Spring 2020

CSC 419 - Programming Languages - Assignment 2

Due date: Tuesday, Feb 11, 2020, by midnight, upload in home folder of class

Please put your full name on the assignments!

All your files for an assignment should be in a subdirectory in your home directory. The subdirectory should be named "assignmentX' where X is the assignment number. The TA will pick up the files from this subdirectory after the deadline for the assignment.

Exercise 2.1

a) Consider the EBNF description:

```
<expr> → <expr> <op> <expr> | <id>
<op>→ * | -
<id> → A
```

Prove that the grammar is ambiguous. Explain your answer. [7 points]

Exercise 2.2*

a) Given the following EBNF declarations (based on Modula-2)

```
<simple expression> → [ + | - ] <term> { + <term> | - <term> | OR <term> }
<term> → <factor> { * <factor> | / <factor> | DIV <factor> | MOD <factor> | AND <factor> | & <factor> }
<factor> → <number> | <ident> | (<simple expression>) | NOT<factor>
<ident> → <letter> { <letter> | <digit> }
```

Construct two examples of a <simple expression>, using only terminals in the language. Try to make each example significantly different from the other so as to illustrate the range of expressions which this definition encompasses. Assume simple definitions for <letter> (i.e., a..z, A..Z), <digit> (i.e., 0..9), and <number> (i.e., <digit>{<digit>}) although the actual definitions in Modula-2 are more complex.
[6 points]

*exercise adopted from Jonathan Mohr

Exercise 2.3

a) Look up the definition of the **list** data structure in Python. Give two examples of lists in Python. [4 points]

b) Similar to what we did in class for the C Union data structure (also posted on the class website), write EBNF (or BNF) descriptions for a **list** data structure in Python. The level of detail could be similar to the Union example. It's fine to keep simple and just choose two data types that will be inside the list (example, integers or strings). Allow lists within lists. (hint: You do not need to show this, but to convince yourself that you are writing the EBNF properly, the EBNF description should be able to generate examples such as the ones that you have in (a)) [3 points]