Fall 2025 CIS 424 Project 2

(Due Oct. 23)

In this project you are asked to write an interpreter which uses the top-down recursive-descent method to parse and evaluate a very simple programming language. A C++ version of the interpreter is given in the Blackboard. You must write the interpreter in Python following the same structure of the C++ code, meaning that you should implement the same functions of C++ code in your Python code. The tiny language's grammar is given below.

<stmt-list> ::= empty | <stmt> { <stmt> }

<stmt> ::= id = <expr> ; |
print <expr> ;

<expr> ::= <term> { + <term> | - <term> }

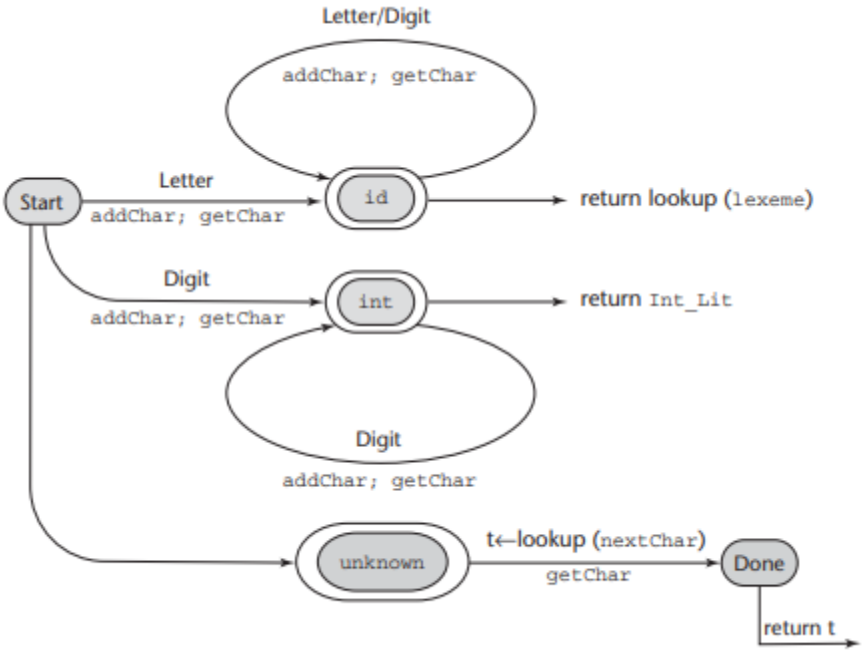
<term> ::= <factor> { * <factor> | / <factor> }

<factor> ::= id | intnum | (<expr>)

The definitions of id and intnum follows Figure 4.1 in the textbook.

Figure 4.1

A state diagram to recognize names, parentheses, and arithmetic operators



The interpreter (parser_xxxxxxx.py where xxxxxxx is your CSU ID) should be written in Python. It takes one input file (sample.tiny) which is a text file consisting of the statements of the above grammar. The input file name sample.tiny is given from the command line. The interpreter parser_xxxxxxx.py reads the program file sample.tiny, checks the syntax and outputs the result if the program is legitimate; otherwise, the interpreter prints "Syntax Errors" (this part is not in the C++ code).

Below are some test examples in sample.tiny:

```
print 2 + 3 * 4 ;
s = 2 + 3 ;
t = 9 - 2 ;
print s * t ;
print ( s + t ) * ( s - t ) ;
```

Please issue "python3 parser_xxxxxxx.py sample.tiny" on the **spirit** machine (Python version 3.8.10) to test your Python program **before** you turnin it. Please use "if __name__ == '__main__':" in your code (read section 3.17 at https://google.github.io/styleguide/pyguide.html).
You also need to hand in a parser_xxxxxxx.pdf document which includes your experiences in developing, debugging and testing your Python code. Please do NOT put the entire source codes into the .pdf file. The cover page should contain your name, your CSU ID and the login id you used to turnin the project.

Turning it in

You must submit your program and pdf document electronically by using the following command on the **grail** machine

turnin -c cis424x -p proj2 parser_xxxxxxx.py parser_xxxxxxx.pdf

Start on time and good luck. If you have any questions, send e-mail to s.tomar14@vikes.csuohio.edu or w.xiong15@csuohio.edu.
Sample running results:

```
spirit:~/proj2sol% python3 eval.py empty.tiny
>>> Empty .tiny file.
spirit:~/proj2sol% python3 eval.py sample.tiny
print 2 + 3 * 4 ;
>>> 14
s = 2 + 3 ;
t = 9 - 2 ;
print s * t ;
>>> 35
print ( s + t ) * ( s - t ) ;
>>> -24
```