**实验报告**

**一、实验目标**

在词法分析器的基础上，借助yacc工具实现一个语法分析器，要求编写适当的语义动作，能够按照规约顺序输出需要用到的规约规则，同时绘制SysY代码的语法树。

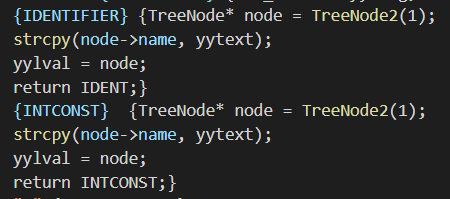
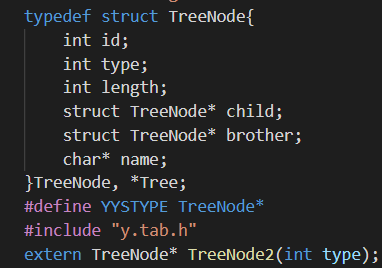
**二、代码说明**

**1. 词法分析器的修改**

为了匹配语法分析器，要对之前的词法分析程序做一些修改。每当匹配到终结符时，不再打印类别信息，而是返回语法分析器中对应的终结符符号，比如：



此外，为了生成语法树，需要为识别到的标识符和常数建立语法树结点，因此将yylval的类型由int转变为自定义的TreeNode\*类型，并且提前声明使用到的TreeNode2函数（在yacc.y中定义）。这里没有为其它终结符建立结点，因为它们的表示是固定的，所以选择在之后的语法分析程序中新建对应结点。



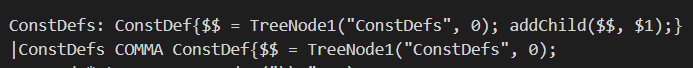
**2. 语法规则的定义**

语法规则的定义大部分按照文档直接编辑，但是对于部分正则式，需要进行修改，比如



需要变为



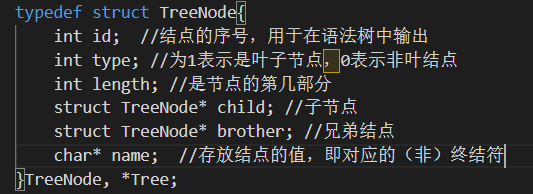


规约规则的输出，直接在语义动作中printf即可，会按照规约顺序输出。

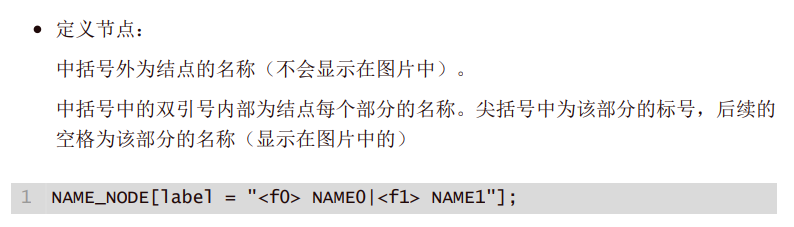
**3. 语法树的建立**

（1）结点定义

之前已经提到，我将yylval设置为自定义的TreeNode\*类型，所以语法规则中每个匹配上的非终结符和标识符、常数都是TreeNode\*类型。



其中，id对应Tree.dot文件中的节点名称NAME\_NODE；length对应节点每个部分的标号<f >；name对应结点每部分的名称NAME



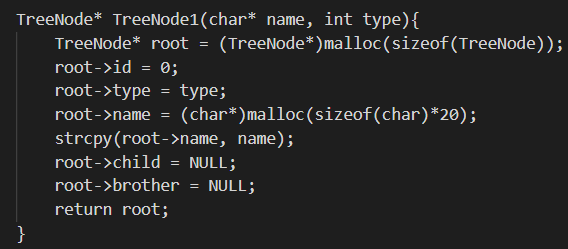
（2）语法树结构

语法树为每个匹配到的终结符和规约得到的非终结符都建立结点，每当匹配上语法规则后，就将规约元素作为根节点，匹配上的（非）终结符作为子节点（第一个匹配上的作为子节点，之后匹配上的作为子节点的兄弟节点）。即横向的兄弟节点链表存放规约元素的规约序列。举例来说，语法规则ConstDecl->CONST INT ConstDefs SEMI中，ConstDecl是根节点，const是其子节点，int是const的兄弟节点，ConstDefs是int的兄弟节点，’;’是ConstDefs的兄弟节点。然后非终结符ConstDefs又可以在其他语法规则中作为根节点，以此类推，就得到语法树的结构：纵向的父子结点表示规约，横向的兄弟节点链表存储规约序列。

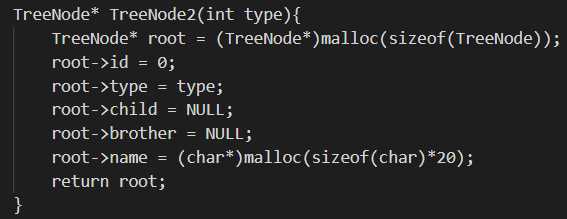


（3）与语法树相关的函数：

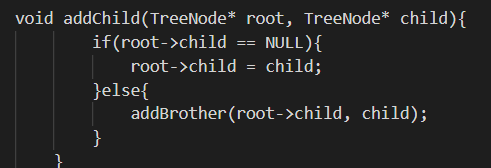
* TreeNode1函数在给定name和type的情况下新建节点，对其进行初始化并将其返回



