

编译原理实验 1 词法分析器

19335015 陈恩婷

1. 实验描述

手动设计实现，或者使用 Lex 实现词法分析器。

2. 实验原理

词法分析器的功能是输入源程序，输出单词符号。单词符号是一个程序语言的基本语法符号。程序语言的单词符号一般可以分为关键字、标识符、常数、运算符、界符等。

单词种别通常用整数编码。

3. 词法分析器设计

(1) 支持的单词范围

支持的单词包括以下几种：

1. 标识符 (identifier), 编号 <00>
2. 字符 (character), 编号 <01>
3. 字符串 (string), 编号 <02>
4. 关键字 (keyword), 每个关键字一个编号
5. 界符 (delimiters), 每个界符一个编号

词法分析器所支持的所有关键字和界符如下所示：

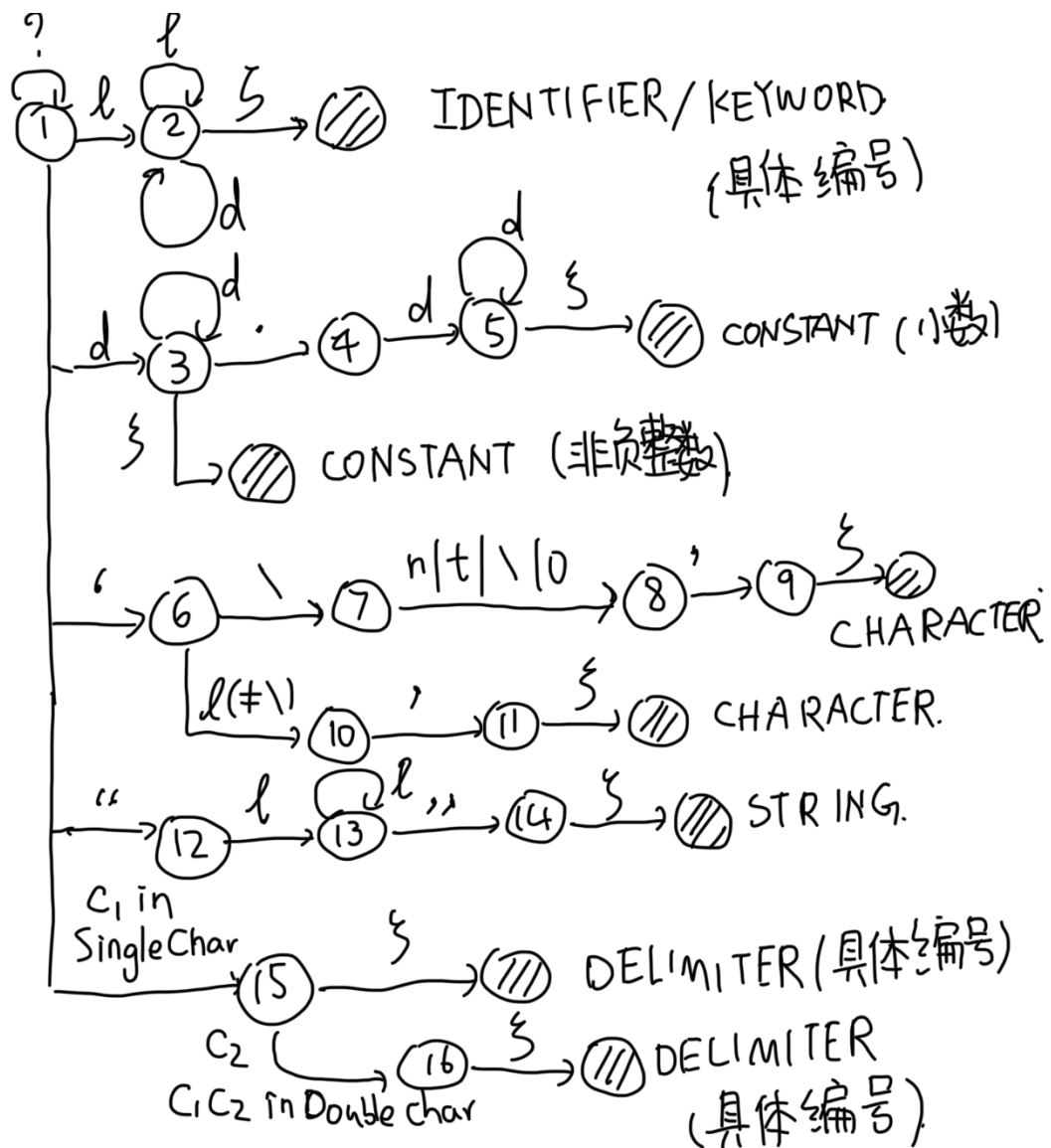
```

const vector<string> Codes = {"IDENTIFIER", "CHARACTER", "STRING", "CONSTANT"};
const vector<string> Keywords
    = {"auto", "break", "case", "char", "const", "continue", "default", "do",
       "double", "else", "enum", "extern", "float", "for", "goto", "if", "int",
       "long", "register", "return", "short", "signed", "sizeof", "static", "struct",
       "switch", "typedef", "union", "unsigned", "void", "volatile", "while"};
const vector<char> SingleChar = {'=', '+', '-', '*', '/', '%', '>', '<', '(', ')',
                                  '{', '}', ';', ':', '.', '&', '|'};
const vector<string> DoubleChar = {"==", ">=", "<=", "+=", "-=", "/=", "*=", "++",
                                   "--", "&&", "||"};

