Compilers

ANTLR Introduction: Nested Lists

3rd Bachelor Computer Science 2015-2016

Bart Meyers
bart.meyers@uantwerpen.be

18 March 2016

The goal of this assignment is to get familiar with the ANTLR lexer, parser and parse tree in Python.

1 Installation and usage of ANTLR and Python bindings

ANTLR (`http://www.antlr.org`) can be used out-of-the-box as a Java jar package. The latest version can be downloaded from `http://www.antlr.org/download/antlr-4.5.2-complete.jar`. It requires Java to run.

ANTLR converts a grammar to Python classes using the following command:

```
java -jar antlr-4.5.2-complete.jar -Dlanguage=Python2 MyGrammar.g4
-visitor
```

`MyGrammar.g4` is the text file containing your grammar. Multiple examples can be found here: `https://pragprog.com/titles/tpantlr2/source_code`. Using the `-Dlanguage` flag, the target language can be chosen (*i.e.,* Python2 or Python3). Using the `-visitor` flag, a default parse tree visitor is generated.

In order to manipulate the generated parse in Python, bindings must be installed. This can be done either automatically by executing the pip command:

```
pip install antlr4-python2-runtime
```

Or in case of Python 3:

```
pip install antlr4-python3-runtime
```

Alternatively (if you do not have installation rights), the source code of the bindings can be downloaded from:

```
https://pypi.python.org/pypi/antlr4-python2-runtime

https://pypi.python.org/pypi/antlr4-python3-runtime
```

Place the subfolder `antlr4` in your Python path.

A quick introduction on the Python bindings can be found here: `https://github.com/antlr/antlr4/blob/master/doc/python-target.md`.

2 Nested Lists in ANTLR

- Write a grammar that describes a simple language for nested lists of integers, such as the following example:

  ```
  (1, (2, 3, 4, 5), (), ((6)))
  ```

  Generate Python files from this grammar using ANTLR as described above.

- In Python, extend the generated listener to print the flattened list (because of the simplicity of this example, we do not convert the more verbose ANTLR parse tree to a more convenient and concise abstract syntax tree). For the example:

  ```
  1, 2, 3, 4, 5, 6
  ```

- In Python, extend the generated visitor to do exactly the same thing. Visitors can be used to customize the parse tree traversal (default is visitChildren), although you will not need this in this simple example.

- In Python, extend the generated listener to perform a semantic check: check whether all integer elements are in increasing order. If this is not the case, a suitable error message that includes line and character number of the wrong item, should be printed. For example, consider the input:

  ```
  (1, (2, 3, 4, 5), (), ((6), 3))
  ```

  After the semantic check, the following error message is printed and the traversal is quit:

  ```
  Wrong order at line 1, char 28:  3 is smaller than 6
  ```

Use the python `dir()` function with ANTLR objects as parameter to find out what fields and methods are available, and use the Python API: `http://msdl.cs.mcgill.ca/people/bart/compilers/antlr4-python2-doxygen.zip` (open index.html). For more information on the API, you can use the Java API: `http://www.antlr.org/api/Java/index.html`.