Swathi Sri Vishnu Priya Rayala

Onid: rayalas

CS 562: Applied Software Engineering

Part 3 – Bug/Progress Report

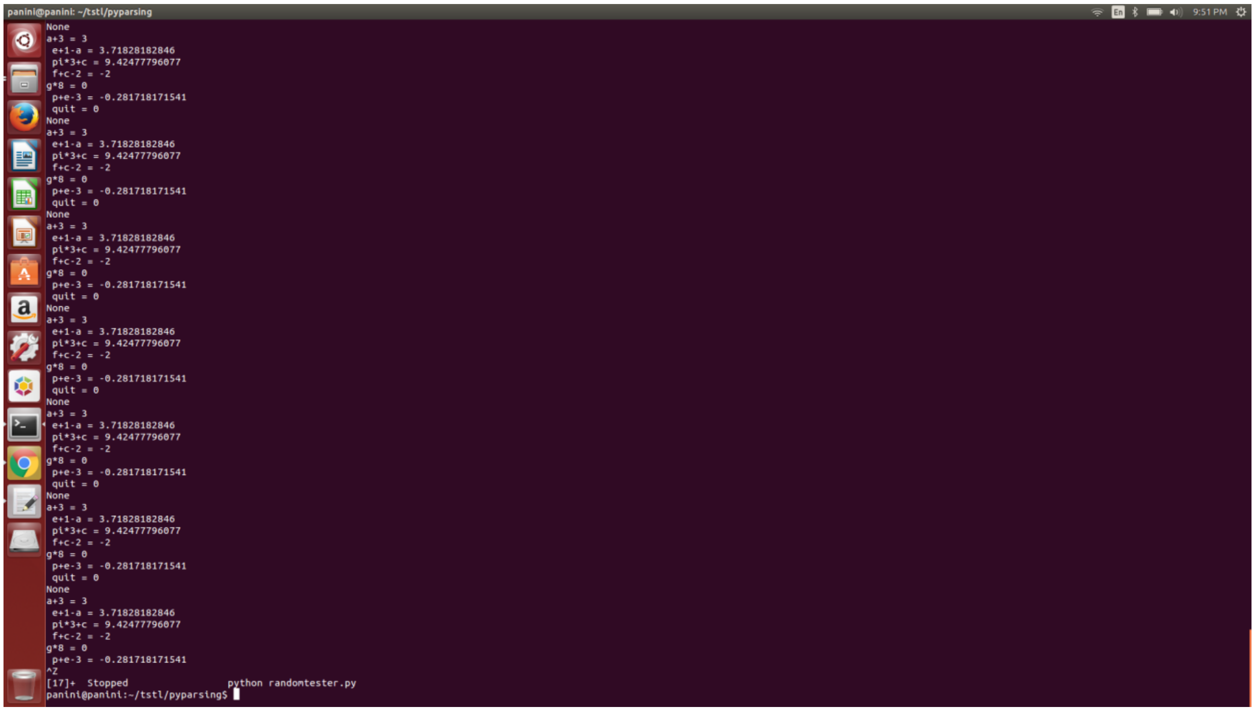
Describe some bugs you found

I have tested pyparsing.py library. I have created a grammar for executing the arithmetic expressions and SQL Queries. There few errors that I found while testing the grammar with random tester of TSTL.

Arithmetic Expressions – The grammar failed to evaluate expressions that contain variables.

For example, expressions like ((3 + pi) * e) - 12 are evaluated correctly. However, expressions like (a + 3) * 4 are not evaluated correctly. The grammar interprets the variables a, b, etc as zero. But, the variables like e and pi are taken as 2.71828182846 and 3.14159265359 respectively. This is something weird as the grammar should give an error rather than taking the value as zero.

Also, the expressions with only 2 mathematical constants (e and pi) are interpreted and evaluated correctly. The mathematical constant i or i^2 is not accepted by the grammar. It is however evaluated as zero.

