Exercise09: Lex

Objectives:

Learn about Type-3 or regular grammar. Learn how to write a recognizer for regular grammars.

Work with your group (or by yourself). Each group is to upload only one submission.

1 Warm Up: Try Some Examples

1. First, open blackboard, go to Course Contents, and then download exercise09.zip file into your workspace (U:\workspace or something like that!). Then, unzip.

2. Take a look at the slides on grammer in 02_grammer.pdf.

3. Take a look at LEX

   http://dinosaur.compilertools.net/lex/

   http://web.mit.edu/gnu/doc/html/flex_1.html (FLEX)

4. Flex Tutorial

   Flex is downloadable from: http://sourceforge.net/projects/flex/files/

   Please read the concepts in below links. And also try examples provided in the tutorial:

   http://flex.sourceforge.net/manual/Simple-Examples.html#Simple-Examples

5. To use Flex, you can connect to Pyrite pyrite.cs.iastate.edu (e.g. using Putty for example) which has lex/flex/flex++ installed on it. Use lex or flex to compile your .l file and cc to compile lex.yy.c into an executable.

6. Play and get familiar with the warm-up examples given in 03_examples folder. Make sure to read the 0_README.txt file first. Then, open each .l file and try and understand how it works. Then, use lex to generate the lex.yy.c file and then compile it to create an executable. Finally, run it and type in some input to see it working.

2 Lexical Analysis of XML files

Using lex or flex, write a simplified xml lexical analyzer. Assume that the input stream is a xml file. Your code should tokenize the xml file successfully. For each construct in the xml file, you are to print out the **name** of construct with details of its associated attributes or elements, if any. Here are some input/outputs of the program.

INPUTS	OUTPUTS
<?xml version=1.0 encoding=UTF-8 ?>	xml declaration version = 1, encoding = UTF-8
<Greeting>Hello, world.</Greeting>	element name = Greeting, text=Hello, world.
< Section />	empty element name=Section

Your code should also have the ability to detect if the xml file has a tag which is not closed properly (HINT: use a data structure to keep track of opened but not yet closed tag) .

```
Example: <?xml version="1.0" encoding="UTF-8"?>
            <head>
              <title>ABC Products</title>
              <meta http-equiv="Content-Type" content="text/html;" />
            <body />
```

This is invalid because <head> is not closed.

Also, remember that element names in xml follow these rules:

- Names can contain letters, numbers, and other characters
- Names cannot start with a number or punctuation character
- Names cannot start with the letters xml (or XML, or Xml, etc)
- Names cannot contain spaces

7. Submission

Submit the lex file, the generated lex.yy.c file, and the executable all zipped into one file.

Please include your group number in the .zip file name, for example, group_xx.zip.