序，需要其他软件运行文件，同时 cpp 文件与输入输出文件必须在同一路径下。

5.3 建议

可以改进程序，使程序可以实现由用户决定是否编译程序的功能

5.4 测试结论（说明能否通过）

程序通过全部测试

（二）项目任务二

1.总体设计

1.1 需求概述

（1）以文本文件的方式输入某一高级程序设计语言的所有语法对应的 BNF 文法，因此系统需要提供一个操作界面，让用户打开某一语言的所有语法对应的 BNF 文法的文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。

（2）求出文法的每个非终结符号的 First 集合和 Follow 集合，并需要提供窗口以便用户可以查看该结果（可用两张表格的形式分别进行呈现）

(3)需要提供窗口以便用户可以查看文法对应的 LR(0)DFA 图。（可以用画图的方式呈现，也可用表格方式呈现该图点与边数据）

（4）构造出 SLR(1)分析表，并需要提供窗口以便用户可以查看该结果（可用表格形式进行呈现）

（5）采用 SLR(1)语法分析方法进行语法分析并生成相应的语法树，每个语句的语法树结构可根据实际的需要进行定义。 （语法树需要采用树状形式进行呈现）

（6）以 TINY 语言的所有语法以及第一项任务的测试结果 sample.lex 作为测试，并生成对应的语法树并呈现出来。

1.2 总体结构与模块外部设计

界面设计：widgt.cpp

功能设计：grammar.cpp

2.数据结构设计

2.1 逻辑结构设计

2.1.1 预处理文法

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

按行分割文法，采用字符串流方法，分割出每一个产生式并存储在可变数组中

添加起始非终结符，并对文法进行从 0 开始的编号，方便构建后续结构

文法检查扩充，判断是否文法开始符号对应的文法规则是否有多条，即判断文法中有没有"|"，如果存在“|”，则对文法进行扩充。对文法进行遍历访问求得其非终结符与终结符集合

2.1.2 求文法的 first 集合，follow 集合

first 集合，设 G=(VT，VN，S，P)是上下文无关文法 ，FIRST(α)={a|α 能推导出 aβ,a∈VT， α，β∈V*}，特别的，若α能推导出ε,则规定ε∈FIRST(α).

求 first 集合步骤如下：

1.若 X∈VT，则 FIRST(X)＝{X}。(简单讲，终结符的 FIRST 集就是它本身)

2.若 X∈VN，且有产生式 X→a…，a∈VT， 则 a∈FIRST(X)X→ε,则ε∈FIRST(X)。 （简单讲，若是非终结符 X，能推导出以终结符 a 开头的串，那么这个终结符 a 属于 FIRST（X），若 X 能够推导出空符号串 ε，那么空符号串 ε 也属于 X 的 FIRST 集）

3.X→Y…是一个产生式且 Y∈VN 则把 FIRST(Y)中的所有非空符号串 ε 元素都加入到 FIRST(X)中。

4.若 X∈VN；Y1，Y2，…，Yi∈VN，且有产生式 X→Y1 Y2 … Yn；当 Y1 Y2 … Yn-1 都能推导出 ε 时，则 FIRST(Y1)、FIRST(Y2)、…、FIRST(Yn-1)的所有非空元素和 FIRST(Yn) 包含在 FIRST(X)中。即： FIRST(X)=(FIRST(Y1)－{ε}) ∪ （FIRST(Y2)－{ε}）∪…∪（FIRST(Yn-1)－{ε}）∪{FIRST(Yn)}

5.当(4)中所有 Yi 能够推导出 ε,(i=1,2,…n)， 则 FIRST(X)=(FIRST(Y1)－{ε}) ∪（FIRST(Y2)－ {ε}∪…∪（FIRST(Yn) －{ε}）∪{ε}

反复使用上述步骤直到每个符号的 FIRST 集合不再增大为止。

follow 集合也就是紧接非终结符后面的终结符的集合称为非终结符的后继符号集，记为 FOLLOW(该非终结符)

求解 follow 集合步骤如下：

1.设 S 为文法中开始符号，把{#}加入 FOLLOW(S)中(这里“#”为句子括号)。

2.若 A→αBβ 是一个产生式，则把 FIRST(β)的非空元素加入 FOLLOW(B)中。如果β能够推导出 ε 则把 FOLLOW(A)也加入 FOLLOW(B)中。

