



南开大学  
Nankai University

南 开 大 学

计 算 机 学 院

编译系统原理实验报告

---

## 实现词法分析器构造算法

---

张刘明 2110049

年级：2021 级

专业：信息安全-法学

指导教师：王刚

2023 年 11 月 5 日

# 目录

<b>一、 上下文无关文法</b>	<b>1</b>
(一) SysY 语言特性	1
(二) CFG 描述 SysY 语言特性	1
1. 关键字	1
2. 变量	1
3. 常量	1
4. 运算符和表达式	1
5. 语句	1
6. 函数	1
(三) 形式化定义	1
1. 变量声明	1
2. 常量声明	1
3. 表达式	1
4. 赋值表达式	1
5. 逻辑表达式	1
6. 关系表达式	1
7. 算数表达式	1
8. 函数	1
9. 系统操作	1
10. 循环语句	1
11. 分支语句	1
12. 条件语句	1
<b>二、 正则表达式→NFA</b>	<b>2</b>
(一) 正则表达式构造 CFG	2
(二) 构造数据结构	2
1. 状态	2
2. 转换	2
3. NFA 结构	2
4. 构造操作	2
5. 解析和构建	2
6. 扩展性	2
7. 内存管理	2
(三) Thompson 算法的详细思路	2
1. 基本 NFA 片段的构造	2
2. 连接操作	2
3. 选择操作	2
4. 闭包操作	2
5. 算法过程	2
6. 算法的优缺点	2
(四) 编程实现	2
1. 添加转换	2
2. 构造 NFA	2

3.	NFA 的操作 . . . . .	2
4.	Thompson 算法的构造 . . . . .	2
5.	主函数 . . . . .	2
<b>三、 NFA-&gt;DFA</b>		<b>3</b>
(一)	从 NFA 到 DFA 的子集构造法的算法思路 . . . . .	3
1.	计算 e-closure(T) 的算法 . . . . .	3
2.	NFA 到 DFA 的转换 . . . . .	3
(二)	子集构造法的算法实现 . . . . .	3
1.	使用的数据结构 . . . . .	3
2.	功能 . . . . .	3
<b>四、 DFA 最小化</b>		<b>4</b>
(一)	DFA 最小化的实现思路 . . . . .	4
(二)	代码实现 . . . . .	4
(三)	数据结构 . . . . .	4
1.	DFA 结构 . . . . .	4
2.	split 函数 . . . . .	4
3.	minimize 函数 . . . . .	4
<b>五、 正则表达式-&gt;DFA</b>		<b>5</b>
(一)	直接进行正则表达式到 DFA 的转换 . . . . .	5
(二)	解析组合子的构建 . . . . .	5
(三)	代码实现 . . . . .	5
(四)	算法的效率分析 . . . . .	5
1.	算法的准确性 . . . . .	5
2.	Thompson 算法的效率 . . . . .	5
3.	子集构造法的效率 . . . . .	5
4.	DFA 最小化的效率 . . . . .	5
5.	整体性能考量 . . . . .	5
<b>六、 总结</b>		<b>6</b>

## 一、 上下文无关文法

### (一) SysY 语言特性

### (二) CFG 描述 SysY 语言特性

1. 关键字
  2. 变量
  3. 常量
  4. 运算符和表达式
  5. 语句
  6. 函数
- 代码

逐列访问平凡算法

### (三) 形式化定义

1. 变量声明
2. 常量声明
3. 表达式
4. 赋值表达式
5. 逻辑表达式
6. 关系表达式
7. 算数表达式
8. 函数
9. 系统操作
10. 循环语句
11. 分支语句
12. 条件语句

## 二、 正则表达式->NFA

### (一) 正则表达式构造 CFG

### (二) 构造数据结构

1. 状态
2. 转换
3. NFA 结构
4. 构造操作
5. 解析和构建
6. 扩展性
7. 内存管理

### (三) Thompson 算法的详细思路

1. 基本 NFA 片段的构造
2. 连接操作
3. 选择操作
4. 闭包操作
5. 算法过程
6. 算法的优缺点

### (四) 编程实现

1. 添加转换
2. 构造 NFA
3. NFA 的操作
4. Thompson 算法的构造
5. 主函数

### 三、 NFA→DFA

#### (一) 从 NFA 到 DFA 的子集构造法的算法思路

1. 计算  $e\text{-closure}(T)$  的算法
2. NFA 到 DFA 的转换

#### (二) 子集构造法的算法实现

1. 使用的数据结构
2. 功能

NIKU

## 四、 DFA 最小化

(一) DFA 最小化的实现思路

(二) 代码实现

(三) 数据结构

1. DFA 结构

2. split 函数

3. minimize 函数

NIKU

## 五、 正则表达式->DFA

(一) 直接进行正则表达式到 DFA 的转换

(二) 解析组合子的构建

(三) 代码实现

(四) 算法的效率分析

1. 算法的准确性
2. Thompson 算法的效率
3. 子集构造法的效率
4. DFA 最小化的效率
5. 整体性能考量

NIJUB



## 六、 总结

本次实验初步了解了编译器的工作原理, 了解了 LLVM 的工作原理, 并可以自主使用 LLVM IR 进行自己的编程, 为后续的编译器实验和操作打下了坚实基础。

实验分工:

艾明旭, 阶乘的实现以及相关内容的研究以及 LLVM 代码的研究及的编写

张刘明: 斐波那契的实现和相关内容研究以及相关原理的探讨

NIKU

## 参考文献

- [1] 杨侯哲, 李煦阳, 杨科迪, 费迪, 周辰霏, 谢子涵, and 杨科迪. 编译器开发环境部署. 2023.
- [2] 杨侯哲, 李煦阳, 孙一丁, 李世阳, 杨科迪, 周辰霏, 尧泽斌, 时浩铭, 贺祎昕, 张书睿. 预备工作 1——了解编译器及 llvm ir 编程. 2023.

NIKU