CSC402 - Assignment #2

1 Introduction

Here we add actions to our parser. The idea is that we want to count how many TOP LEVEL sexp's a particular μLisp program contains. For example, for the first and second program below your processor should print out the value '1' and '3', respectively. Here is the grammar of the μLisp language again for completeness sake,

```
prog : sexp prog
     | ""

sexp : '(' exp ')'
     | var
     | num

exp : '+' sexp sexp
    | '-' sexp sexp
    | '*' sexp sexp
    | '/' sexp sexp
    | 's' var sexp
    | 'p' sexp

var : 'x' | 'y' | 'z'

num : '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
```

2 Problems

1. (10pts) Develop a bottom-up parser for μLisp written in Python/Ply. A good spot to start are the exp0_gram.py file in the Plipy 'code' folder. You can use ulisp_lex.py for lexing/stream processing. This file is also available in the 'code' folder. Show that your parser works correctly for the programs below.

2. (10pts) Add actions with the necessary code in order to count the top level sexp's. Show that it works correctly for all the programs listed below.

3 Example Programs

Demonstrate that your processors work by processing the following μLisp programs:

1.
```
(p (+ (* 3 2) 1))
```

2.
```
(s x 1)
(s y x)
(p (+ x y))
```

3.
```
(s x 1)
(p x)
(s y 2)
(p y)
(p (+ x y))
```

Your processor should *reject* the following program:

```
(s x 1)
(s y x)
(p (+ x p))
```

4 Deliverables

The processors should be written in Python using Ply. Hand in your source code together with a Jupyter Notebook for each processor that shows that your processor works. To submit your work create a zip file of your sources and the notebooks and submit it through BrightSpace. Assignments submitted in formats other than Jupyter Notebooks will not be graded and a failing grade will be recorded.