编译原理实验报告：词法分析器

15331416 赵寒旭 数字媒体技术

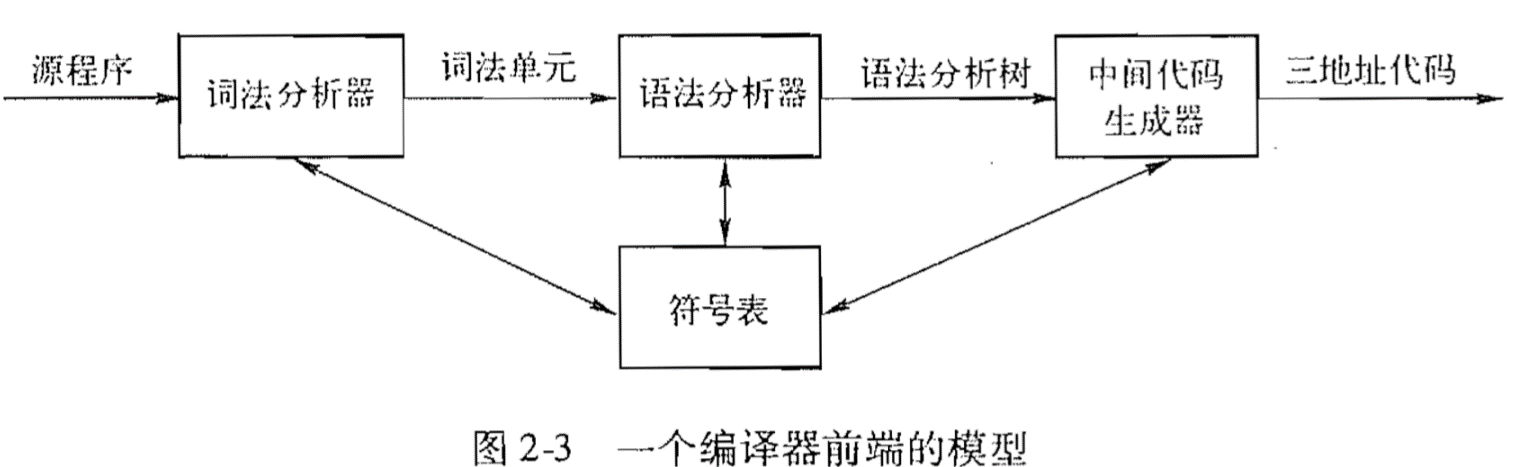
1. 实验目的

用lex实现一个词法分析器

2. 实验思路

2.1 词法分析器的作用

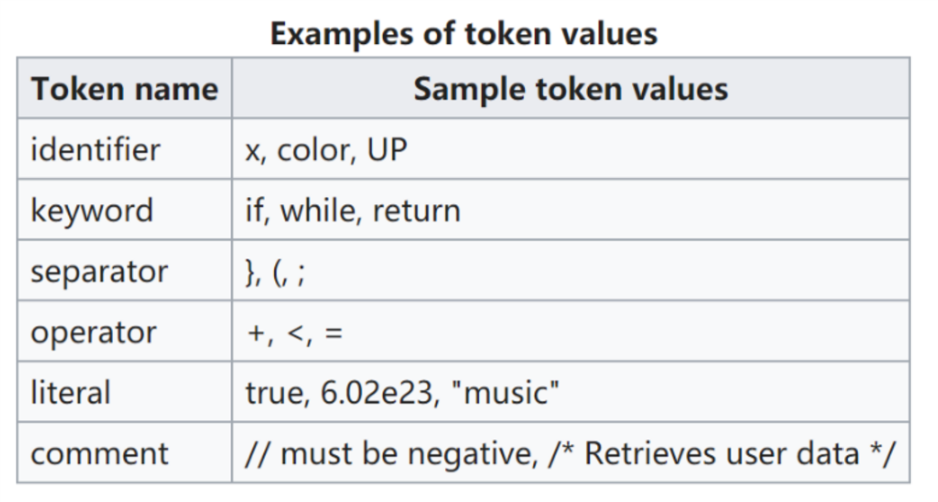
词法分析是编译的第一阶段，主要任务是读入源程序的输入字符，将它们组成词素，生成并输出一个词法单元序列，每个词法单元对应一个词素。本次实验中词法分析程序使用lex自动生成。