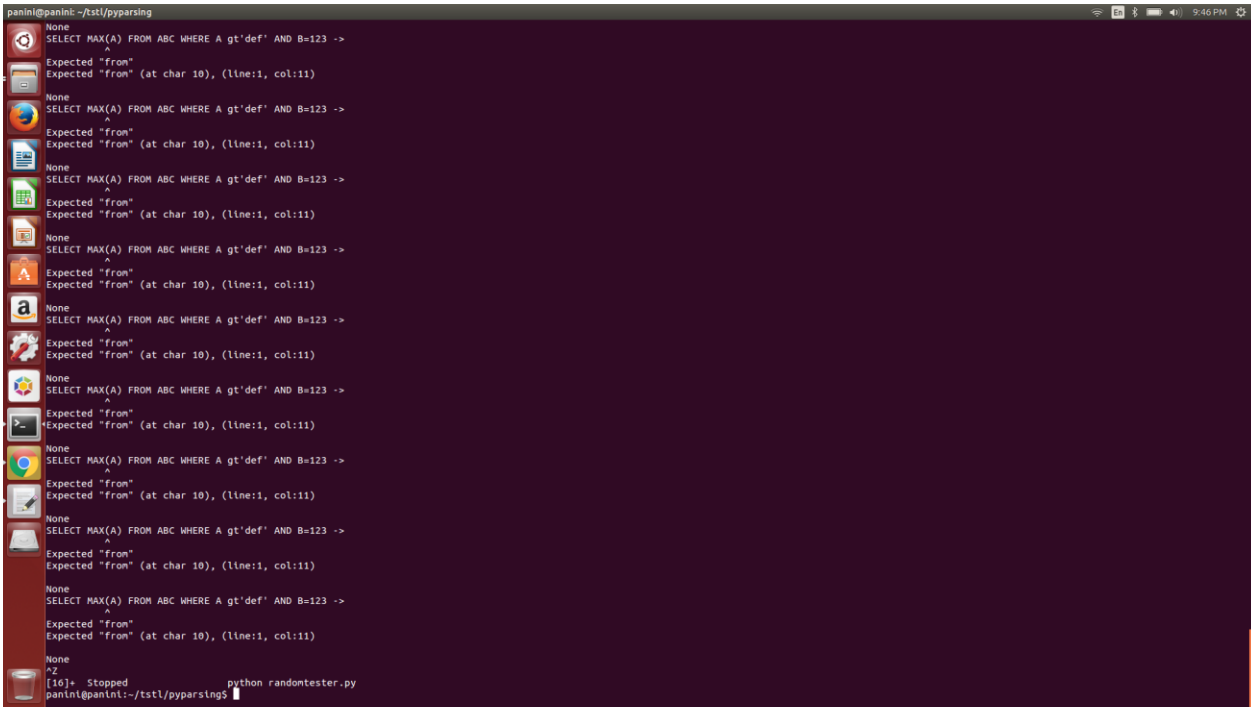


```
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
 quit = 0
None
a+3 = 3
 e+1-a = 3.71828182846
 pi*3+c = 9.42477796077
 f+c-2 = -2
g*8 = 0
 p+e-3 = -0.281718171541
^Z
[17]+  Stopped                 python randomtester.py
panini@panini:~/tstl/pyparsing$
```

SQL Queries – The grammar failed to show up column names of type MAX(*column_name*)

For queries like, *SELECT A, B, C from ABC_Table* are successfully executed by the grammar by displaying the column names as [A, B, C] and Table name as ABC_Table

But for queries like *SELECT MAX(A) from ABC_Table*, the grammar failed to display the column names. However, it is successful to display the Table Name.



```
None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
SELECT MAX(A) FROM ABC WHERE A gt'def' AND B=123 ->
          ^
Expected "from"
Expected "from" (at char 10), (line:1, col:11)

None
^Z
[16]+  Stopped                 python randomtester.py
panini@panini:~/tstl/pyparsing$
```

Explain progress to date

The following test cases have been done so far –

Random expressions are generated and tested correctly

Lengthy expressions are evaluated correctly

The four operators (+, -, *, /) are used for testing

Expressions with mathematical constants (e and pi) are tested

SQL queries with different column names are tested

SQL queries with WHERE conditions are verified

Estimate your progress by end of term

The following are to be done –

Should verify expressions having “^ “ operator. This is necessary to test because when a number raised to the power of a large number, the result would be very large. In that case, the performance for the computation of the expression can be verified.

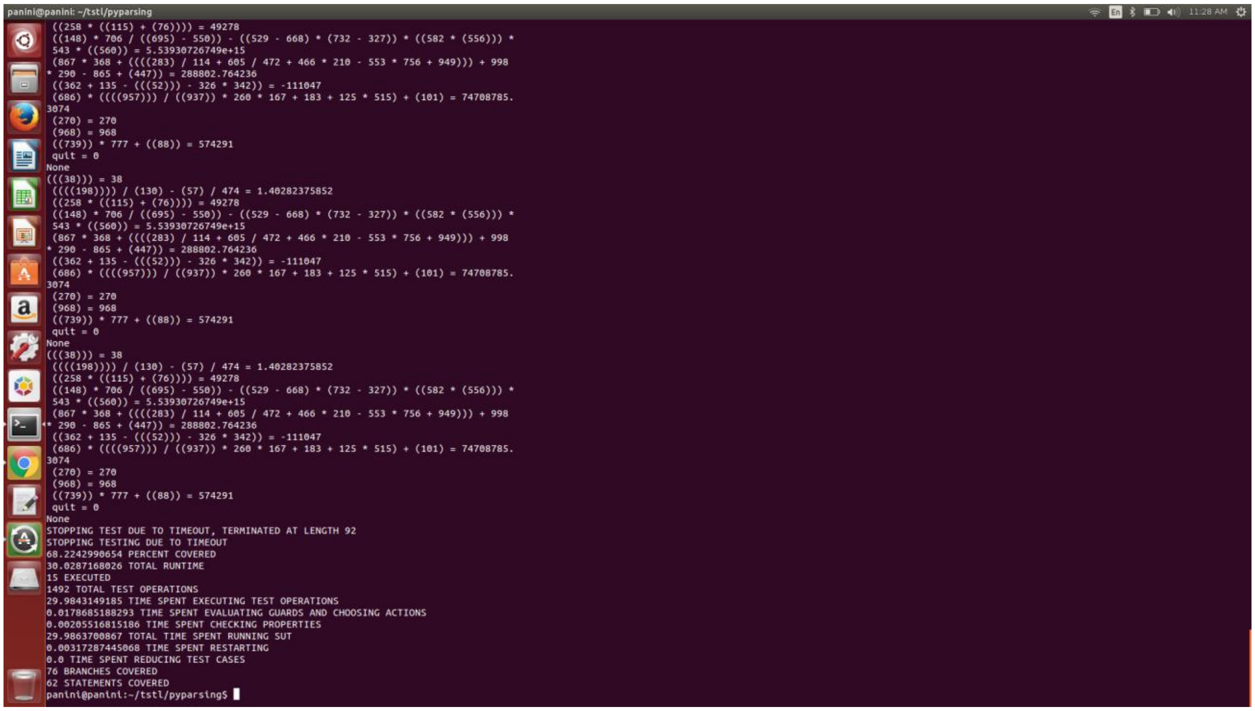
Also, the assignment operations should be verified when testing the expressions. For example, area = 3 * 4, should be verified.