3.反复使用(b)直到每个非终结符的 FOLLOW 集不再增大为止。

2.1.3LR（0）DFA 图

LR 分析法，L：对输入进行从左到右的扫描 R：反向构造出一个最右推导序列 LR(k) 需要向前查看 k 个输入符号的 LR 分析,一般 k = 0 或者 k = 1

初始项目：圆点在最左边，规约项目：圆点在最右边，后继项目：两个项目之间间隔一个圆点

2.1.4SLR（1）分析表

为了解决 LR(0)分析过程中的冲突的问题(移进归约冲突、归约归约冲突),出现了 SLR 分析，也叫做 SLR(1)分析，即在不知道是否移进或者规约时,向后查看一个输入符号，看规约项目产生式左部的非终结符的 FOLLOW 集，如果下一个输入符号在 FOLLOW 集中就规约，看移进项

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

目的下一个终结符是和下一个输入符号一样，如果相同就移进。

对于每一个状态，如果一个状态 a 吸收了一个终结符转移到另一个状态 b 的话，那么就在状态 a 那一行的 action 那里写上即可，对于结尾的状态，我们在#那一栏写 acc，表示接受，这里的话就是状态 1 表示的项目，如果一个状态时归约项目，那么就去我们改写的扩广文法那里找它是第几个产生式，然后前面加 r 即可。如果一个状态 a 吸收了一个非终结符转移到另一个状态 b 的话，就在状态 a 那一行的 goto 那里写上 b 即可。

2.1.5 语法树

在语法分析的过程中，需要根据语法规则构建语法树的节点，每个节点代表一个语法结构，如表达式、语句、函数等。节点之间通过父子关系来表示语法结构的层次关系。

在构建语法树的过程中，需要考虑语法规则的优先级和结合性，确保语法树能够正确地反映源代码的语法结构。

2.2 物理结构设计

采用类的形式表示语法结构

```
class Grammar {
private:
    map<string, vector<vector<string>>> m; //记录文法的每一条产生式，通过键值对的方式
    string start;  //记录开始符号
    set<string> ntSet; //非终结符集合
    set<string> tSet;//终结符集合
    map<string, set<string>> m_first; //记录每一个非终结符号对应的 first 集合
    map<string, set<string>> m_follow; //记录每一个非终结符对应的 follow 集合
    vector<vector<Project>> DFA_nodes; //记录 DFA 结点
    map<int, map<string, int>> forwards; //记录移进关系
    map<int, map<string, int>> backs; //记录归约关系
    bool isSLR1; //是否为 SLR(1)文法
    string reason; //不是 SLR(1)文法的原因

    void first_follow_all(); //求所有非终结符的 first 和 follow 函数
    set<string> first(vector<string> value); //求 value 序列的 first 集合
    void node_relationship(); //求 DFA 图，顺便把归约关系也求了，顺便判断是否为 SLR1
    void extend(int k); //扩展结点 k
public:
    Grammar(string inputString = ""); //构造函数(完成初始化操作)
    string get_start(); //得到开始符号
    set<string> get_ntSet();  //得到非终结符集
```

