编译原理实验报告

顾田 2021300004042

1. 实验目的

实验的目的是使用 Bison 和 Flex 完成基本的逻辑判断语句布尔结果的输出。例如：

```
输入：3 <= 2 || 13 >= 19 && 1 < 2
输出：Output: FALSE, 3, 1
```

输出中，FALSE 是运行结果，第一个数字 3 是题目中判断的总次数，第二个数字 1 是因为短路规则而跳过的判断次数。

在 13>=19 && 1<2 中，根据短路规则，13>=19 的结果已经为否（FALSE），那么就不再需要做 1<2 的逻辑判断。因此短路跳过 1 次逻辑判断。

2. 文法和语法设计

2.1 词法分析

输入的符号有：==, <, !, >, <=, >=, &&, ||以及括号()。在 mytool.l 文件中，我们将这些符号解析为 token，例如：

```
"&&" {
    return AND;
}
```

2.2 语法分析

对输入的表达式 S，设计以下语法规则推导：

语法规则	说明
S -> E NEWLINE	每一个换行符是一个语句
E -> E AND E **| E OR E**	AND(&&)/OR(||)表达式的结果是两侧语句逻辑与的结果。
E -> NOT E	NOT(!)表达式的结果是表达式的否定。
E -> LPAREN E RPAREN	括号中的表达式结果
E -> R	如果没有 AND/NOT/OR 符号和括号，表达式的

	结果可以直接运算。
R -> VALUE LT VALUE **| ……** **| VALUE NOTEQ VALUE**	对单一表达式的结果直接求值。

2.3 运算优先级与结合性

实验中运用到了这样一段代码：

```
%left OR
%left AND
%nonassoc LT GT LTEQ GTEQ EQ NOTEQ
%nonassoc NOT
%left LPAREN RPAREN
```

其中，%left 表示左结合性，即如果两个运算符具有相同的优先级，左边的运算符先结合。%nonassoc 表示无结合性。

题目条件中，OR 和 AND 都是左结合，且 AND 的优先级高于 OR。括号也是左结合，且具有最高的优先级，括号内的表达式首先被求值。这个优先级和结合性定义确保了表达式按照预期进行解析和计算。

2.4 短路次数的计算

短路次数的计算比较麻烦。在

```
E -> E AND E | E OR E
```

这条规则中，E 并不一定只有一个表达式。例如：

```
3 >= 2 || 13 <= 19 && 1 < 2
```

这个表达式短路次数应为 2 次。但假如只在逻辑或判断的左边结果为 True 时给总短路次数加 1，将会得到错误的结果。

在处理这个问题时，我使用了一个结构体。这个结构体可以记录逻辑表达式的结果（int val），此外还可以记录表达式包含基本逻辑表达式的个数（int total）和表达式之前的短路次数。在语法树的顶端结点，可以直接用 int short 读出程序执行过程中的短路次数：

```
typedef struct
{
 int val;
 int total;
 int short1;
} myStruct;
```

例如，在 AND 逻辑的判断时，如果发生短路，在代码注释处我们做如下处理：

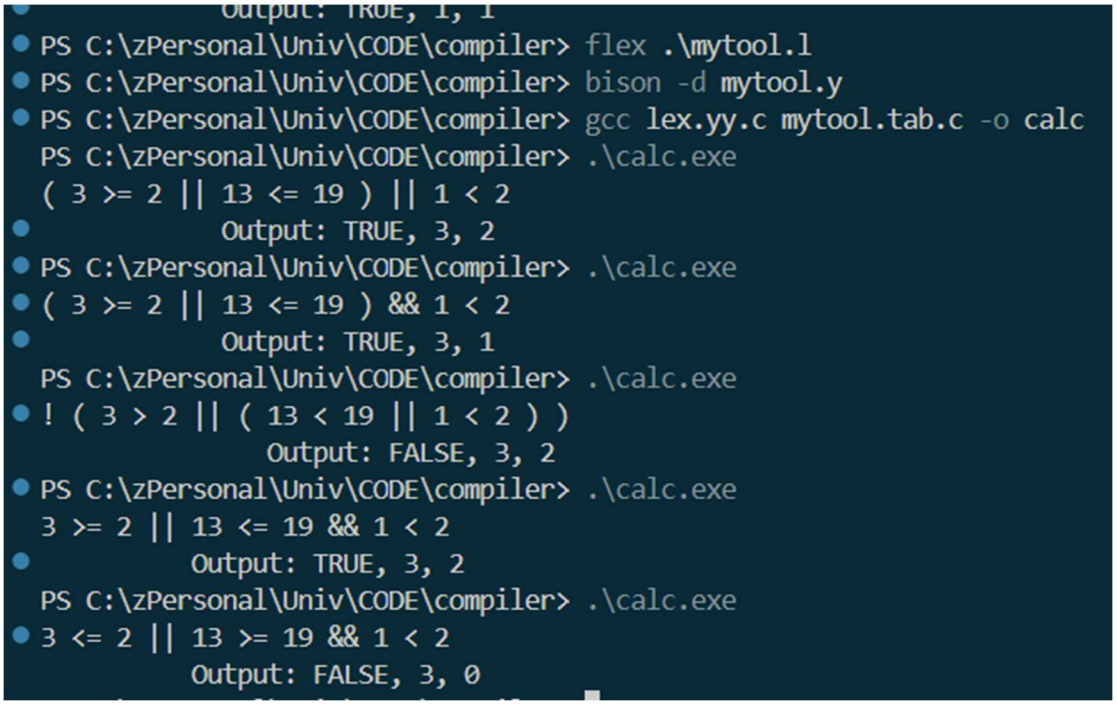
```
E AND E {
    if ($1.val) {
        $$.val = $3.val;
    } else {
        $$.val = 0;
       //短路次数 short1 增加右侧表达式的基本表达式总数
        $$.short1 += $3.total;
    }
   $$.total = $1.total + $3.total;
}
```

这样在程序运行结束后我们便可以正确获取短路规则跳过判断的总次数。

3. 实验结果

3.1 测试结果

本次实验完成了所有样例的测试，下面是部分样例执行结果：



3.2 实验总结

在本次实验中，通过使用 Bison 和 Flex 实现了对基本逻辑判断语句的解析与布尔结果的输出，成功地完成了实验的各项要求，得到了预期的结果。以下是实验中的一些关键点和总结：

初次接触 Bison 和 Flex 时，遇到了如何将词法分析生成的 token 正确传递给语法分析器的问题。通过阅读相关文档和示例代码，理解了词法分析器的规则与 Bison 的接口。

实现短路逻辑的跳过次数统计较为复杂。在实现过程中，使用了结构体记录表达式的值、总判断次数和短路次数，通过在语法规则中正确更新这些值，解决了短路逻辑的统计问题。

通过实验，我深入理解了词法分析和语法分析的原理和实现方法，掌握了使用 Flex 定义词法规则和 Bison 定义语法规则的基本技巧。