

实验内容

对照着 `my_parser.cpp` 中的 TODO 依次完成了

- `p_Type()` 的实现
- `p StmtList()` 中不同 First 对应的 action
- `p_Statement()` 中对 `If / LBRACE / SEMICOLON` 的解析
- `p_VarDecl()` 的实现
- `p_Return()` 的实现
- `p_If()` 的实现
- `p_Expression()` 的实现
- `p_Assign()` 的实现
- `p_LogicAnd()` 的实现
- `p_Relational()` 的实现

思考题

1. 在框架里我们使用 EBNF 处理了 `additive` 的产生式。请使用课上学习的消除左递归、消除左公因子的方法，将其转换为不含左递归的 LL(1) 文法。

```
additive : multiplicative TEMP2  
  
TEMP1 : '+' multiplicative | '-' multiplicative'  
  
TEMP2 : (TEMP1 TEMP2)?
```

2. 对于我们的程序框架，在自顶向下语法分析的过程中，如果出现一个语法错误，可以进行错误恢复以继续解析，从而继续解析程序中后续的语法单元。请尝试举出一个出错程序的例子，结合我们的程序框架，描述你心目中的错误恢复机制对这个例子，怎样越过出错的位置继续解析。

错误例子：第二行未少打了分号。

```
int main(){  
    int a  
    return 0;  
}
```

针对这个例子，执行到 `p_Statement()` 中发现 `p_Expression()` 结束后，发现下一个字符不是 `SEMICOLON`，在此处进行报错，并忽略刚才对表达式的解析，返回 `NULL`，于是完成错误恢复，然后继续解析。

3. 指出你认为的本阶段的实验框架/实验设计的可取之处

终于有标注 TODO 了，感动。