

PR001-CACT-实验说明

熟悉ANTLR的安装和使用

下面的步骤依赖于ANTLR的C++工具链。

第一步：登陆本组服务器账号和GitLab账号，参见<实验环境说明>。

第二步：克隆模板，并安装antlr4的C++运行时，这是ANTLR生成源码需要依赖的。

在服务器主目录下执行：

```
git clone http://124.16.71.62:8000/Teacher/cact.git
cd cact/deps
unzip antlr4-cpp-runtime-4.8-source.zip -d antlr4-runtime
cd antlr4-runtime && mkdir build && cd build
cmake ..
make -j8
```

此时：

- 运行时库libantlr4-runtime.a：在antlr4-runtime/dist目录下。
- 头文件antlr4-runtime.h：在antlr4-runtime/runtime/src目录下。

第三步：跑通样例。以下命令均以cact文件夹为工作路径。

1. 查看是否安装JDK及版本（版本应在1.6及以上）：

```
java -version
```

2. 使用ANTLR工具生成visitor的C++代码。

```
cd grammar
java -jar ../deps/antlr-4.8-complete.jar -Dlanguage=Cpp \
    -no-listener -visitor ./Hello.g4
```

3. 编译cact项目文件夹。

```
mkdir build && cd build && cmake .. && make
```

4. 测试。

```
cp ./build/compiler ./
./compiler
```

正确结果如下：

```
root@cact:~/demo$ ./compiler
enter rule [r]!
the ID is: world
```

完成上述步骤后，无论课上课下，请各组长截图发给助教，作为一次平时作业成绩。

实验步骤

作业提交

若选择CACT实验，则将cact目录push到各组GitLab账号，参见<实验环境说明>。

编程：词法语法分析

学习demo源码，并依据<CACT语言规范>进行修改。

几个提示：

1. 入门推荐ANTLR的Visitor模式。用Listener模式亦可。
2. 注意规则排序，优先级高的排在前：'<='与'<'!='：前者在前，否则会拆。
3. 词法Token应与上下文无关：比如：语法规则 $a \rightarrow A$ ， $b \rightarrow B$ ，若文法规则A与B都能匹配X，文法层面不能确定。
4. 善于利用ANTLR工具：
 1. 自行查阅资料，通过ANTLR的locals文法为RuleContext添加属性，建构自己的AST。
 2. @header块可以给生成的各CPP文件添加#include。
5. 回想实验1整体目标（掌握编译器完整工作流程）及本次提交目标，避免纠结于细节中。

测试与打分

第一次实验成绩由两部分组成：实验报告60%、测试样例通过率40%。

1. 实验报告：请展开对以下问题的思考。如何...
 - 设计编译器的目录结构？
 - 把表达式优先级体现在文法设计中？
 - 设计数值常量的词法规则？
 - 替换ANTLR的默认异常处理方法？
2. 测试用例：
 - 公开样例（28个，80%）放在课程网站与GitLab上
 - 隐藏用例（7个，20%）请自行编译并测试。