2.2 词法单元类型

引自<https://en.wikipedia.org/wiki/Lexical_analysis#Token>

本文词法单元类型及内容参考下表：



identifier（标识符）：由程序员选择的名称

keyword（关键字）：已由编程语言确定的名称

separator（分隔符）：标点符号和成对的分隔符号

operator（操作符）：操作参数产生结果

literal（文字量）：数字和文本

comment（注释）：包括行注释和块注释

2.3 词法分析器生成工具Lex

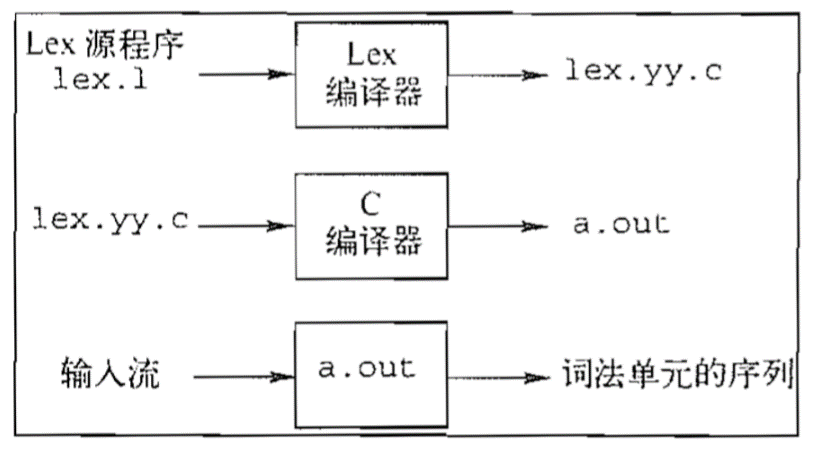
Lex支持使用正则表达式来描述各个词法单元的模式，由此给出一个词法分析器的规约。