Discuss quality of SUT

The library is well maintained and well documented with comments to be understood by a naïve user. The library is designed in such a way that it can be scalable. New requirements for a grammar can be easily incorporated in to it with the addition of a class and methods for it. Also, the response time for any method of the library is acceptable.

Code Coverage

About 68% of the code is covered by testing with TSTL. However, few other methods are still left to test. As pyparsing is a very large library, I tried to incorporate as many functions as I can to implement the grammar and test with TSTL. I have covered testing the functions with various inputs by randomly generating using probability function. This helped to test the performance of the library for very large inputs and computational speed.



```
 ((258 * ((115) + (76)))) = 49278
 ((148) * 706 / ((695) - 550)) - ((529 - 668) * (732 - 327)) * ((582 * (556))) *
 543 * ((560)) = 5.53930726749e+15
 (867 * 368 + ((((283) / 114 + 605 / 472 + 466 * 210 - 553 * 756 + 949))) + 998
* 290 - 865 + (447)) = 288802.764236
 ((362 + 135 - (((52))) - 326 * 342)) = -111047
 (686) * ((((957))) / ((937)) * 260 * 167 + 183 + 125 * 515) + (101) = 74708785.
3074
 (270) = 270
 (968) = 968
 ((739)) * 777 + ((88)) = 574291
 quit = 0
None
(((38))) = 38
 ((((198)))) / (130) - (57) / 474 = 1.40282375852
 ((258 * ((115) + (76)))) = 49278
 ((148) * 706 / ((695) - 550)) - ((529 - 668) * (732 - 327)) * ((582 * (556))) *
 543 * ((560)) = 5.53930726749e+15
 (867 * 368 + ((((283) / 114 + 605 / 472 + 466 * 210 - 553 * 756 + 949))) + 998
* 290 - 865 + (447)) = 288802.764236
 ((362 + 135 - (((52))) - 326 * 342)) = -111047
 (686) * ((((957))) / ((937)) * 260 * 167 + 183 + 125 * 515) + (101) = 74708785.
3074
 (270) = 270
 (968) = 968
 ((739)) * 777 + ((88)) = 574291
 quit = 0
None
(((38))) = 38
 ((((198)))) / (130) - (57) / 474 = 1.40282375852
 ((258 * ((115) + (76)))) = 49278
 ((148) * 706 / ((695) - 550)) - ((529 - 668) * (732 - 327)) * ((582 * (556))) *
 543 * ((560)) = 5.53930726749e+15
 (867 * 368 + ((((283) / 114 + 605 / 472 + 466 * 210 - 553 * 756 + 949))) + 998
* 290 - 865 + (447)) = 288802.764236
 ((362 + 135 - (((52))) - 326 * 342)) = -111047
 (686) * ((((957))) / ((937)) * 260 * 167 + 183 + 125 * 515) + (101) = 74708785.
3074
 (270) = 270
 (968) = 968
 ((739)) * 777 + ((88)) = 574291
 quit = 0
None
STOPPING TEST DUE TO TIMEOUT, TERMINATED AT LENGTH 92
STOPPING TESTING DUE TO TIMEOUT
68.2242990654 PERCENT COVERED
30.0287168026 TOTAL RUNTIME
15 EXECUTED
1492 TOTAL TEST OPERATIONS
29.9843149185 TIME SPENT EXECUTING TEST OPERATIONS
0.0178685188293 TIME SPENT EVALUATING GUARDS AND CHOOSING ACTIONS
0.00205516815186 TIME SPENT CHECKING PROPERTIES
29.9863700867 TOTAL TIME SPENT RUNNING SUT
0.00317287445068 TIME SPENT RESTARTING
0.0 TIME SPENT REDUCING TEST CASES
76 BRANCHES COVERED
62 STATEMENTS COVERED
panini@panini:~/tstl/pyparsing$ █
```