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
    set<string>get_tSet();   //得到终结符集
    set<string> get_first(string key); //得到非终结符 key 的 first 集合
    set<string> get_follow(string key);//得到非终结符 key 的 follow 集合
};
采用类表示 DFA 图结构
class Project { //项目
public:
    string key; //项目的 key
    int value_num; //项目 key 对应的 value 的编号
    int index; //所处位置
    int type; //1 为移进项 2 为归约项
    Project(string a = "", int b = 0, int c = 0, int d = 1) {
        this->key = a;
        this->value_num = b;
        this->index = c;
        this->type = d;
    }
    bool operator == (const Project& other) {
        return (this->key == other.key) && (this->value_num == other.value_num) &&
(this->index == other.index) && (this->type == other.type);
    }
};
采用结构体表示语法树结构
struct TreeNode{
    int childNum;
    string key;
    string content;
    vector<TreeNode*>child;
    bool is_FinalWord;
};
```

3.程序描述：

3.1 功能结构图

华 南 师 范 大 学 实 验 报 告

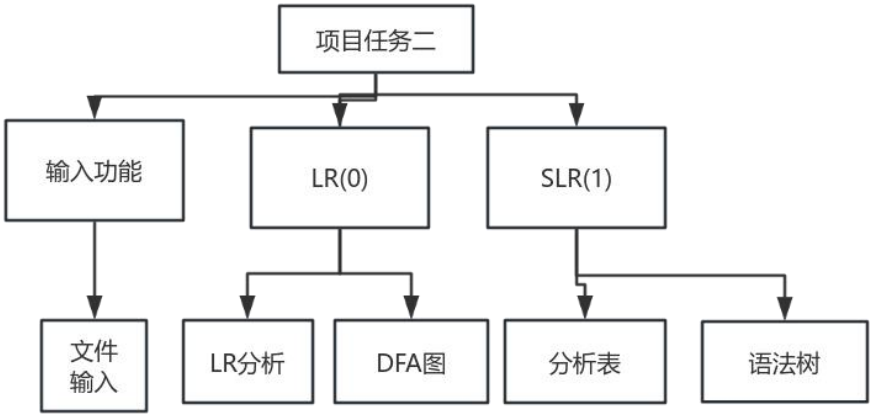
学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分



3.2 求 first、follow 集合

3.2.1 输入项目

以文本文件的方式输入某一高级程序设计语言的所有语法对应的 BNF 文法，因此系统需要提供一个操作界面，让用户打开某一语言的所有语法对应的 BNF 文法的文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。

求出文法的每个非终结符号的 First 集合和 Follow 集合，并需要提供窗口以便用户可以查看该结果（可用两张表格的形式分别进行呈现）

3.2.2 算法

预处理文法：

```
vector<string> v;
if (v[0].find('|') != string::npos) {
        v.insert(v.begin(), string(s + "'" + " -> " + s));
        s += "'";
    }
    this->start = s;      //更新文法开始符号
```

求非终结符集

```
    for (auto it = m.begin(); it != m.end(); ++it) {
        this->ntSet.insert(it->first);
    }
```

求终结符集

```
    for(const auto&entry:m){
        for(const auto&production:entry.second){
            for(const auto&symbol:production){
                if(ntSet.find(symbol)==ntSet.end())
                    tSet.insert(symbol);
```

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

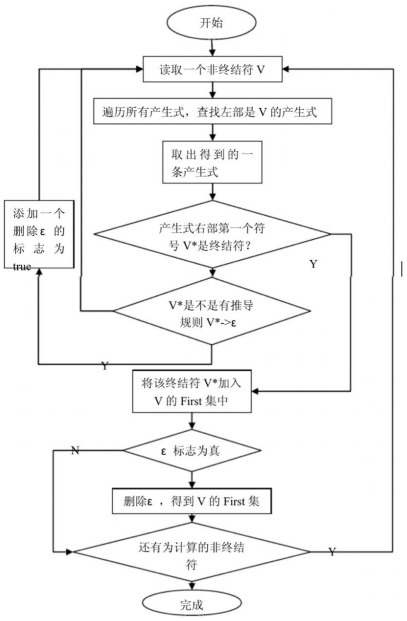
```
            }
        }
    }
```

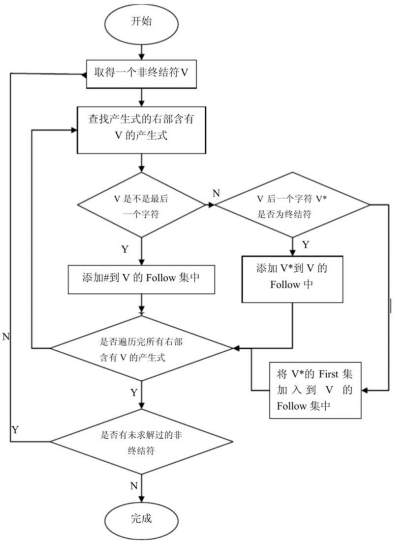
求 first 集合

```
 for (k = 0; k < value.size(); ++k) {
       auto&&          first_set_k          =
this->get_first(value[k]); //获取产生式右部每
个元素的 first 集
       for (auto& s : first_set_k) {
             if    ((s    !=    "@")    &&
(this->m_first[key].find(s)                 ==
this->m_first[key].end())) {

this->m_first[key].insert(s);
                    isChange = true;
              }
         }
        if       (first_set_k.find("@")       ==
first_set_k.end()) {  //未找到空串结束当前循环
           break;
          }
         }