```

(2) 自动机的设计

根据自动机的相关知识，设计自动机如下：



4. 实验结果

先编译程序: g++ source.cpp -o clang.exe

例 1

输入程序 demo.c:

```
int main(){
    int a; int b = 12;
    char d = 'p';
    double c = a*b;
    c += b;
    printf("Hello World!");
}
```

运行命令: clang.exe demo.c tokens.txt

输出结果 tokens.txt:

```
int <20>
main <22>
( <45>
) <46>
{ <47>
int <20>
a <00>
; <49>
int <20>
b <00>
= <37>
12 <03>
; <49>
char <07>
d <00>
= <37>
'p' <01>
; <49>
double <12>
c <00>
= <37>
a <00>
* <40>
```

```

b <00>
; <49>
c <00>
+= <64>
b <00>
; <49>
printf <00>
( <45>
"Hello World!" <02>
) <46>
; <49>
} <48>

```

例 2

输入程序 demo.c:

```

int main (void)
{
    int    a=1, d=2, c;
    if(a<=d)
    {
        c=a;
        a=d;
        d=c;
    }
    char    ch[10]="ok";
    char    x, y='a';
    c=a + d;
}

```

运行命令: clang.exe demo.c tokens.txt

输出结果 tokens.txt:

```

int <20>
main <22>
( <45>
void <34>
) <46>
{ <47>
int <20>
a <00>
= <37>

```

```
1 <03>
, <50>
d <00>
= <37>
2 <03>
, <50>
c <00>
; <49>
if <19>
( <45>
a <00>
<= <63>
d <00>
) <46>
{ <47>
c <00>
= <37>
a <00>
; <49>
a <00>
= <37>
d <00>
; <49>
d <00>
= <37>
c <00>
; <49>
} <48>
char <07>
ch <00>
[ <59>
10 <03>
] <60>
= <37>
"ok" <02>
; <49>
char <07>
x <00>
, <50>
y <00>
= <37>
'a' <01>
; <49>
c <00>
```

```
= <37>
a <00>
+ <38>
d <00>
; <49>
} <48>
```

如上所示，可见词法分析器成功地完成了任务。

5.实验总结

本次实验通过自己实现一个词法分析器，复习了理论课上讲到的形式语言基础、自动机基础、词法分析等内容，让我对这些部分有了更深刻的认识。

实验中我采用的是 C++ 语言来实现词法分析器程序，一开始实现时有考虑过 MATLAB 等其他语言，不过在简单的尝试过程中意识到 C++ 对字符和字符串的处理会更有优势（有更多方便使用的函数），而且 C++ 一般来说会比 MATLAB 更快，所以最终选用了 C++ 语言。

这次实验让我受益匪浅，希望再接再厉，学好编译原理的知识。