（1）用Lex创建一个词法分析器的过程

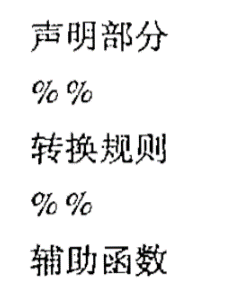
1) 用Lex语言写一个输入文件lex.l，描述将要生成的词法分析器。

2）Lex编译器将lex.l转换成c语言程序lex.yy.c

3）lex.yy.c被c编译器便以为一个.exe文件，得到一个读取输入字符流并生成词法单元流的可运行的词法分析器。



（2）Lex程序的结构



1）声明部分

首先可有起始于"%{"符号，终止于"%}"符号包括include语句、声明语句在内的C语句。

包括变量和明示常量，可以声明词法单元的名字或正则定义。

2）转换规则

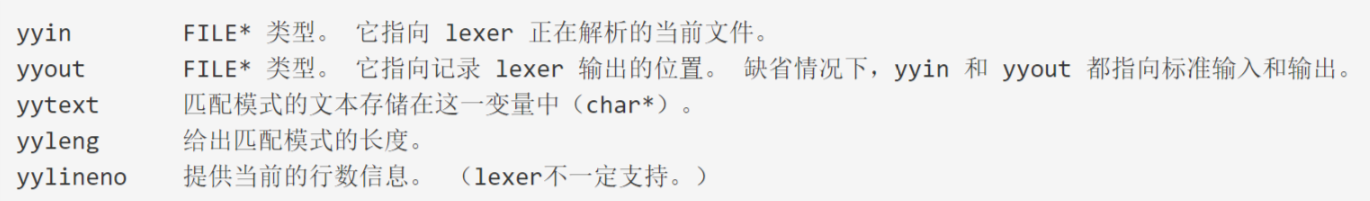
形式：模式 { 动作 }

每个模式是一个正则表达式，可以使用声明部分中定义的变量。

3）辅助函数

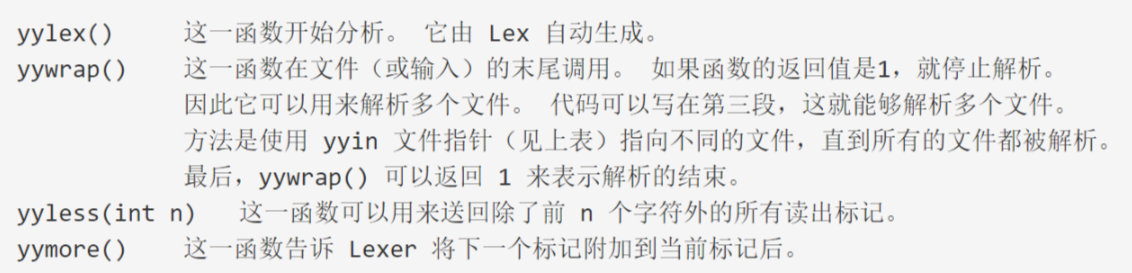
包含各个动作需要使用的所有辅助函数。

2.4 Lex变量



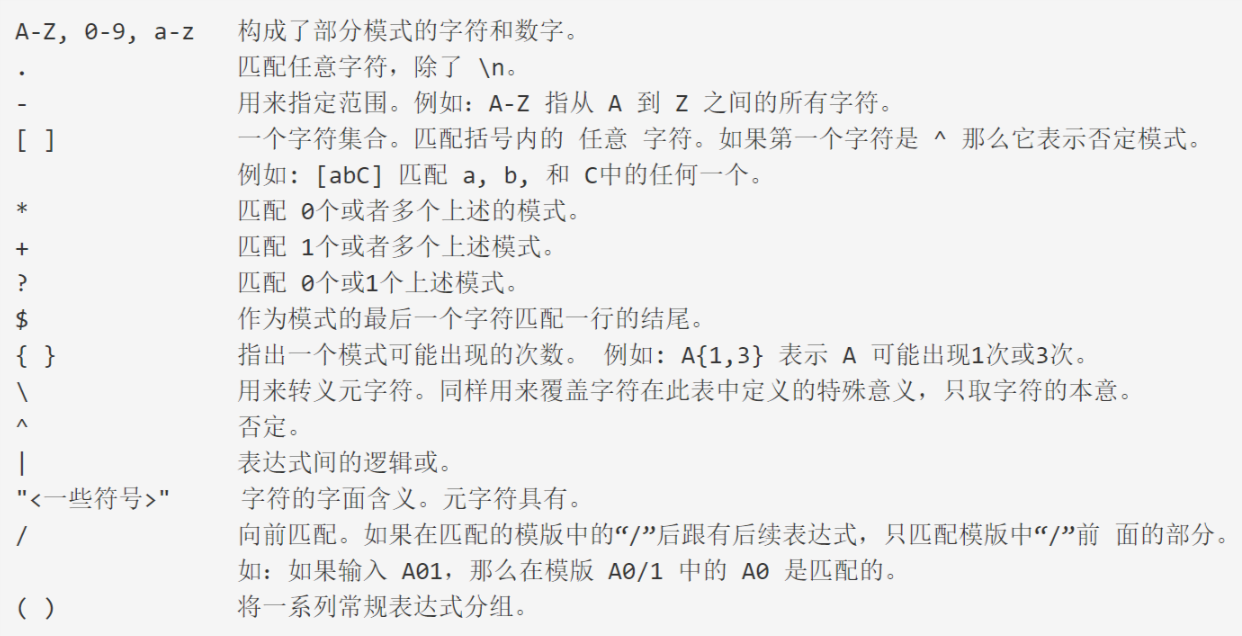
实验中使用yyin指向待解析代码文件，用yytext获得模式匹配文本