If(k==value.size()and(this->m_first[key].find("
@")==this->m_first[key].end())) {  //假设当前产
生式中的每个元素的 first 中都有空串，然后该产生式
左部非终结符对应的 first 没有空串则添加空串
        this->m_first[key].insert("@");
        isChange = true;
}
```





3.2.3 程序逻辑

计算所有非终结符的 First 集合的。首先，对所有非终结符进行初始化，将它们的 First 集合都设为空集。然后通过一个循环遍历所有产生式，对每个产生式的右部进行遍历，计算每个符号的 First 集合，并根据计算结果更新每个产生式左部对应的 First 集合。在计算过程中，如果发现 First 集合发生了变化，就将 isChange 设置为 true，并在下一轮循环中继续计算。直到所有非终结符的 First 集合不再发生变化，循环结束。整个算法相当于对每个产生式进行迭代计算，直到不再有变化为止，保证所有非终结符的 First 集合都是正确的。

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

3.3LR（0）DFA 图

3.3.1 输入项目

需要提供窗口以便用户可以查看文法对应的 LR(0)DFA 图。（可以用画图的方式呈现，也可用表格方式呈现该图点与边数据）

3.3.2 算法

```
void Grammar::get_lr(vector<vector<string> > &table)
{
    set<string> sSet;       //表头，从移进项和归约项里面找
    for (auto& x : this->forwards) {
        for (auto& y : x.second) {
            sSet.insert(y.first);
        }
    }
    vector<string> header;
    header.push_back(" ");
    for(auto&s: sSet)
        header.push_back(s);

    table.resize((this->forwards.size()),    std::vector<string>(header.size(),
""));
    table.push_back(header);


    for (auto& x : this->forwards) {
        int from = x.first;
        table[from][0]=to_string(from);
        for (auto& y : x.second) {
            string t = y.first;
            int to = y.second;
            auto it = find(header.begin(),header.end(),t);
            int index=distance(header.begin(),it);
            table[from][index]=to_string(to);
        }
    }
}
```

3.3.3 程序逻辑

1.创建一个空集合 sSet，用于存储表头的内容，即移进项和归约项。

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

2.遍历文法的前瞻集合（forwards），将其中的移进项和归约项的符号添加到 sSet 中。

3.创建一个包含空格的字符串向量 header，并将 sSet 中的符号依次添加到向量中。

4.根据文法的前瞻集合的大小，调整 LR 分析表的大小，并将 header 向量作为 LR 分析表的第一行。

5.对于文法的前瞻集合中的每一个项目，将其状态和对应的移进项或归约项填入 LR 分析表中的相应位置

经过以上步骤，程序将完成 LR 分析表的构建工作，并返回包含 LR 分析表内容的向量 table。对文法进行分析，通过构建 DFA 图来检测文法是否满足 SLR(1)文法的要求。在函数中，首先初始化一个 DFA 结点并将其添加到 DFA 节点列表中，然后循环处理每个结点，扩展结点并处理其中的每个项目。通过对移进项和归约项的处理，来判断文法是否满足 SLR(1)文法的要求。如果在处理过程中发现冲突，则将 isSLR1 标记为 false，并记录冲突的原因。最后，在检查所有结点后，再次检查是否存在移进-归约冲突，如果存在冲突则将 isSLR1 标记为 false，并记录冲突的原因。最终返回文法是否为 SLR(1)文法的判断结果。

3.4SLR（1）分析表

3.4.1 输入项目

构造出 SLR(1)分析表，并需要提供窗口以便用户可以查看该结果（可用表格形式进行呈现

3.4.2 程序逻辑

实现 SLR 分析表的构建，它首先创建一个表头，然后遍历文法的移进项和归约项，将所有可能的状态符号添加到表头中。然后，对于每个 DFA 节点，它将计算相应的移进项和归约项，根据这些项填充表格的对应单元格。最后，将填充好的行添加到最终的分析表中。整体逻辑是生成 SLR 分析表所需的所有信息的结构，并将它们组织在正确的行和列中。

3.5 语法树

3.5.1 输入项目

采用 SLR(1)语法分析方法进行语法分析并生成相应的语法树，每个语句的语法树结构可根据实际的需要进行定义。 （语法树需要采用树状形式进行呈现）

3.5.2 算法

