Lab 7 FLCD – PARSER

Vieru Tudor-Gabriel & Toncea Ion-Alin

937/2

GitHub Repository

https://github.com/TonceaAlin/FLCDlab

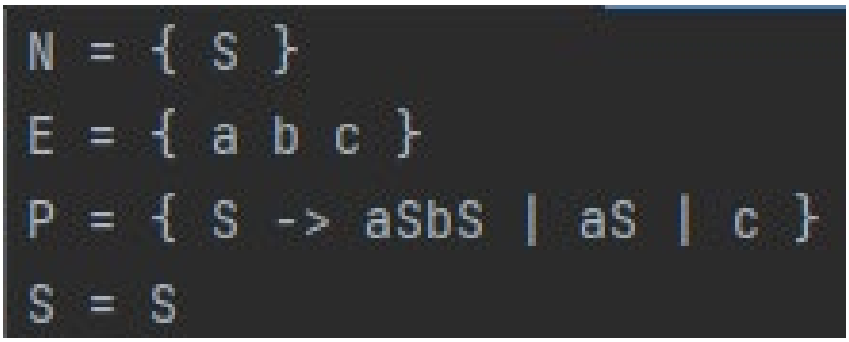
Statement

Implement a parser using the recursive descendent algorithm.

Example

g1.txt – sequence – output

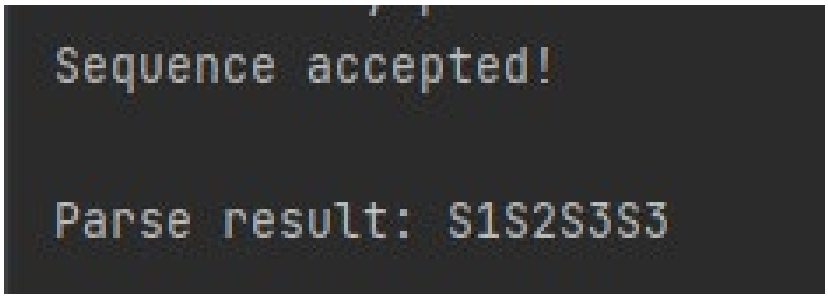
g1.txt format:



Sequence:

aacbc

Parsing output:



out1.txt

1233

Implementation

-Grammar

```
class Grammar:
    def __init__(self, N, E, P, S):
        self.N = N
        self.E = E
        self.S = S
        self.P = P
```

Holding information about terminals(N), non-terminals(E), the starting symbol(S) and the set of productions(P).
Most of the methods are implemented to operate on the g1.txt file.

-Parser

```
class Parser2:
    def __init__(self, grammar, sequence):
        self.grammar = grammar
        self.state = 'q'
        self.index = 1
        self.workingStack = []
        self.inputStack = grammar.getInitialState()
        self.sequence = sequence
```

Referencing to the grammar above, having information about the state in which the recursive descendent algorithm is(q, b, f or e), the index from the sequence, the actual sequence and the stacks(working and input).
We implemented the needed methods to parse the program with the recursive descendant algorithm: expand, back, advance, momentary insuccess and another try. Using all of them in a while and checking each piece from the sequence will get us to the final result.