

编译原理 – 词法分析程序实验报告

61520324-许睿

日期: November 14, 2022

1 实验目的

2 实验过程

2.1 miniC 的文法定义

本实验选择C语言作为实验测试语言, 选择的miniC语言包括1中所示的保留字

表 1: miniC中使用的保留字

reseved words	description
int	miniC的基本数据类型
void	main函数的返回值类型
if-else	条件分支语句
while	循环分支语句
return	用于函数返回

此外该实验定义的miniC还支持两个无符号整型数的+, -, *, /二元运算, 两个无符号整型数的<, >, ==, <=, >=二元关系运算, 变量的声明和定义, 函数的声明和定义, 注释的忽略. 下面给出miniC的文法定义规则:

2.1.1 标识符的正规文法, 正规式, NFA 和 DFA

正规文法和正规式 标识符要求开头字符为英文字母的大小写或下划线, 不能以数字开头, 中间可以出现任意长度的单词或数字或下划线.

```
identifier = letter (letter | digit)*  
letter -> [a-z] | [A-Z] | _  
digit -> [0-9]
```

正规式转化为 NFA 根据以上正规文法的定义, 可以经过如1转化得到 NFA:

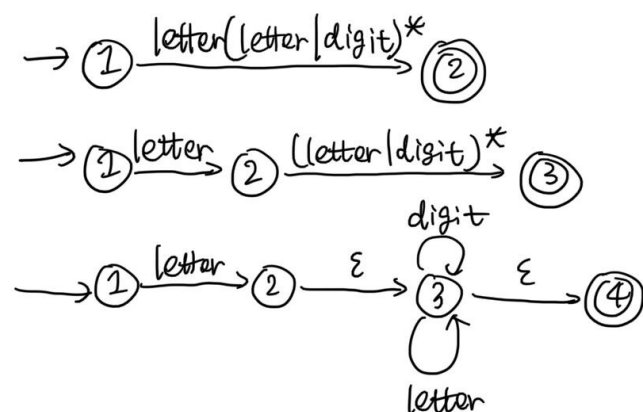


图 1: 标识符的非确定有限状态机推导过程

NFA 转化为 DFA 将最后得到的 NFA 确定化: 首先求 DFA 中的相应状态如表2

表 2: DFA 中相应状态转化表

	letter	digit
S0=(1)	(2,3,4)	-
S1=(2,3,4)	(3,4)	(3,4)
S2=(3,4)	(3,4)	(3,4)

根据状态转换表得到相应的 DFA 如2所示

图 2: 标识符对应的 DFA

该 DFA 可以经过最小化后得到3所示的最小化 DFA

图 3: 标识符对应的最小化 DFA

2.1.2 无符号数常量的正规文法, 正规式, NFA 和 DFA

正规文法和正规式

```
digits -> digit digit*
```

正规式转化为 NFA

NFA 转化为 DFA

2.1.3 四则运算的正规文法, 正规式, NFA 和 DFA

正规文法和正规式

```
# 正规文法
expr -> expr (+ | -) term | term
term -> term (* | /) factor | factor
factor -> (expr) | digits | identifier
```

```
# 正规式
```

正规式转化为 NFA

NFA 转化为 DFA

2.1.4 关系运算的正规文法, 正规式, NFA 和 DFA

正规文法和正规式

```
uneqExpr -> expr | uneqExpr < assignExpr | uneqExpr > assignExpr  
eqExpr -> expr | uneqExpr = assignExpr | uneqExpr ! assignExpr  
assignExpr -> expr | = expr
```

正规式转化为 NFA

NFA 转化为 DFA

2.1.5 /**/型注释的正规文法, 正规式, NFA 和 DFA

正规文法和正规式

```
cmt -> / cmtStart  
cmtStart -> \* doc  
doc -> [^\*]* doc | \* cmtEnd  
cmtEnd -> \* cmtEnd | [^\*/] doc | /
```

正规式转化为 NFA

NFA 转化为 DFA

3 词法分析程序

4 测试

5 实验总结