```
if(forwards[f].find(ts1)!=forwards[f].end())
        {
            q.erase(q.begin());
            t=forwards[f][ts1];
            TreeNode* tt1=new TreeNode();
            tt1->key=ts.type;
            tt1->content=ts.s;
            tt1->is_FinalWord=true;
            st.insert(st.begin(),tt1);
```

学生姓名 秦紫茉 学 号 20212121030

专 业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

```
            TreeNode*tt2 = new TreeNode();
            tt2->content=to_string(t);
            tt2->key="other";
            st.insert(st.begin(),tt2);
        }
```

3.5.3 程序逻辑

自底向上的语法分析方法，通过不断更新分析栈中的节点，逐步构建出完整的语法树结构

1.将输入内容转换为 token 序列，并添加一个表示结束的“$” token。

2.初始化输入队列、分析栈和根节点,将根节点入栈。

3.不断循环直到输入队列为空：

根据当前状态和待处理 token 确定转移规则。

如果是移进状态，则进行状态转换，并在分析栈中添加相应节点。

如果是归约状态，则进行规约操作，并在分析栈中生成相应的非终结符节点，并将子节点连接到该节点上。

如果出现其他情况（语法错误），则输出错误信息并结束循环。

3.最终返回构建好的语法树根节点。

4.测试用例

4.1 输入（数据和命令）

以 TINY 语言的所有语法以及第一项任务的测试结果 sample.lex 作为测试,并生成对应的语法树并呈现出来。

```
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 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
```

