**Name: Romel Niño O. Paano Date: 3/15/17**

**Section: CSA**

**Introduction:**

In this program can tell if the sentence that you have given is compatible in the given grammar. For, example if a user entered A=A it will check in this manner.

Grammar:

<assign> -> <id>= <expr>

<id> -> A | B | C

<expr> -> <expr>+<term>

| <expr>-<term>

|<term>

<term> -> <term>\*<factor>

|<term>/<factor>

|<factor>

<factor> -> (expr)

|<number>

|<id>

<number> -> <digit><number>

|<digit>

<digit> -> 0|1|2|3|4|5|6|7|8|9

In derivation it will do this:

<assign> => <id>= <expr>

=> A = <expr>

=> A = <term>

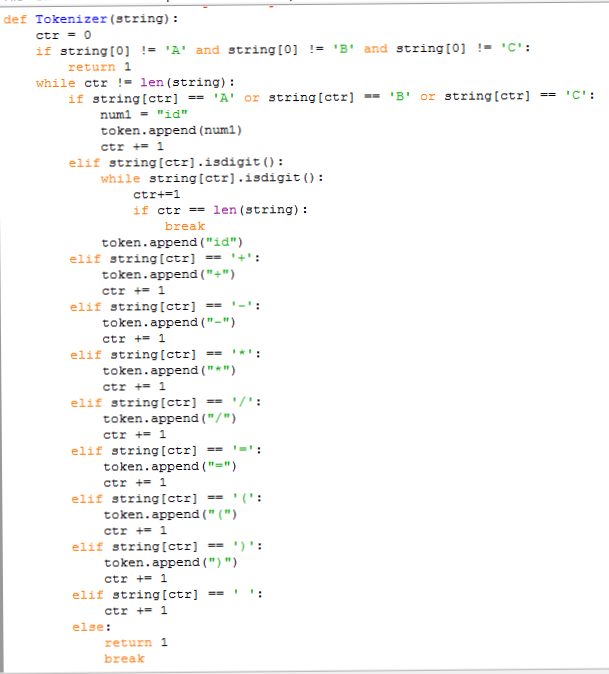
=> A = <factor>

=> A = <id>

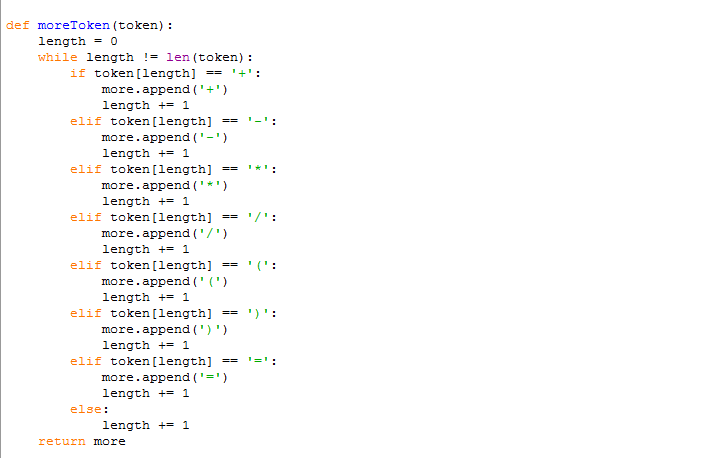
=> A = A

**Design:**

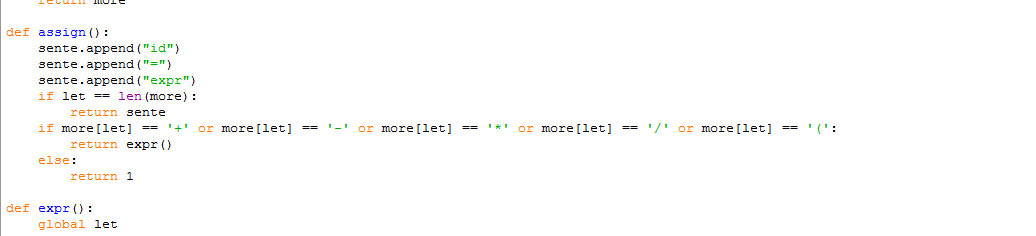
In programming, it is a bit complicated because a programmer need to understand how the grammar word or how a typical human will solve it. In order to program it must be tokenized first then parse it with the grammar.



This part of the program is the tokenizer because it breaks the strings in to token of the original string. And bit of parsing because if program detects that there is no equivalent to the program it will automatically returns 1.



In part I called it “moreToken” because it takes only the operators an stores it in the a list.



In this one, this is one of the parsing part of the program. In the first part it appends “id = expr” to sente (sente is short for sentence). Then check if equal to the length of “more” list or check the operation but if it doesn’t it will return 1 which equivalent to False.

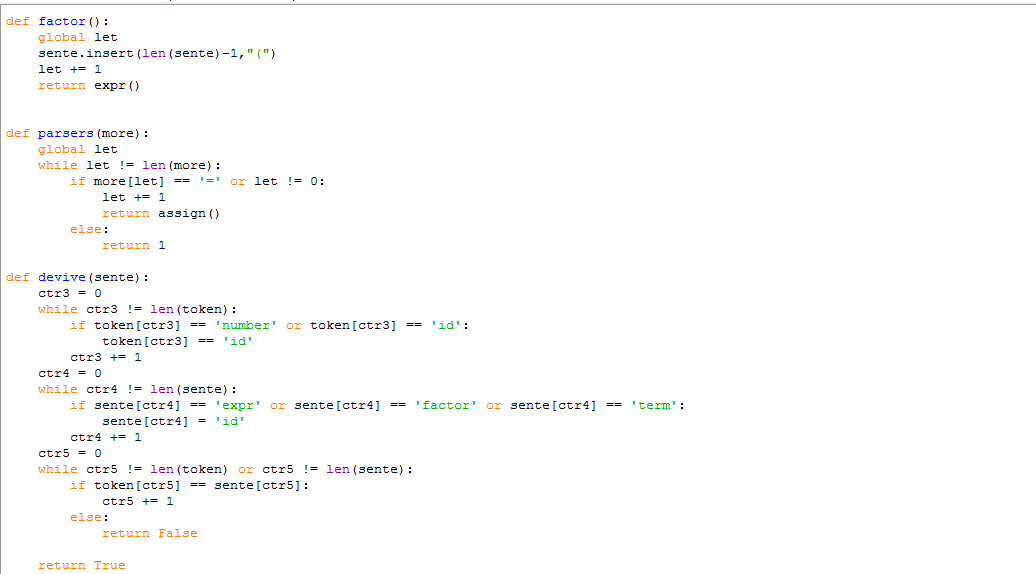


This second part will only be avaible if the first part is not satisfied. I can’t explain it properly but I can give an example.

Input: A=A+A+A

It will produce: id = expr + term

Id = expr + term + term



The first part of the picture is the third and last part of the parsing. It will in insert this “(” in the program then calls the expr()

The second part is the first part of the parsing but it is not really parsing because it only checks if the first operator is an equal sign then call the assign()

The thirf part is devive (sorry about the spelling error it should be derive). It only do is to rewrite all the expr, term, number, factor into id to make it easier to compare.