算符优先文法分析器设计报告

杨涵章 518030910022

linqinluli@sjtu.edu.cn

一、 项目要求

从文本文件读入一个上下文无关的算符文法，构造算符优先分析表并以文本文件的形式输出表格。

二、 设计过程

1、 基本数据结构：

非终结符结构体：包含非终结符符号字符串、对应的 firstvt 字符串集合、对应的 lastvt 字符串集合。以及对应的增加、删除 firstvt、lastvt 元素的成员函数。

终结符结构体：包含终结符符号字符串、对应的前继字符串集合、后继字符串集合。以及对应的增加、删除前继后继字符串集合元素的成员函数。

分析器类：主要的分析器类，用于储存非终结符集合和终结符集合结构体数组，以及错误信息等。其成员函数包含读入函数、计算 firstvt、lastvt 集合的函数、设置分析表函数、输出函数以及各部分函数所需功能函数。

2、 算符文法的读入：

测试文法中所给出的算符文法的形式较多，需要处理各种情况。初步思路为首先读入全部语句，读取过程中得到非终结符集合，用于后续的处理。然后对于使用 '|' 符号分隔的语句，使用 strtok 函数进行处理，将

输入的测试文法分割成单独子句进行处理。对于每一个子句，同样使用 strtok 函数进行处理，以‘ ’作为分隔符分离出各个字符，依次对各个字符进行处理。对于一个子句，用三个指针对应连续的三个字符，滚动前进，依次设置当前非终结符的 firstvt 集合、lastvt 集合，以及各个终结符的前继和后继集合，最终将输入的算符文法信息储存到分析器类中，同时对应各种错误设置错误信息。同时在滚动过程中，设置分析表中的‘=‘，即两个终结符中为一个非终结符。

3、 firstvt 集合、lastvt 集合构造：

遍历所有的非终结符进行处理：首先遍历该非终结符的 firstvt 集合、lastvt 集合，删除所有与该非终结符相同的字符，然后重新遍历 firstvt 集合、lastvt 集合：遇到非终结符，便将对应非终结符号的 firstvt 集合、lastvt 集合中所有元素加入到当前非终结符的 firstvt 集合、lastvt 集合中；遇到终结符不进行处理。

遍历完所有的非终结符后判断所有的非终结符的 firstvt 集合、lastvt 集合是否包含非终结符：包含则再次遍历直到没有；不包含则完成 firstvt 集合、lastvt 集合的构造。

4、 分析表设置：

首先将‘$’加入终结符集合，并设置它的前继和后继为开始非终结符。

遍历所有的终结符：遍历该终结符的前继，对于每一个前继，前继的 lastvt 集合中的终结符‘>‘该终结符，对于每一个后继，该终结符’< ‘后继的 firstvt 集合中每一个终结符。

5、 输出分析表：

首先判断错误信息，如果错误直接输出对应错误信息到命令行，不再输出分析表。如果输入正确并为算符优先文法，则按照分析表对应输出完整表格，未包含信息仅输出空格。

三、 测试文法：

大作业要求仅给出两种文法，我设计过程中基于这两种文法设计以下几种测试文法。

Test 1: 即 ppt 给出的测试样例 1，经过运行，得到分析表，与 ppt 中分析表进行比对， 结果正确，并能输出格式良好的文本文档（“out.txt”）。

E -> E + T | T

T -> T * F | F

F -> (E) | i

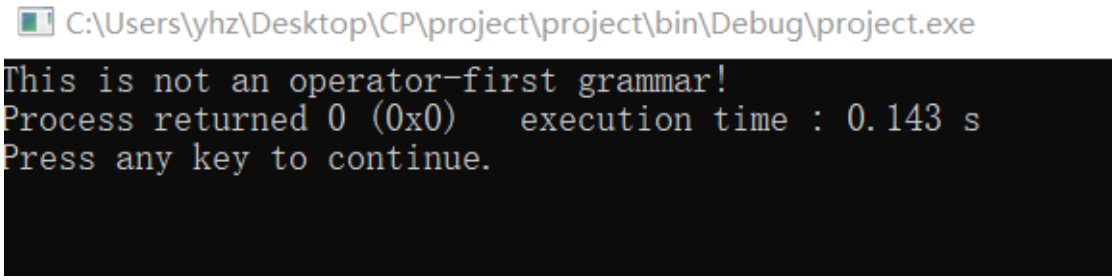
输出：

	+	*	(	)	i	$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
$	<	<	<		<	=

Test 2: 即 ppt 给出的测试样例 2，经过运行，发现对于表中同一元素有两种情况，具有二义性，得到结论，该文法不是算符优先文法，并将错误信息输出到终端。

E -> E + E | E * E | (E) | id

输出：



Test 3: 将测试样例 1 中的单个字符替换为随机单词，得到分析表，与测试样例 1 中分析表进行比对， 结果正确，并能输出格式良好的文本文档（“out.txt”）。