2.5 Lex函数



实验中使用yylex()函数分析代码文本，使yywrap()返回值为1解析完毕停止解析。

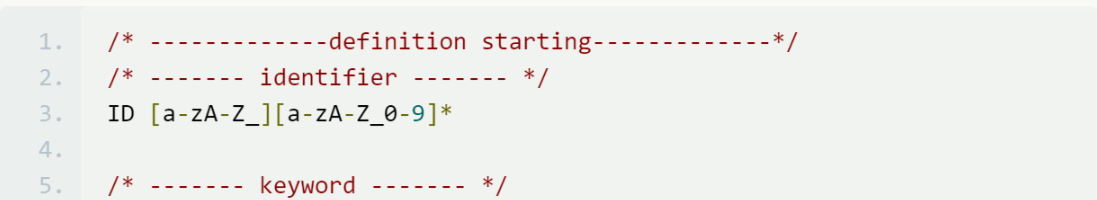
2.6 正则表达式相关字符及其含义列表



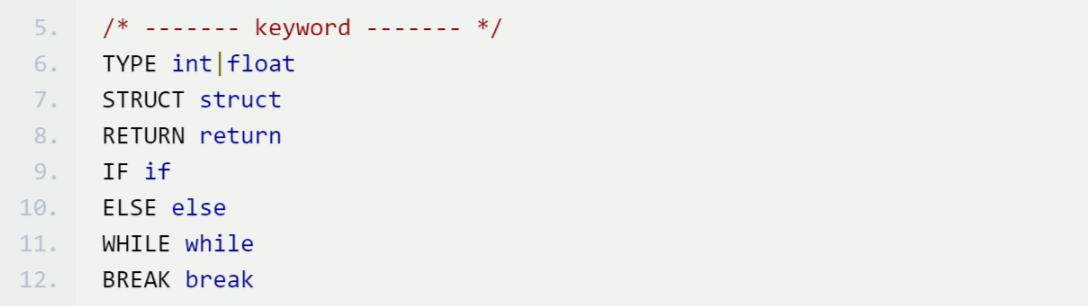
3. 实验步骤

3.1 定义需要匹配的具体词法单元对象

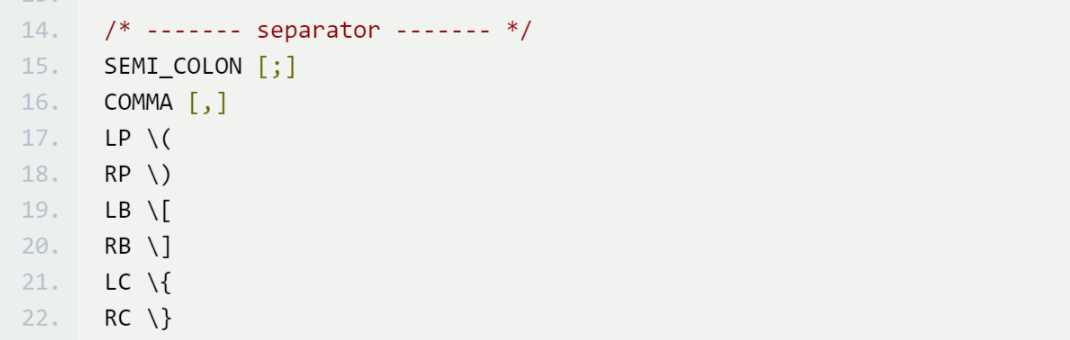
（1）标识符



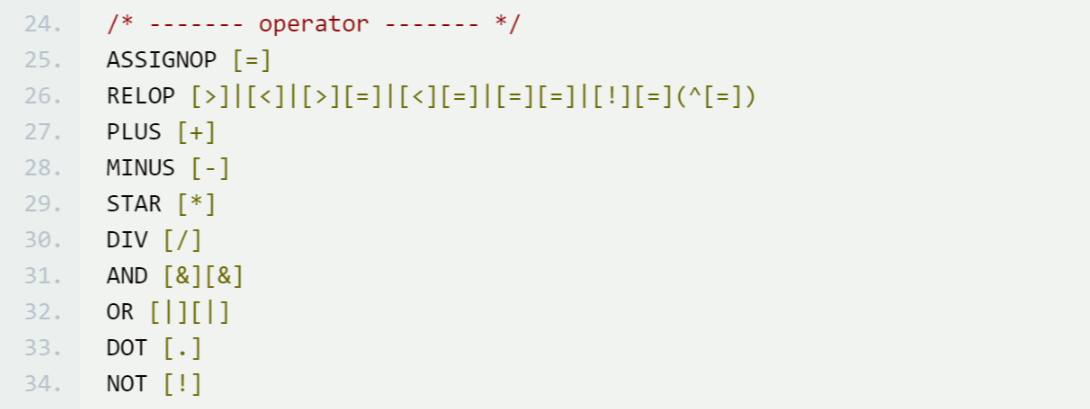
（2）关键字



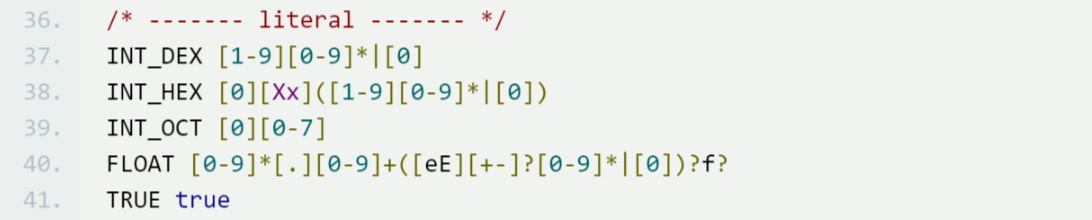
（3）分隔符



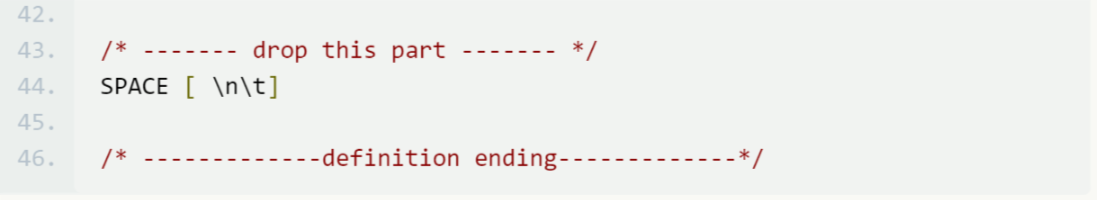
（4）操作符



（5）文字量



（6）空白换行及制表符（直接跳过）



3.2 匹配后操作的实现

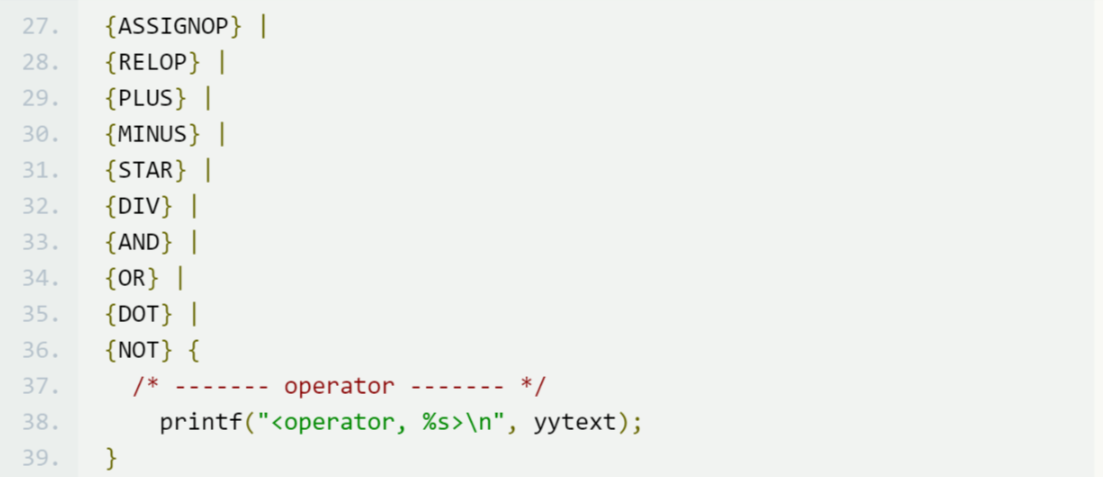
（1）关键字



（2）分隔符



（3）操作符



