**CPSC 323 Project 2**

**Overview**

This program involves creating a program that traces input strings over a given set of characters and symbols that end with a '$' sign. Given the CFG and the parsing table, we have to trace input strings over the set {id+\*)(} ending with '$'. We also have to test with the three given input strings (1)(id+id)id$ (2) idid$ and (id\*)$.

**Setup**

1. Clone or download this repository to your local machine
2. git clone <https://github.com/kevinvqle/323hw2.git>
3. Make sure you have Python downloaded on your device

**Usage**

1. Open a terminal and navigate to the directory where the \*\*CFGScaner.py\*\* script is located.
2. Run the lexer by exexuting the following command.
   1. python3 CFGScaner.py
3. The output will be output as step, stack, input and action with all 3 strings being tested.

**Results**

[Running] python3 -u "/Users/alexly/Desktop/323hw2/program.py"

Parsing string: ( id + id ) \* id $

Step: 0, Stack: ['0'], Input: ['(', 'id', '+', 'id', ')', '\*', 'id', '$', '$'], Action: S4

Step: 4, Stack: ['0', '(', '4'], Input: ['id', '+', 'id', ')', '\*', 'id', '$', '$'], Action: S5

Step: 5, Stack: ['0', '(', '4', 'id', '5'], Input: ['+', 'id', ')', '\*', 'id', '$', '$'], Action: R6

Step: 3, Stack: ['0', '(', '4', 'F', '3'], Input: ['+', 'id', ')', '\*', 'id', '$', '$'], Action: R4

Step: 2, Stack: ['0', '(', '4', 'T', '2'], Input: ['+', 'id', ')', '\*', 'id', '$', '$'], Action: R2

Step: 8, Stack: ['0', '(', '4', 'E', '8'], Input: ['+', 'id', ')', '\*', 'id', '$', '$'], Action: S6

Step: 6, Stack: ['0', '(', '4', 'E', '8', '+', '6'], Input: ['id', ')', '\*', 'id', '$', '$'], Action: S5

Step: 5, Stack: ['0', '(', '4', 'E', '8', '+', '6', 'id', '5'], Input: [')', '\*', 'id', '$', '$'], Action: R6

Step: 3, Stack: ['0', '(', '4', 'E', '8', '+', '6', 'F', '3'], Input: [')', '\*', 'id', '$', '$'], Action: R4

Step: 9, Stack: ['0', '(', '4', 'E', '8', '+', '6', 'T', '9'], Input: [')', '\*', 'id', '$', '$'], Action: R1

Step: 8, Stack: ['0', '(', '4', 'E', '8'], Input: [')', '\*', 'id', '$', '$'], Action: S11

Step: 11, Stack: ['0', '(', '4', 'E', '8', ')', '11'], Input: ['\*', 'id', '$', '$'], Action: R5

Step: 3, Stack: ['0', 'F', '3'], Input: ['\*', 'id', '$', '$'], Action: R4

Step: 2, Stack: ['0', 'T', '2'], Input: ['\*', 'id', '$', '$'], Action: S7

Step: 7, Stack: ['0', 'T', '2', '\*', '7'], Input: ['id', '$', '$'], Action: S5

Step: 5, Stack: ['0', 'T', '2', '\*', '7', 'id', '5'], Input: ['$', '$'], Action: R6

Step: 10, Stack: ['0', 'T', '2', '\*', '7', 'F', '10'], Input: ['$', '$'], Action: R3

Step: 2, Stack: ['0', 'T', '2'], Input: ['$', '$'], Action: R2

Step: 1, Stack: ['0', 'E', '1'], Input: ['$', '$'], Action: acc

String is accepted.

Parsing string: id \* id $

Step: 0, Stack: ['0'], Input: ['id', '\*', 'id', '$', '$'], Action: S5

Step: 5, Stack: ['0', 'id', '5'], Input: ['\*', 'id', '$', '$'], Action: R6

Step: 3, Stack: ['0', 'F', '3'], Input: ['\*', 'id', '$', '$'], Action: R4

Step: 2, Stack: ['0', 'T', '2'], Input: ['\*', 'id', '$', '$'], Action: S7

Step: 7, Stack: ['0', 'T', '2', '\*', '7'], Input: ['id', '$', '$'], Action: S5

Step: 5, Stack: ['0', 'T', '2', '\*', '7', 'id', '5'], Input: ['$', '$'], Action: R6

Step: 10, Stack: ['0', 'T', '2', '\*', '7', 'F', '10'], Input: ['$', '$'], Action: R3

Step: 2, Stack: ['0', 'T', '2'], Input: ['$', '$'], Action: R2

Step: 1, Stack: ['0', 'E', '1'], Input: ['$', '$'], Action: acc

String is accepted.

Parsing string: ( id \* ) $

Step: 0, Stack: ['0'], Input: ['(', 'id', '\*', ')', '$', '$'], Action: S4

Step: 4, Stack: ['0', '(', '4'], Input: ['id', '\*', ')', '$', '$'], Action: S5

Step: 5, Stack: ['0', '(', '4', 'id', '5'], Input: ['\*', ')', '$', '$'], Action: R6

Step: 3, Stack: ['0', '(', '4', 'F', '3'], Input: ['\*', ')', '$', '$'], Action: R4

Step: 2, Stack: ['0', '(', '4', 'T', '2'], Input: ['\*', ')', '$', '$'], Action: S7

Step: 7, Stack: ['0', '(', '4', 'T', '2', '\*', '7'], Input: [')', '$', '$'], Action:

Error: Invalid action. String is not accepted.

[Done] exited with code=0 in 0.05 seconds

**Authors**

Kevin Le

Christine Le

Alex Ly