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

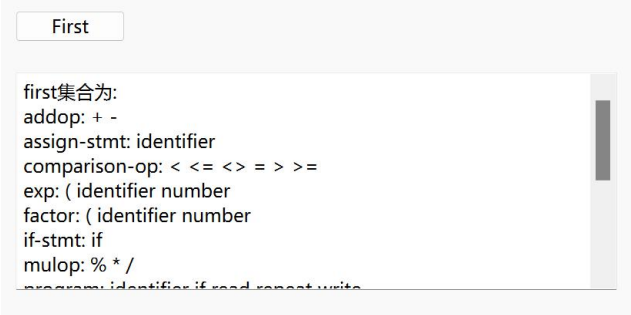
课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

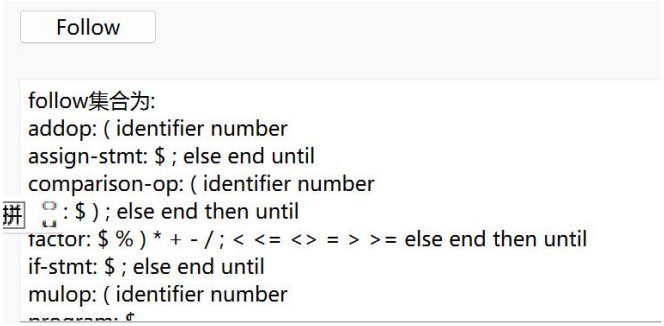
实验指导老师 黄煜廉 实验评分

4.2 输出

first 集合：



follow 集合：



LR(0)DFA

SLR(1)分析表：

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

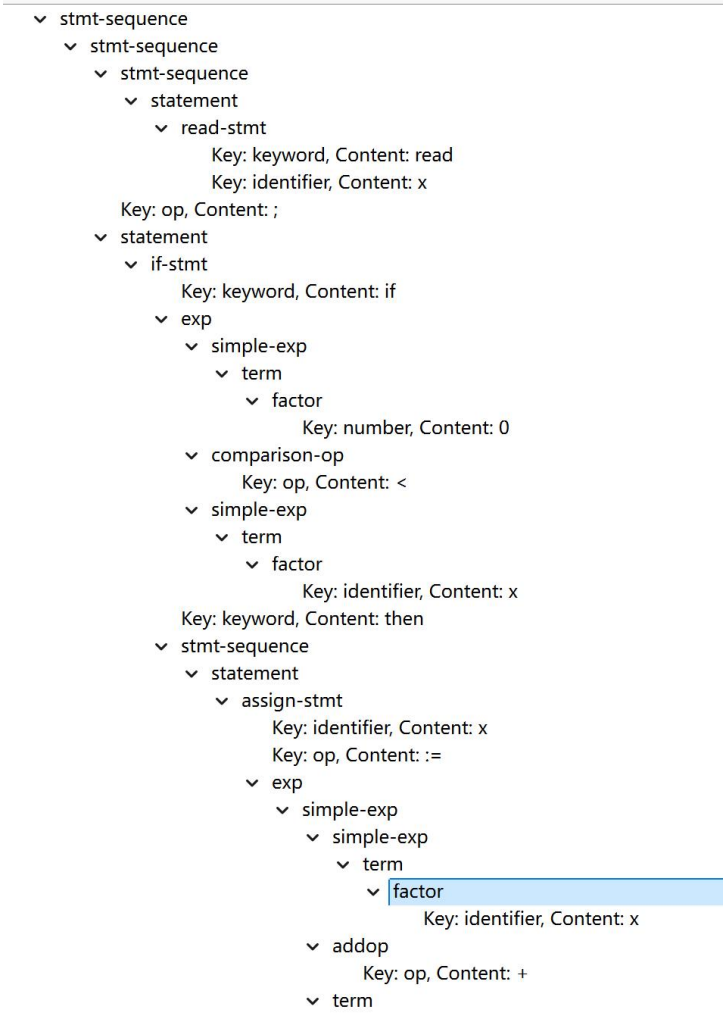
课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

	$	%	(	)	*
0					
1	acc				
2	r(stmt-...				
3	r(statement -...				
4	r(statement -...				
5	r(statement -...				
6	r(statement -...				
7	r(statement -...				
8			s18		

语法树：



4.3 步骤及操作

1. 点击“导入文法”按钮，点击目录下的文法文件，点击“打开”按钮，将文件导入程序
2. 点击“分析文法”按钮，程序对输入的文法进行分析

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

3. 依次点击“First”“Follow”“LR(0)DFA 图结点”“LR(0)DFA 图有向边”“LR(0)”“判断 SLR(1)”“SLR(1)分析表”在对应下方文本框得到相应数据

4.点击“语法分析”按钮，点击“编码文件”按钮打开在项目一中得到了编码文件，将其导入进程序，点击“语法分析”在下方文本框得到进行分析时的相关数据

5.点击“语法树”按钮，得到程序生成的语法树数据

项目任务三：mini-c 语言作为测试

（1）以 mini-c 的词法进行测试，并以至少一个 mini-c 源程序进行词法分析的测试（该 mini-c 源程序需要自己根据 mini-c 词法和语法编写出来，类似于 sample.tny）。

Mini-c 语法如下：

