词法分析

上海交通大学

一、 目标

本次作业的目标为利用 flex 编写一个词法分析器，实现对 Seal++语言的词法提取。

二、 说明

本次作业主要利用 flex，定义 Seal++中的词法格式，从而分析 Seal++文件中的符号，将其转化为合适的 Seal 输出文件，用于进一步的语法分析。本次实验会牵涉到 flex、正则表达式[1]等内容，对于 flex 将有相关参考资料。

三、 flex 简介

flex 是 lex 的一个开源实现（见课本附录 A），它能够根据用户定义的正则表达式，对输入文件中的字符串进行匹配，并且对匹配的结果做出相应的处理。这也是词法分析器的基本功能。

flex 能够将用户编写的规则文件编译为 C 源代码（C 与 C++语法相似，故可以以 C++形式继续开发），而编译后的文件可以直接作为库引用。而库代码往往非常繁杂，但是此次作业重点不在编程，故在此次代码设计任务中，其他部分都已经给出，同学只需要编写相应的 flex 规则文件，也即词法构成规则即可。

flex 规则文件的基本结构如下

```
%{
声明（Declarations）
%}
定义（Definitions）
%%
规则（Rules）
%%
处理函数（User subroutines）
```

声明和处理函数部分是可选的，可以在其中编写一些辅助用的函数。定义部分也是可选的，但是通常对正则表达式编写会非常有用，例如定义

```
LOWERCASE    [a-z]
```

简明的定义了小写字母。flex 中使用的常见的正则表达式可以查阅课本附录 A，在此不再赘述。举例如[xy]，表示字符 x 或者 y。

在 flex 中最重要的是规则部分，例如

```
[0-9]+ {// 处理函数}
```

将会对匹配到的符合[0-9]+的字符串（数字串）做对应的处理函数动作。注意，正则表达式以最长匹配原则，也即如果有[0-9]+和[0-9a-z]+两个正则式，则对于 2a 这个串，将按照第二个[0-9a-z]+匹配，而不按照一个[0-9]+一个[0-9a-z]+匹配，因为前一种匹配方式更长。

[1] 可参考 https://www.runoob.com/regexp/regexp-syntax.html

四、 文件说明

- Little_lexer.l

flex 的规则文件，也即此次作业要编写的文件，目前为空白。

- 测试文件和测试答案

在 sealpps 文件夹里给了 5 个样例文件，在 test_results 文件夹中有对应的 5 个词法分析结果文件。可以利用自己编写的词法分析器预先分析，然后与结果对比，结果要求与样例答案完全一致。

- 需要分析的符号等

TYPEID = "Int","Float","String","Bool","Void"

Symbol={"%","~","^","-","*","/","+","-",";",",","=",":","|","&","{","}","(",")",",","<",">","%f","%s" ,"%lld","|","&","||","&&"}

'#'为注释符号，其后的数值皆不分析

KEYWORD= "fprintf","while","aafor","if","else","continue","break","return"

五、 如何测试

首先，在测试之前，请确保机器上安装了 flex，具体的，ubuntu 机器请运行 sudo apt install flex bison 来安装。

使用./do.sh 进行编译

运行 python test.py 来测试。

当然也可以单个测试，./little_lexer < ./sealpps/1.sealpp

六、 处理结果要求

此次作业要求提交的代码满足如下 1 个条件：

1. 输出结果

要求将输入文件代码的每个识别出的单词符号，一行一个输出，格式为

#<单词符号出现行号>　<类型>　[值]

要求将每个符号都定义相关规则。例如，如果第三行匹配到了一个单词符号为 BOOL_CONST 类型，且值为 true，则对应行的输出为#3 CONST 1。将所有数值类型的值输出都为#num_line CONST value。又例如，在第一行匹配到一个单词符号左花括号{，由于其没有值，故输出为#1 {。

七、 如何评分

作业提交之后，我们将根据同学提交的词法分析器对若干个已有的样本进行分析（与给出的实例样本不完全相同），其中样本可能包含词法错误，包含错误的样本要求输出错误及其行，给出错误原因和出错行为通过，无错误样本要求将输出结果与标准结果比对，完全一致的通过，否则不通过。按照所有样本分析通过率给分（即例如满分 10 分，通过率 0%，给 0 分，通过率 70%，给 7 分）。

八、 文件提交要求

仅提交所有源代码文件，最重要的是那个名叫 little_lexer.l 的文件，如果同学希望自己用很多别的.h 文件或者库，那就需要自己改 do.sh 中编译的语句。最终在助教端测试的时候，助教会直接运行 do.sh 来编译得到 little_lexer 的二进制文件并用其进行测试。

放置在一个名为<学号>的目录下，并且将整个目录打包为<学号>_<姓名>.tar 格式。

九、 提交截止时间

请同学们在 2021.11.14 的 0:00（2020.11.13 的 24:00）之前，将结果提交到 canvas（迟交结果按 0 分处理）。