（4）文字量



（5）标识符

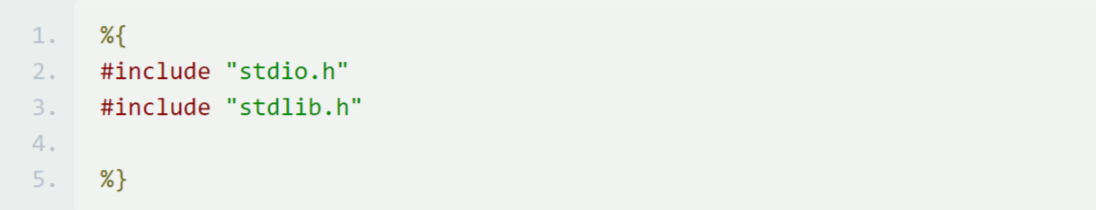


（6）空白，换行及制表符



3.3 c语言辅助部分

（1）头文件包含



（2）yywrap返回值设置

当词法分析程序遇到文件结尾时，它调用例程yywrap()来找出下一步要做什么，如果yywrap()返回0，则扫描程序就继续扫描，如果返回1，则扫描程序就返回报告文件结尾的零标记。

lex库中的yywrap()的标准版本总是返回1，但是可以用自己的值来替代它，如果yywrap()返回指示有更多输入的0，那么它首先需要调整指向新文件的yyin，可能需要使用fopen()。

本次实验设置返回值为1，仅读取一个文件。



（3）main函数



yyin指向待解析代码文件。

由lex创建的扫描程序有入口点yylex()。调用yylex()启动或重新开始扫描。如果lex动作执行将数值传递给调用程序的return，那么yylex()的下次调用就从它停止的地方继续。

4. 实验结果

4.1 运行步骤

4.2 运行结果

5. 遇到的问题

Lex中的冲突解决

选择在Lex程序中先被列出的模式，id应该放在后面！