```
id ([a-zA-Z]|_)([a-zA-Z0-9]|_)*
num ([0-9])([0-9])*|([0-9])([0-9])*\.([0-9])([0-9])*
reserve_word else|if|int|float|return|void|do|while
op +|\-|\*|/|%
relationOp <|<=|>|>=|==|!=
assignOp =
mark ;|_|,|\(|\)|\[|\]|\{|\}
note //(.)*\.
```

词法分析程序的测试用例如下：

```
int esef = 5 ;
float Ts = 2 ;
do {
    Ts = Ts + 1 * 4 - 9 / 3 ;
    esef = esef - (1 + 9) % 4 ;
} while (esef > 0);
if (Ts > 0 )
    TS = Ts -1;
else
    Ts = Ts +1;
```

华 南 师 范 大 学 实 验 报 告

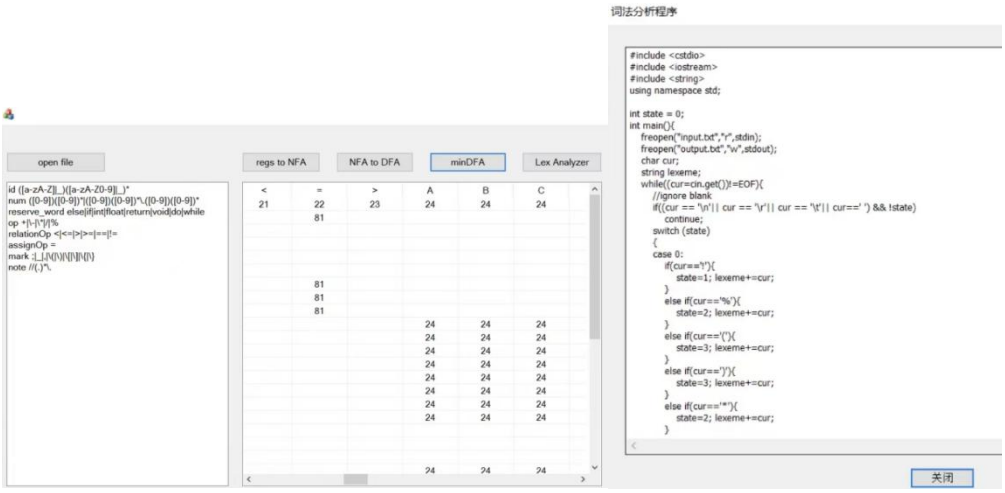
学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分



（2）以 mini-c 的语法进行测试，并以测试步骤（1）的源程序所生成的单词编码文件进行语法分析，生成对应的语法树。

first 集合：

first集合为:
additive-expression: (ID NUM
addop: + -
argument-list: (ID NUM
arguments: (@ ID NUM
call: ID
compound-stmt: {
condition-stmt: if

follow 集合：

follow集合为:
additive-expression: !=) + , - ; < <= == > >=]
addop: (ID NUM
argument-list:) ,
uments:)
call: != %) * + , - / ; < <= == > >=]
compound-stmt: $ (; ID NUM do double else float if int return
void while { }

LR(0)DFA

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

SLR(1)分析表：

	!=	$	%	(	)
0					
1		acc			
2		r(definition-lis...			
3		r(definition ->...			
4		r(definition ->...			
5					
6					
7					

语法树：

华 南 师 范 大 学 实 验 报 告

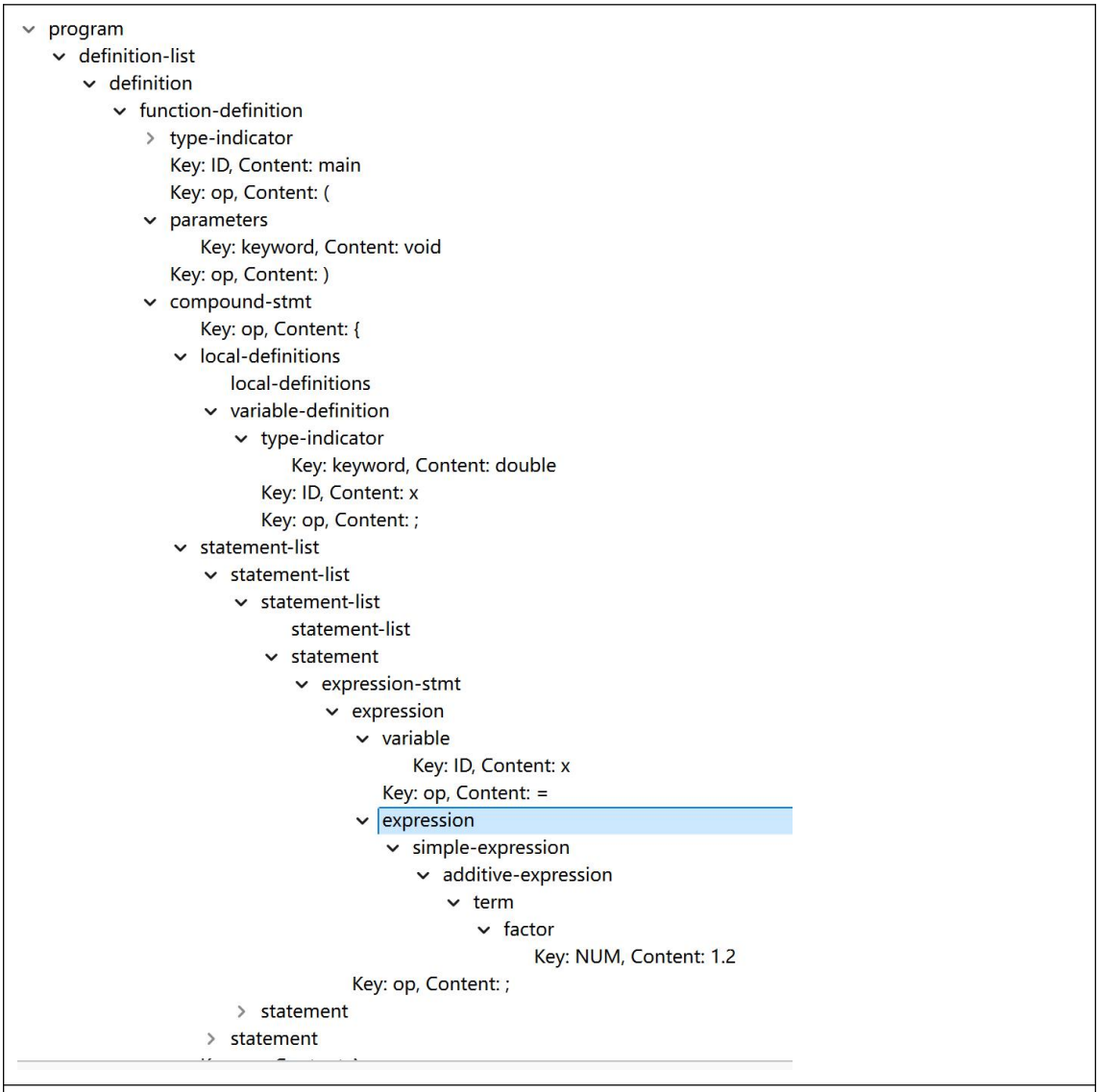
