# c2python实验报告

## 1. 实验内容

实现了从C语言到Python语言的简单编译器，支持下列C语言特性：

1. 简单的变量作用域（全局变量/局部变量）
2. 选择语句（if-else）、循环语句（for、while）
3. `#include`,`#define`预处理指令
4. 结构体和数组（非指针）
5. 浮点数、科学计数法

给出测例，实现了回文检测、排序、KMP字符串匹配、四则运算计算示例程序的翻译。

## 2. 实验环境

**操作系统：**Windows
**编程语言：**Python
**使用python库：**PLY(Lex、Yacc)

## 3. 实验原理

C语言源文件经过`lex.py`词法分析得到`token`流，再经过`yacc.py`语法分析得到抽象语法树，在`interpreter.py`程序中进行语义分析、中间代码生成，并最终输出目标代码Python源文件。

- **预处理：**将c代码的预处理命令展开
- **词法分析：**

  **功能**：词法分析完成了token的识别、空白符和注释的过滤。对匹配出错的字符流报错。

  **实现**：我们使用ply库中的ply.lex实现词法分析。包含了C99标准文档中的保留字，对于同类的其它词素进行整理得到了合乎规范的tokens并用正则表达式书写了模式。同时，注释、换行和部分错误处理也在词法分析阶段实现。
- **语法分析：**

  **功能**：语法分析对源程序的单词流进行分析，分析是否符合语言的语法规则，同时把分析结果以语法分析树或与之等价的形式表现出来。对于不符合语法规则的单词流报告语法错误。

  **实现**：使用ply库中的ply.yacc实现，语法的产生式规则整体参考了C99标准文档中的语法规则。我们在文法产生式匹配成功时会进行抽象语法树(`ast_.py`)的构造，调用`ASTInternalNode`为抽象语法树添加一颗内部节点，该节点包含了词法单元名和其子节点列表；终结符被匹配时会添加新的叶子节点`ASTLeafNode`。对源程序完成语法分析的同时也生成了一棵抽象语法树供语义分析使用。
- **语义分析：**

  **功能**：在语义分析中我们主要完成了类型检查、名字的作用域分析工作。根据语法分析生成的抽象语法树直接生成后端目标代码。

  **实现**：我们使用了语法制导的语义处理，为每个抽象语法树节点设置了code综合属性，表示每个节点翻译成后端代码的结果，自下而上递归计算每个节点的属性值，最后得到根节点的code属性就是翻译的结果。

注意到C99文档中的产生式包含很多左递归文法，所以为了保证翻译后的后端代码和翻译前的前端代码保持声明、定义顺序的一致，首先对扫描得到的声明列表和定义列表反序，然后先后对变量声明和函数声明的节点通过树遍历的方法自下向上生成后端代码。

下面是具体的细节：

1. 首先根据以下产生式提取代码中的声明和函数定义：

   > $$translation\_unit \rightarrow external\_declaration$$
   > $$translation\_unit \rightarrow translation\_unit \quad external\_declaration$$

   然后分离声明列表中的全局变量声明(定义)和函数声明，忽略函数声明，提取所有的全局变量记录到变量表中。

2. 然后`generate_code`函数对每个节点递归自下而上计算综合属性`code`，通过`code_translation`函数实现。

   `code_tranlation`是与主要产生式关联的语义规则的集合，描述了具体如何计算code属性。

3. 实现简单的作用域：在函数定义中，首先deepcopy符号表，然后进到作用域后对抽象语法树的叶子节点查看是否已经有相同名字的变量，将其重命名

4. 生成后端代码后使用yapf格式化代码

# 4.难点与创新

1. C语言标准库的实现：我们使用python预先写好标准库的常用函数如*strlen, printf, gets*等，然后在翻译的目标文件加入以下命令：

   ```
   from cstdio import *
   from cstring import *
   ```

2. 预处理指令：扫描程序记录宏定义和引入的头文件，忽略标准的c头文件。宏定义直接查找替换，对于自己写的头文件也转换成后端代码在目标文件中展开。

3. 结构体：每个抽象语法树的节点也维护一个`struct`属性表示该节点是否属于一个结构体。在生成后端代码时传入一个列表记录当前是否在嵌套的结构体中。

   - 如果节点是`struct_or_union`，将结构体的标识符加入到列表中
   - 如果递归生成的时候传入的结构体列表不为空，说明当前树节点已经在一个结构体中

   可以看到上述设计非常好地兼顾了综合属性和继承属性的特点。可以实现c语言的结构体语法。

4. 缩进问题：使用嵌套列表方式，以嵌套层数表示目标代码的缩进。

# 5. 运行结果

测试样例翻译后的结果见test目录。

1. kmp

   ```
   Please enter the text:
   12345123321

   Please enter the pattern:
   123
   A match occurs at 1
   A match occurs at 6
   ```

2. palindrome

```
D:\Github\c2python-compiler\src\test>python palindrome.py
Please input a string:12344321
It is a palindrome.
```

3. sort

```
D:\Github\c2python-compiler\src\test>python sort.py
10 14 16 18 20 23 26 26 29 32 35 37
```

4. 四则运算

```
Please enter an expression(without blank symbol):
(2+3)*(5*2)
The answer is 50.000000
```

## 6. 小组分工