Es -> Es + Ttdsa | Ttdsa

Ttdsa -> Ttdsa * Ffff | Ffff

Ffff -> (Es) | i

输出：

	+	*	(	)	i	$
+	>	<	<	>	<	>
*	>	>	<	>	<	>
(	<	<	<	=	<	
)	>	>		>		>
i	>	>		>		>
$	<	<	<		<	=

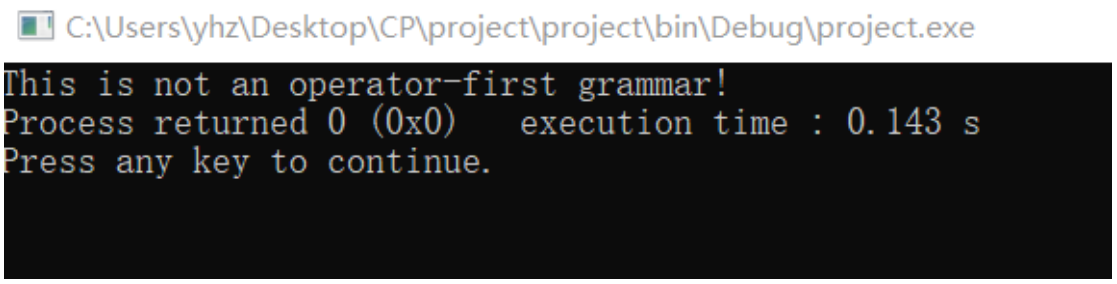
Test 4: 测试文法中存在连续两个非终结符，不属于字符优先文法，得到结论，该文法 不是算符优先文法，并将错误信息输出到终端。

E -> E + T E | T

T -> T * F | F

F -> (E) | i

输出：



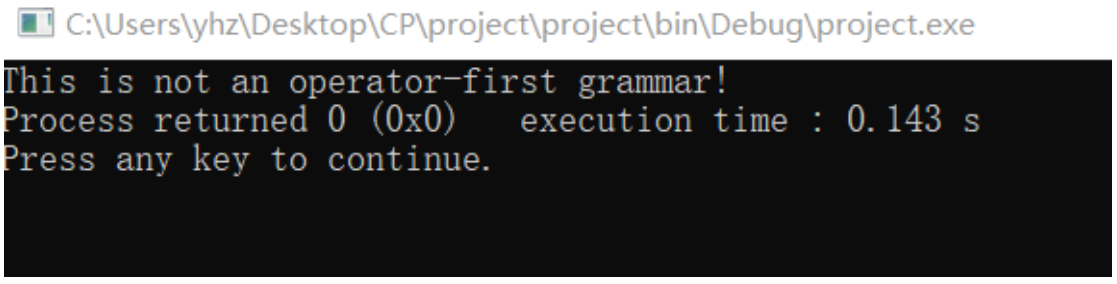
Test 5: 测试文法中包含 ε 符号，得到结论，该文法 不是算符优先文法，并将错误信息输出到终端。

E -> E + T | T

T -> T * F | F

F -> (E) | ε

输出：



四、 测试说明

运行环境：

该算符优先文法分析器使用程序语言为 c++，编译器为 GUN GCC，对应信息如下：

```
Have g++ follow the coming C++0x (aka c++11) ISO C++ language standard  [-std=c++0x]
```

测试方法：

保证 main.cpp 文件和 5 个 test 文件，1 个 out.txt 文件在同一目录，编译运行 main.cpp，如需修改测试对象，修改对应的测试文件名宏定义即可：

#define TESTFILE "test1.txt"

即将“test1.txt”更改为“test2.txt”或其他测试文件。如果是算符优先文法，输出结果保存在 out.txt 文件中。