学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分



四、实验总结（心得体会）

在项目任务一中，我们深入研究了词法分析程序的原理和算法，并成功地编写了一个简单的词法分析程序。通过实验过程中的反复测试和调试，我们加深了对词法分析知识的理解和应用，发现了一些词法分析程序中可能出现的问题和改进方向。同时也意识到词法分析程序的

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

性能和效率对编译器整体的质量和效率有着重要影响。在未来的学习和实践中，我们将继续深入学习编译原理知识，不断完善和优化词法分析程序，提高编译器的质量和性能。

通过项目任务二，我们对编译原理中的语法分析有了更深入的认识和理解，也增强了对编程实践的信心和能力。在今后的学习和实践中，我们将继续努力，为提高编译器的质量和性能不断努力。

五、参考文献

编译原理基本定义(LR(0)与 SLR(1)) lr0 和 slr1-CSDN 博客

语法分析笔记（四）——LR(0) SLR LR(1) LALR lr slr lalr-CSDN 博客

正 则 表 达 式 转 NFA ， DFA, 最 小 化 DFA 将 正 则 表 达 式 ((+1-)?dd*)(.dd*)?(ele)((+1-)?dd*))?转换成 nfa-CSDN 博客

六、项目自评

项目任务一

80 分。

系统可以完成实验要求的必做任务中的全部要求，规定时间内提交系统和文档及讲解视频。系统编程风格好，设计思路基本清晰，界面简洁，便于使用。

文档中详细描述了系统设计思路，使用流程图清晰简洁的描述了实验项目中每一项功能的设计思路和实现的方法。

视频中根据系统和文档中的内容对整个项目进行了讲解。

项目任务二

75 分。

系统可以完成实验要求的必做任务中的全部要求，规定时间内提交系统和文档及讲解视频。系统编程风格好，设计思路基本清晰，界面简洁，便于使用。

文档中描述了系统设计思路，使用语言清晰简洁的描述了实验项目中每一项功能的设计思路和实现的方法，其中代码和实验项目一相比较多。

视频中根据系统和文档中的内容对整个项目进行了讲解。

项目任务三

80 分

系统可以完成实验要求的必做任务中的全部要求，规定时间内提交系统和文档及讲解视频。系统编程风格好，设计思路基本清晰，界面简洁，便于使用。

华 南 师 范 大 学 实 验 报 告

学生姓名 秦紫茉 学　号 20212121030

专　业 计算机科学与技术（师范） 年级、班级 2021 级计师 1 班

课程名称 编译原理项目 实验项目 词法分析程序、语法分析程序

实验时间 2024 年 5 月 1 日

实验指导老师 黄煜廉 实验评分

文档中详细描述了系统设计思路，使用流程图清晰简洁的描述了实验项目中每一项功能的设计思路和实现的方法。 视频中对实验要求的 mini—c 语法的相关要求进行了详细测试。