Introduction to ANTLR

## 1. Installing ANTLR

**Step 1:** Install Java 8.

1.1. Go to https://www.oracle.com/java/technologies/downloads/

1.2. Download and install **Java SE Development Kit 8u321**

1.3. Download and install **Java SE Runtime Environment 8u321**

**Step 2:** Go to [https://www.antlr.org](https://www.antlr.org/) and follow their installing instructions.

For example: In case Linux OS or MacOS, open terminal and run below commands.

$ cd /usr/local/lib

$ sudo wget https://www.antlr.org/download/antlr-4.9.2-complete.jar

$ export CLASSPATH=".:/usr/local/lib/antlr-4.9.2-complete.jar:$CLASSPATH"

$ alias antlr4='java -jar /usr/local/lib/antlr-4.9.2-complete.jar'

$ alias grun='java org.antlr.v4.gui.TestRig'

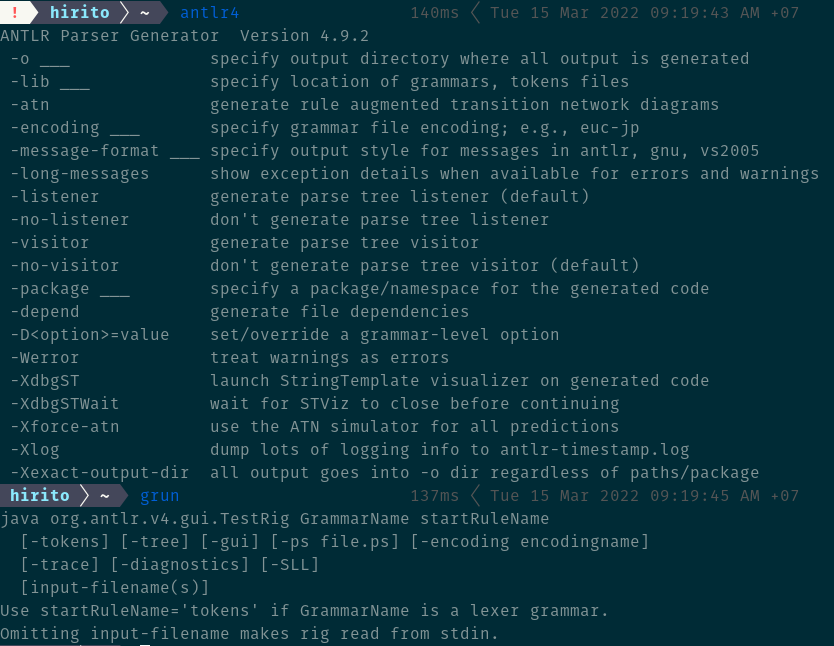
Then ANTLR will be installed at ”/usr/local/lib/antlr-4.9.2-complete.jar”

**Step 3:** Open another terminal and type following commands to see if you successfully install ANTLR.

$ antlr4

$ grun

If you see the outputs are the same like belows, you have gone on the right direction.



## 2. Setting up environment

**Step 1:** Install Python 3 (Python 3.8.10 is recommended)

1.1. Go to https://www.python.org/downloads/

1.2. Download a Python installer with version you desire

1.3 Run the downloaded installer

**Step 2:** Make Python 3 default Python (Optional)

MacOS and Linux OS:

$ alias python='python3’

$ alias pip='pip3’

**Step 3:** Install ANTLR runtime for Python 3

$ pip install antlr4-python3-runtime

**Step 4:** Set ANTLR\_JAR

$ export ANTLR\_JAR='/usr/local/lib/antlr-4.9.2-complete.jar'

## 3. How to write new programming language and perform lexical analysis

**Step 1:** Write formal definition of BKIT language in BKIT.g4

Note: At the moment we only focus on token definition. Thus, you may want to only change this part:

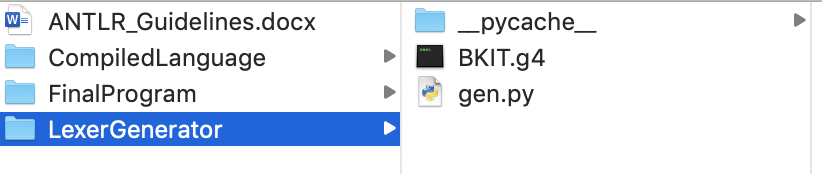
INT: [0-9]+ ;

ID: [a-z]+ ;

Feel free to add/remove other tokens as you wish.

**Step 2:** Put BKIT.g4 in LexerGenerator folder

Note: In this folder, we have already written a module named *gen.py* to generate the *Lexer* for your program,



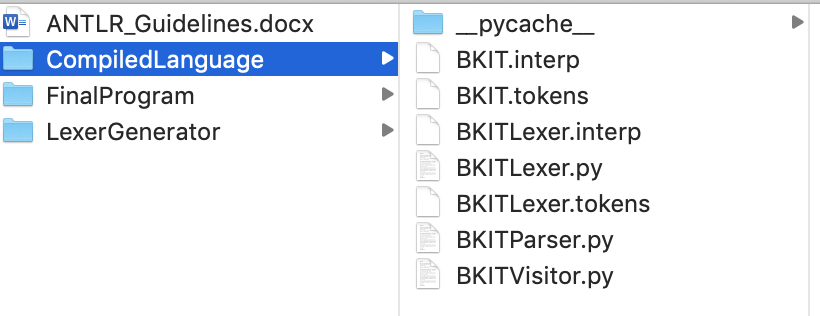
**Step 3:** Move to the LexerGenerator and execute following commands to generate lexer for BKIT language

$ python gen.py BKIT.g4

Note: in your *gen.py* file, we already set a *pathname* as

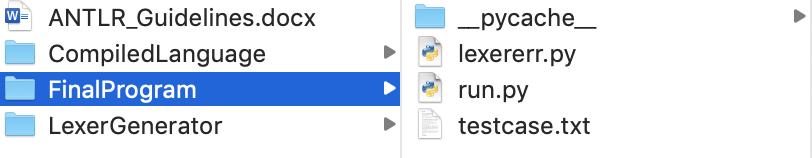
TARGET = '../CompiledLanguage'

It means that the source code of our generated lexers will be placed at the folder *CompiledLanguage*. Once checked, you will see that the following files have been generated.



You may observe that the name convention of the source code is according to the name of your g4 file.

**Step 4:** Move to *FinalProgram* folder. In this folder, you may notice that there are some other files already existing as follows.



*lexererr.py* defines some error handles. *run.py* serves as the role of main program and *testcase.txt* is an input file for testing

**Step 5:** Execute following command.

$ python run.py testcase.txt

Note: once again, in the source code of *run.py*, you may observe that we have defined the path to the *Lexer* source code, which will be integrated in the final compilation process.

TARGET = '../CompiledLanguage/'