

实验二语法分析(Parser)报告

191850189王涛

1.实现的功能

本次实验的主要任务是

1.根据C++语法编写了parser的.g4文件

这一块是最重要的，需要对于参考文件中的一部分语法进行修改；

此外exp部分的优先级以及结合性要十分注意：

exp部分取负和取非的优先级是一样的，应该书写为<assoc=right> (MINUS | NOT) exp，如果书写为 (<assoc=right> MINUS exp | <assoc=right> NOT exp)是错误的，实验的时候发现这样写两者的优先级antlr应该不会识别为一样的，会导致在程序中别的地方全部正确的情况下，最后的四个测试用例分数都会比较低（会出现大量额外的输出，自己设置了一些用例暂时没有测出原因）。

也就是说表达式写在同一行然后再加括号并不能保证处理的时候将它们识别为优先级一样的。

2.使用visitor对语法树进行遍历

①在正确的情况下打印按要求打印

②在有语法错误时按要求打印报错信息

都是通过重载相应的对象方法实现的

3.实现勘误备选分支

在数组定义时，对中括号里是字母或者浮点数的情况进行特殊处理，本次实验并没有处理在数组访问时中括号里出现浮点数的情况（数组访问时，中括号里出现字母变量是正确的，不需要处理）；

```
a[3.5  
  
]
```

类似上面的情况，默认报错会报错在第二行，勘误的时候要将错误报在第一行

2.实验中应该注意的点

1.没有语法错误时，如果遇到空的语法单元不需要输出，输出的时候用System.err.out()

可以通过getChildren方法判断，如果返回的是0则不进行打印；此外应该还可以通过一个获得语法单元text的方法判断，这里选择的是第一种方法

2.每行只报一个错误

可以想到的处理方式是将上一次打印时输出的行号进行记录，下一次打印的时候与该记录进行比较，如果相等则不输出，但是这样的处理方式可能会出现一些问题，因为在一些文件的报错中（还没有实现勘误备选分支时测试的），报错的行号并不是递增的；

所以最后使用的是arraylist先存储，每次输出判断是不是已经输出过；

还可以的方法是开一个比较大的真值数组，初值全部设置为true，某一行输出了，对应数组相应的地方就设置为false，不过随着文件的增大，数组一开始开得也会比较大，是一种用空间换时间的方法

3.实验中很有用的工具(或者是方法)

①shift连续按两次查找方法

②右键点击某个语法单元选择Test Rule Program可以先输入语句进行测试，antlr会生成语法树，书写parser.g4文件的时候很有用