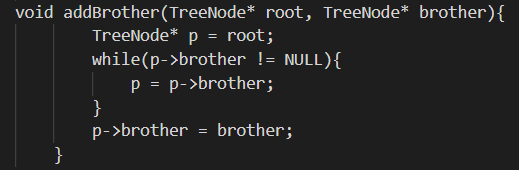
* TreeNode2函数与TreeNode1相同，只不过不提供name



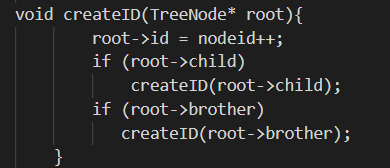
* addChild函数为root结点添加子节点child，如果该节点已经有子节点，则将child添加为子节点的兄弟节点



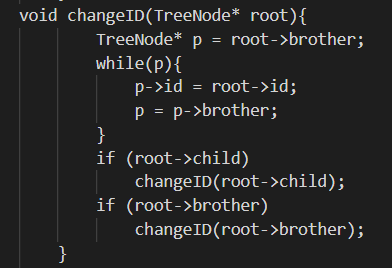
* addBrother函数为root添加兄弟节点brother（在链表最后）



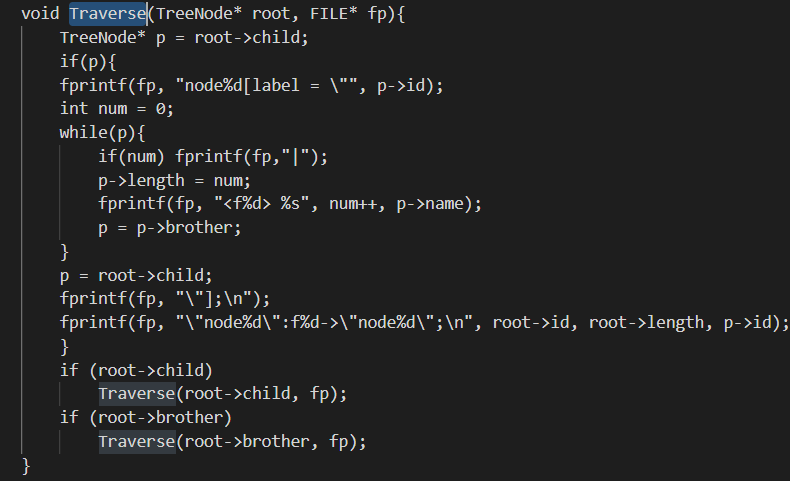
* createID函数在语法树完整建立后遍历所有节点，分配id值。因为语法树自底向上建立，所以需要在建立完成之后才分配序号。



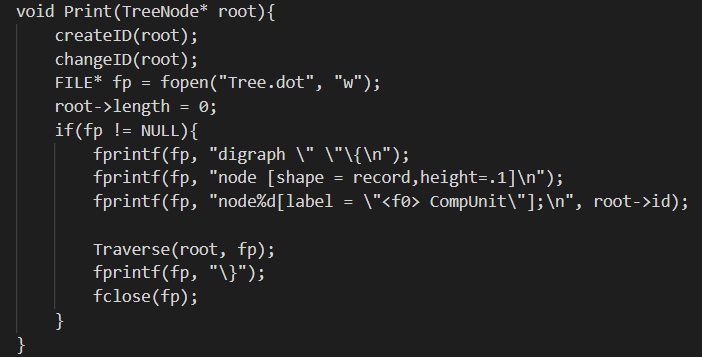
* changeID函数是为了语法树图片的美观设立，将每个规约序列的id值变成相同的（第一个结点的id），即在语法树中表示为一个节点。将length设置为每个符号在规约序列中所处的位置



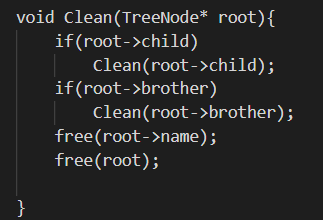
* Traverse函数对语法树进行先序遍历，输出结点和结点之间的连边关系到文件Tree.dot



* Print函数调用Traverse函数，按照graphviz的规则输出到Tree.dot文件

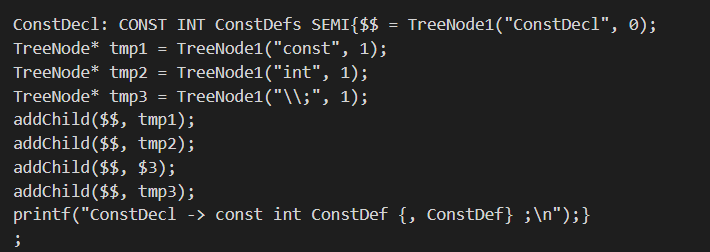


* Clean函数释放malloc申请的空间



（4）语义动作中语法树的建立

用一个例子来说明在语法规则的语义动作中如何建立语法树：首先为规约元素ConstDecl创建非叶结点，并为三个终结符‘const’、‘int’和‘；’创建叶节点；接着将规约序列“const int ConstDefs ;”对应的四个节点依次添加为ConstDecl的子节点，最后输出对应的规约规则即可。

****

**4.注意事项**

（1）if-else的处理

对if-else语句中的移进-规约冲突，做如下优先级处理即可：





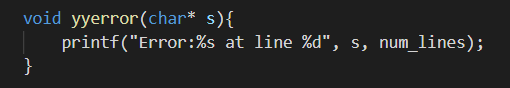
（2）四则运算优先级

做如下定义，使‘\*’‘/’的优先级比‘+’‘-’更高



(3)错误恢复

重新定义yyerror函数，当遇到错误时输出对应的错误和所在行号。



**三、结果展示**

程序可以正确运行，没有移进-规约冲突。应用于测试样例test.sy，得到测试样例按顺序的规约(较长，只展示一小部分)和语法树。

